

# AI-Driven Blind Signature Classification for IoT Connectivity: A Deep Learning Approach

Jianxiong Pan, Neng Ye, *Member, IEEE*, Hanxiao Yu, Tao Hong, Saba Al-Rubaye, *Senior Member, IEEE*, Shahid Mumtaz, *Senior Member, IEEE*, Anwer Al-Dulaimi, *Senior Member, IEEE*, and Chih-Lin I, *Fellow, IEEE*

**Abstract**—Non-orthogonal multiple access (NOMA) promises to fulfill the fast-growing connectivities in future Internet of Things (IoT) using abundant multiple-access signatures. While explicitly notifying the utilized NOMA signatures causes large signaling cost, blind signature classification naturally becomes a low-cost option. To accomplish signature classification for NOMA, we study both likelihood- and feature-based methods. A likelihood-based method is firstly proposed and showed to be optimal in the asymptotic limit of the observations, despite high computational complexity. While feature-based classification methods promise low complexity, efficient features are non-trivial to be manually designed. To this end, we resort to artificial intelligence (AI) for deep learning-based automatic feature extraction. Specifically, our proposed deep neural network for signature classification, namely DeepClassifier, establishes on the insights gained from the likelihood-based method, which contains two stages to respectively deal with a single observation and aggregate the classification results of an observation sequence. The first stage utilizes an iterative structure where each layer employs a memory-extended network to explicitly exploit the knowledge of signature pool. The second stage incorporates the straight-through channels within a deep recurrent structure to avoid information loss of previous observations. Experiments show that DeepClassifier approaches the optimal likelihood-based method with a reduction of 90% complexity.

**Index Terms**—Non-orthogonal multiple access, signature classification, deep learning, recurrent neural network, automatic feature extraction

## I. INTRODUCTION

**D**RIVEN by the unprecedented proliferation of machine-type devices and intelligent applications, novel massive access technologies are desperately required for future Internet of Things (IoT) [1]–[5]. As a promising candidate technology, non-orthogonal multiple access (NOMA) constructs a number of non-orthogonal signatures over limited orthogonal physical

This work was supported in part by the National Nature Science Foundation of China under Grants 62101048, 62171030 and 62071038, in part by the Advance Research Projects of 13th Five-Year Plan of Civil Aerospace Technology under Grant B0105, and in part by the FCT project (Intelligent and Sustainable Aerial-Terrestrial IoT Networks-BATS) under Grant PTDC/EEI-TEL/1744/2021. (*Corresponding Author: Neng Ye.*)

J. Pan, N. Ye and H. Yu are with the School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China. Email: panjianxiong@bit.edu.cn; ianye@bit.edu.cn; yuhanxiao00@bit.edu.cn.

T. Hong is with the School of Electronics and Information Engineering, Beihang University, Beijing 100083, China. Email: hongtao@buaa.edu.cn.

S. Al-Rubaye is with the School of Aerospace, Transport and Manufacturing, Cranfield University, Bedford MK43 0AL, U.K. Email: s.alrubaye@cranfield.ac.uk.

S. Mumtaz is with Instituto de Telecomunicações, Aveiro 3810-193, Portugal. E-mail: smumtaz@av.it.pt.

A. Al-Dulaimi is with Research and Development Department, EXFO, Montreal, QC H4S 0A4, Canada, E-mail: anwer.al-dulaimi@exfo.com.

C-L. I is with China Mobile Research Institute, Beijing 100053, China, E-mail: icl@chinamobile.com.

resources so as to provide abundant access opportunities [7], [8], and to accommodate the massive connectivity requirements in IoT networks [9], [10]. These well-designed signatures define how the IoT transmitters encode and spread the source information over physical resources in a non-orthogonal fashion [11]–[13]. Accordingly, advanced multi-user detection algorithms are studied to recover the source information of the IoT devices from the superimposed signals at the receiver [14]–[17].

To enable coherent NOMA detection, the receiver requires full knowledge of the multiple-access signatures used by the IoT transmitters. On the one hand, connected-based OMA/NOMA schemes achieve this using centralized scheduling, where the receiver assigns the user-specific signatures to the transmitters for uplink transmission [18], [19]. On the other hand, some NOMA schemes allows the IoT transmitter to automatically select a signature for transmission, and the indexes of the selected signatures are then transmitted to the receiver via uplink signaling [15]. However, both the above approaches require tedious signaling interactions between the transceivers and thus causes large delay and high power consumption, which are large burdens for hardware-limited IoT devices [12]. So the explicit notifications of NOMA signatures may not be suitable for the scenarios with massive low-cost devices in future IoT networks.

As an alternative approach, conducting AI-driven blind signature classification at the receiver can avoid the explicit notifications and thus reduce the signaling overheads [20], [21]. Therefore, blind signature classification can be a good option for NOMA in future IoT networks. However, with multiple overlapped signals and different fading states of the transmitters, the classification of NOMA signatures becomes non-trivial. This work aims to efficiently solve the blind signature classification problem in IoT, which, to the best of our knowledge, still lacks a unified framework for the case of NOMA.

With one single transmitter, the blind signature classification problem falls back to the automatic modulation classification (AMC) problem [22], where different signatures resemble the roles of different modulation schemes. Classical AMC technologies can be divided into likelihood-based (LB) methods [23], [24] and feature-based (FB) methods [22], [25], which respectively exploit the likelihood functions and the statistical features of the received signals. Among the above two types of approaches, the former approach makes optimal decision in Bayesian perspective at the expense of high computational complexity and delay [22]. The latter approach relies on the heuristic feature engineering to enable low-complexity

implementation, while the designed features are normally subject to certain modulations and the classification accuracy performance is not optimal. All these model-driven approaches manually decide the rules for AMC according to the exact knowledge of the modulations. However, the manual design of useful features is tedious for different kinds of constellation mappings [26].

The recent booming of deep learning provides an efficient and automatic approach for FB modulation classification [26]–[32]. Through data-driven end-to-end optimization, deep neural networks (DNNs) have strong ability in feature learning which automatically learns useful features of the inputs as hidden representations [33]. In [28], a sparse deep autoencoder method is proposed to obtain meaningful sparse representations for modulation classification. The authors in [29] and [30] further exploit the scalability of convolution neural network (CNN) to deal with sequence inputs. With the extracted features, these deep learning-based methods can approach near maximum-likelihood performance [27], [29]. Deep recurrent neural network (RNN) structure and long short term memory (LSTM) cells are deployed in [26] to learn long-term temporal dependencies. A two-layer LSTM model is implemented by stacking two classical LSTM layers and trained with supervised training. Performance evaluations show the model achieve better performance than other machine learning methods when facing variable-length time-domain sequences.

Nonetheless, the above-mentioned approaches assume one single transmitter. None of them can deal with the blind signature classification problem for NOMA where the received signal is superimposed and highly-condensed. The authors in [34] resort to capsule network to solve a multi-signal AMC problem in uplink where multiple transmitters simultaneously transmit signals at adjacent frequencies. However, this method is not applicable in NOMA system, since NOMA signals overlap on exactly the same time and frequency resources. The authors in [20] develop a Anderson-Darling test-based machine learning algorithm to determine the modulation order of the interference signal in downlink NOMA system. The very recent work [21] considers the blind signal classification in a two-user downlink NOMA system. However, the methods in [20] and [21] are limited to the quadrature amplitude modulation (QAM) constellations and downlink power-domain NOMA system, and there still lacks a unified framework for blind signature classification for uplink NOMA.

In this paper, we firstly propose an optimal LB method for NOMA signature classification, and then show that this method achieves perfect classification in the asymptotic limit of the observations of the received signals. To reduce the exponentially increased computational complexity invoked in the likelihood calculations, we consider to employ the FB classification methods. Whereas the efficient features are non-trivial to be manually designed in NOMA case, we resort to deep learning for automatic feature extraction and complexity reduction. Due to the superimposed signal space and diversified fading states among the transmitter, we find that straightforward DNN methods do not cope with this problem. So we adopt a model-driven DNN design approach [35], where the insights gained from the optimal LB method play a vital

role. The detailed contributions of this paper are summarized as follows:

- We formulate the blind signature classification problem for NOMA and propose a likelihood-based method to achieve optimal classification accuracy performance under the perspective of maximum a posteriori estimation. We also prove that, under some reasonable assumptions, the classification error asymptotically goes to zero given enough observations of received NOMA signal blocks.
- We propose a deep learning framework, namely DeepClassifier, which automatically extracts efficient features for low-complexity signature classification. To benefit from the communication-domain expertise, DeepClassifier mimics the LB method by considering two processing stages, i.e., the feed-forward stage and the recurrent stage. In the feed-forward stage, we deal with one single observation by considering an iterative layered DNN structure, where the classification result is revised layer by layer. Within each layer, we propose a memory-extended structure which explicitly embeds a cache module to exploit the prior knowledge of NOMA signature pool.
- We propose a deep recurrent accumulation neural network in the recurrent stage of DeepClassifier framework to fully utilize the sequentially received observations. The immediate classification result as well as the extracted features for all observations are commendably aggregated. Specifically, we construct the straight-through channels in the recurrent stage to eliminate the information loss which exists in conventional RNNs.
- We implement DeepClassifier and evaluate its performance. Experiments show that, the proposed design outperforms some straightforward DNN-based methods and can approach the optimal classification performance with greatly reduced computational complexity compared with the LB method.

The rest of this paper is organized as follows. The NOMA signature classification problem are formulated in Section II. The likelihood-based classification method is proposed and analyzed in Section III. Section IV describes the proposed model-based deep learning method. Section V presents the experiment results. Finally, Section VI concludes this paper.

*Notations:* Normal lower-case, bold lower-case and bold upper-case symbols denote the scalars, vectors and matrices, respectively.  $\mathbb{R}$  and  $\mathbb{C}$  denote the fields of real and complex number, respectively.  $|\cdot|$  denotes the modulus of a scalar or the cardinality of a set, and  $\|\cdot\|_j$  represents the  $\ell_j$ -norm.  $\mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  represents circular symmetric complex Gaussian distribution with mean  $\boldsymbol{\mu}$  and covariance  $\boldsymbol{\Sigma}$ .

## II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a synchronized uplink NOMA system with  $K$  IoT devices and one base station (BS), where these devices are pre-configured by the BS to share one periodically appeared physical resource region [15]. Specifically, the source symbols of the  $K$  devices are first mapped to  $N$ -dimensional complex symbol sequences and then overlapped on  $N$  orthogonal physical resource elements (REs), as shown in Fig. 1. We call the bundle of  $N$  REs as a NOMA transmit block.

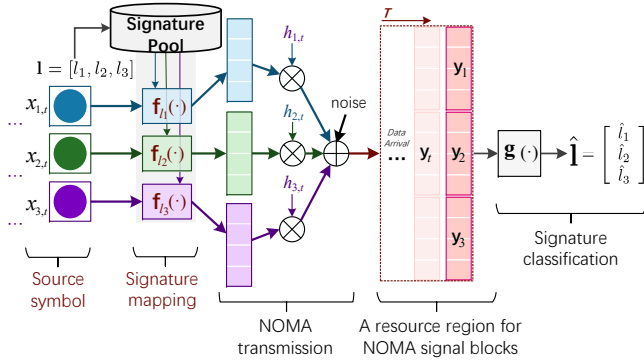


Fig. 1. System model of uplink NOMA with blind signature signature classification in IoT network.

Define the available multiple-access signature pool  $\mathcal{U}$  of the NOMA system consisting of  $J$  signatures as

$$\mathcal{U} = \{\mathbf{f}_1(\cdot), \dots, \mathbf{f}_j(\cdot), \dots, \mathbf{f}_J(\cdot)\}, \quad (1)$$

where each signature, i.e.,  $\mathbf{f}_j(\cdot)$ , refers to a mapping function which maps the input symbol  $x \in \mathcal{X}$  to an  $N$ -dimensional symbol vector and  $\mathcal{X}$  is the input alphabet, with normalized transmit power, i.e.,  $\|\mathbf{f}_j(x)\|_2 = 1, 1 \leq j \leq J, \forall x$ . The index set of  $\mathcal{U}$  is then defined as  $\mathbb{J} = \{1, \dots, j, \dots, J\}$ . To reduce the signaling overhead [15], this paper assumes that each IoT transmitter automatically and uniformly selects a signature out of  $\mathcal{U}$  for uplink transmission, and the BS performs blind signature classification to avoid the explicit notifications. This case is common in NOMA-based IoT network [36], [37], where small data packet is transmitted and tedious signaling interactions become expensive.

Now we describe the signal model. Assume that each physical resource region contains  $T$  NOMA transmit blocks, and the selection of signature remains unchanged within each resource region. For the  $t$ -th block,  $1 \leq t \leq T$ , the channel vector  $\mathbf{h}_t \in \mathbb{C}^K$  of the  $K$  IoT transmitters is defined as

$$\mathbf{h}_t = [h_{1,t}, \dots, h_{k,t}, \dots, h_{K,t}]^\top, \quad (2)$$

where  $h_{k,t}$  is the instantaneous channel coefficient of the  $k$ -th transmitter. While each IoT transmitter can be randomly deployed within the cell, we assume that  $h_{k,t} \sim P_h(h), \forall k, t$ , where  $P_h(h)$  is the probability density function. Further, we consider perfect channel information at the BS via orthogonal reference signal ports [15].

Assume that the  $k$ -th transmitter deploys the  $l_k$ -th signature  $\mathbf{f}_{l_k}(\cdot)$  to transmit source symbol  $x_{k,t} \in \mathcal{X}$ . We denote  $\mathbf{l} = [l_1, \dots, l_k, \dots, l_K] \in \mathbb{J}^K$  as the vector of the indexes of the deployed signatures. The received signal  $\mathbf{y}_t \in \mathbb{C}^N$  at the BS is then given by

$$\mathbf{y}_t = \sum_{k=1}^K h_{k,t} \mathbf{f}_{l_k}(x_{k,t}) + \mathbf{w}_t = \mathbf{h}_t^\top \mathcal{F}_1(\mathbf{x}_t) + \mathbf{w}_t, \quad (3)$$

where  $\mathbf{w}_t \sim \mathcal{CN}(0, \sigma_0^2 \mathbf{I})$  denotes the additive white Gaussian noise (AWGN) with variance  $\sigma_0^2$ , and  $\mathcal{F}_1(\mathbf{x}_t)$  is defined as

$$\mathcal{F}_1(\mathbf{x}_t) = [\mathbf{f}_{l_1}(x_{1,t}), \dots, \mathbf{f}_{l_K}(x_{K,t})]^\top \in \mathbb{C}^{K \times N}, \quad (4)$$

with  $\mathbf{x}_t = [x_{1,t}, \dots, x_{k,t}, \dots, x_{K,t}]$ .

To fully exploit the information conveyed by the entire resource region, signature classification is conducted after receiving the continuous  $T$  observations of the NOMA transmit blocks. We can write the overall received signal for the  $T$  blocks in a compact form, as follows

$$\begin{aligned} \mathbf{Y} &= [\mathbf{y}_1, \dots, \mathbf{y}_t, \dots, \mathbf{y}_T] \\ &= \text{tr}([\mathbf{h}_1, \dots, \mathbf{h}_T]^\top [\mathcal{F}_1(\mathbf{x}_1), \dots, \mathcal{F}_1(\mathbf{x}_T)]) + \mathbf{W} \quad (5) \\ &= \text{tr}(\mathbf{H}^\top \mathcal{F}_1(\mathbf{X})) + \mathbf{W}, \end{aligned}$$

where  $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_T]$ ,  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T]$ ,  $\mathcal{F}_1(\mathbf{X}) = [\mathcal{F}_1(\mathbf{x}_1), \dots, \mathcal{F}_1(\mathbf{x}_T)]$ , and  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_T]$ .

At the BS, a classifier  $\mathbf{g}(\cdot)$  is deployed for signature classification given the channel information and the received signal, as defined by

$$\mathbf{g} : \mathbf{Y}, \mathbf{H} \rightarrow \hat{\mathbf{l}} \in \mathbb{J}^K, \quad (6)$$

where  $\hat{\mathbf{l}} = [\hat{l}_1, \dots, \hat{l}_k, \dots, \hat{l}_K]$  is the estimated index vector. The mis-classification probability can be derived by calculating the differences between  $\hat{\mathbf{l}}$  and the true index vector  $\mathbf{l}$ , and can be formally defined as

$$P_e = \frac{\|\hat{\mathbf{l}} - \mathbf{l}\|_0}{K}, \quad (7)$$

where  $\|\cdot\|_0$  refers to  $\ell_0$ -norm which calculates the number of non-zero elements. In our paper, we aim to design good  $\mathbf{g}(\cdot)$  to minimize  $P_e$ .

### III. LIKELIHOOD-BASED ALGORITHM MODEL AND ASYMPTOTIC ANALYSIS

This section proposes the optimal LB method for NOMA signature classification. Analysis is then conducted to show that the LB method can achieve the perfect classification in the asymptotic limit of the observations. The derived algorithm model provides physical insights into the design of the deep learning method in Section IV.

#### A. Optimal Likelihood-Based Classification Method

Define  $P_{\mathbf{l}|\mathbf{Y}, \mathbf{H}}(\mathbf{l}|\mathbf{Y}, \mathbf{H})$  as the posterior probability distribution of the index vector  $\mathbf{l}$  given the received signal  $\mathbf{Y}$  and channel coefficient  $\mathbf{H}$ . To achieve optimal signature classification under the maximum a posteriori (MAP) perspective, we formulate the classification problem as

$$\begin{aligned} \mathcal{P}1: \hat{\mathbf{l}} &= \underset{\mathbf{l}}{\text{argmax}} \ln [P_{\mathbf{l}|\mathbf{Y}, \mathbf{H}}(\mathbf{l}|\mathbf{Y}, \mathbf{H})], \\ &\text{s.t. } \mathbf{l} \in \mathbb{J}^K, \end{aligned} \quad (8)$$

where  $P_{\mathbf{l}|\mathbf{Y}, \mathbf{H}}(\mathbf{l}|\mathbf{Y}, \mathbf{H})$  is given by

$$\begin{aligned} P_{\mathbf{l}|\mathbf{Y}, \mathbf{H}}(\mathbf{l}|\mathbf{Y}, \mathbf{H}) &= \frac{P_{\mathbf{Y}|\mathbf{l}, \mathbf{H}}(\mathbf{Y}|\mathbf{l}, \mathbf{H}) P_{\mathbf{l}}(\mathbf{l})}{P_{\mathbf{Y}}(\mathbf{Y})} \\ &\stackrel{(a)}{\propto} P_{\mathbf{Y}|\mathbf{l}, \mathbf{H}}(\mathbf{Y}|\mathbf{l}, \mathbf{H}) \\ &= \sum_{\mathbf{X}} P_{\mathbf{Y}|\mathbf{l}, \mathbf{X}, \mathbf{H}}(\mathbf{Y}|\mathbf{l}, \mathbf{X}, \mathbf{H}) P_{\mathbf{X}}(\mathbf{X}) \\ &\propto \mathbb{E}_{\mathbf{X}} \left[ \exp \left( -\frac{1}{\sigma^2} \|\mathbf{Y} - \text{tr}(\mathbf{H}^\top \mathbf{F}_1(\mathbf{X}))\|_{\text{F}}^2 \right) \right] \\ &= \mathbb{E}_{\mathbf{X}} \left[ \exp \left( -\frac{1}{\sigma^2} \sum_{t=1}^T \|\mathbf{y}_t - \mathbf{h}_t^\top \mathcal{F}_1(\mathbf{x}_t)\|_2^2 \right) \right], \end{aligned} \quad (9)$$

where  $P_{\mathbf{1}}(\mathbf{1})$  and  $P_{\mathbf{X}}(\mathbf{X})$  are the prior distributions of the index vector and source symbols, respectively, and  $\|\cdot\|_F$  refers to Frobenius-norm. Here, (a) holds since  $P_{\mathbf{Y}}(\mathbf{Y})$  and  $P_{\mathbf{1}}(\mathbf{1})$  are consistent for all  $\mathbf{1}$ .

The objective function of  $\mathcal{P}1$  is further simplified as follows

$$\begin{aligned}
& \ln [P_{\mathbf{1}|\mathbf{Y},\mathbf{H}}(\mathbf{1}|\mathbf{Y},\mathbf{H})] \\
&= \ln \left[ \mathbb{E}_{\mathbf{X}} \left[ \prod_{t=1}^T \exp \left( -\frac{1}{\sigma^2} \|\mathbf{y}_t - \mathbf{h}_t^\top \mathcal{F}_1(\mathbf{x}_t)\|_2^2 \right) \right] \right] \\
&\stackrel{(a)}{=} \ln \left[ \prod_{t=1}^T \mathbb{E}_{\mathbf{x}_t} \left[ \exp \left( -\frac{1}{\sigma^2} \|\mathbf{y}_t - \mathbf{h}_t^\top \mathcal{F}_1(\mathbf{x}_t)\|_2^2 \right) \right] \right] \\
&= \sum_{t=1}^T \ln \left[ \mathbb{E}_{\mathbf{x}_t} \left[ \exp \left( -\frac{1}{\sigma^2} \|\mathbf{y}_t - \mathbf{h}_t^\top \mathcal{F}_1(\mathbf{x}_t)\|_2^2 \right) \right] \right] \quad (10) \\
&= \sum_{t=1}^T \ln \sum_{\mathbf{x}_t} P_{\mathbf{y}|\mathbf{1},\mathbf{x},\mathbf{h}}(\mathbf{y}_t|\mathbf{1},\mathbf{x}_t,\mathbf{h}_t) P_{\mathbf{x}}(\mathbf{x}_t) \\
&= \sum_{t=1}^T \ln [P_{\mathbf{y}|\mathbf{1},\mathbf{h}}(\mathbf{y}_t|\mathbf{1},\mathbf{h}_t)],
\end{aligned}$$

where (a) holds since the transmission symbols  $\mathbf{x}_t$  and the channel coefficients  $\mathbf{h}_t$  of the  $K$  IoT transmitters are independent and identically distributed (i.i.d.) for each  $t$ . Optimal classification solution to  $\mathcal{P}1$  is obtained by calculating the log-likelihood function (10) for all  $\mathbf{1} \in \mathbb{J}^K$ , and then choose the one  $\hat{\mathbf{1}}$  which corresponds the larger value.

### B. Asymptotic Analysis of Classification Accuracy

In single-user case, using LB methods can ensure perfect classification in a asymptotic limit, i.e., given sufficient observations of the received signals. The following part of this section analyzes whether this conclusion still holds in the NOMA case. Define  $\mathbf{1}^*$  as the true index vector of the selected signatures. Our aim is to prove that the following inequation holds for any  $\mathbf{1} \neq \mathbf{1}^*, \mathbf{1} \in \mathbb{J}^K$ , when  $T \rightarrow +\infty$

$$\ln [P_{\mathbf{1}|\mathbf{Y},\mathbf{H}}(\mathbf{1}^*|\mathbf{Y},\mathbf{H})] > \ln [P_{\mathbf{1}|\mathbf{Y},\mathbf{H}}(\mathbf{1}|\mathbf{Y},\mathbf{H})]. \quad (11)$$

For simplicity of notation, we define a decision variable  $\mathcal{D}_t(\mathbf{y}_t, \mathbf{h}_t, \mathbf{1})$  as

$$\mathcal{D}_t(\mathbf{y}_t, \mathbf{h}_t, \mathbf{1}) = \ln [P_{\mathbf{y}|\mathbf{h},\mathbf{1}}(\mathbf{y}_t|\mathbf{h}_t, \mathbf{1})] \quad (12)$$

and denote  $\mathcal{D}(\mathbf{Y}, \mathbf{H}, \mathbf{1})$  as

$$\mathcal{D}(\mathbf{Y}, \mathbf{H}, \mathbf{1}) = \ln [P_{\mathbf{1}|\mathbf{Y},\mathbf{H}}(\mathbf{1}|\mathbf{Y},\mathbf{H})] = \sum_{t=1}^T \mathcal{D}_t(\mathbf{y}_t, \mathbf{h}_t, \mathbf{1}). \quad (13)$$

Comparing (10), (12) and (13), to prove that (11) holds, it is equivalent to prove

$$\mathcal{D}(\mathbf{Y}, \mathbf{H}, \mathbf{1}^*) > \mathcal{D}(\mathbf{Y}, \mathbf{H}, \mathbf{1}), \forall \mathbf{1} \neq \mathbf{1}^*, \mathbf{1} \in \mathbb{J}^K. \quad (14)$$

First of all, we consider an ideal case with a sufficient high signal to noise ratio (SNR), i.e.,  $\sigma_0 \rightarrow 0$ .

**Definition 1** (Ambiguity): The ambiguity of signature classification is defined as the case that, there exists a non-identical mapping  $\pi: \mathbf{1}^* \rightarrow \mathbf{1}' \in \{1, 2, \dots, J\}^K$ , such that

$$\mathbf{h}_t^\top \mathcal{F}_{\mathbf{1}^*}(\mathbf{x}_t) - \mathbf{h}_t^\top \mathcal{F}_{\mathbf{1}'}(\mathbf{x}_t) = 0, \quad (15)$$

where  $\mathbf{x}_t, \mathbf{x}_t' \in \mathcal{X}^K$ .  $\square$

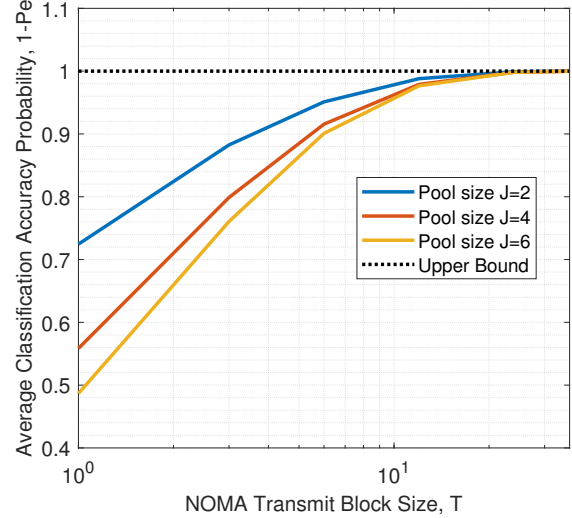


Fig. 2. Performance of NOMA LB signature classification method with  $K = 2$  and  $J = 2, 4, 6$ .

If ambiguity happens, the receiver cannot identify which among  $\mathbf{1}^*$  or  $\mathbf{1}'$  is the true situation. This leads to misclassification. To enable asymptotic analysis, two basic and reasonable assumptions are considered in the following:

**Assumption 1:**  $P_h(h)$  is not a discrete distribution.  $\square$

Assumption 1 normally holds for realistic fading channel models and ensures that the probability of  $h_{k,t} \neq h_{m,t}$  equals to zeros. This avoids the ambiguity of signature classification in a statistical sense.

**Assumption 2:** For  $x, s \in \mathcal{X}$ ,  $\mathbf{f}_j(x) \neq \mathbf{f}_c(s)$  if either  $j \neq c$  or  $x \neq s$ .  $\square$

Assumption 2 is natural, since the opposite will lead to the ambiguity of the source symbols.

In the following Lemma 1, we prove that the ambiguity happens with probability zero, under the aforementioned assumption.

**Lemma 1:** Assume that  $\mathbf{1}^*$  is true. The classification error goes to zero given a sufficient high signal to noise ratio (SNR), i.e.,

$$\lim_{\sigma \rightarrow 0} \Pr [P_{\mathbf{1}|\mathbf{Y},\mathbf{H}}(\mathbf{1}^*|\mathbf{Y},\mathbf{H}) > P_{\mathbf{1}|\mathbf{Y},\mathbf{H}}(\mathbf{1}|\mathbf{Y},\mathbf{H})] = 1, \forall \mathbf{1} \neq \mathbf{1}^*. \quad (16)$$

*Proof:* Please refer to Appendix A.  $\blacksquare$

**Theorem 1** (Asymptotic performance): Assume that  $\mathbf{1}^*$  is true. For any  $\sigma_0$ , the classification error goes to zero when the number of the received signal blocks goes to infinity, i.e.,

$$\lim_{T \rightarrow \infty} \Pr [P_{\mathbf{1}|\mathbf{Y},\mathbf{H}}(\mathbf{1}^*|\mathbf{Y},\mathbf{H}) > P_{\mathbf{1}|\mathbf{Y},\mathbf{H}}(\mathbf{1}|\mathbf{Y},\mathbf{H})] = 1, \forall \mathbf{1} \neq \mathbf{1}^*. \quad (17)$$

*Proof:* Please refer to Appendix B.  $\blacksquare$

Compared with the asymptotic analysis for the single-user case [23], [24], the above two assumptions are required in the NOMA case to ensure perfect classification in the asymptotic limit of the observations. We demonstrate the correctness of Theorem 1 in Fig. 2. Here we consider a NOMA system with  $K = 2$  transmitters and  $J = 2, 4, 6$  available signatures. Rayleigh fading channel model is considered for both trans-

mitters. As show in Fig.2, enlarging the value of  $T$  provides better classification accuracy, which finally converges to 1.

Now we discuss the computational complexity involved in the optimal LB method. For the  $t$ -th NOMA block, the proposed method requires to traverse the entire search space, i.e.,  $\mathbb{J}^K$ , which leads to  $J^K$  times of calculations. Besides, each calculation requires to provide an expectation over  $\mathcal{X}^K$ , i.e.,  $\mathbb{E}_{\mathbf{x}_t}[\cdot]$ , which leads to another  $|\mathcal{X}|^K$  times of calculations. Therefore, the computational complexity for this LB method is in the order of  $\mathcal{O}(c_{\text{LB}} T J^K |\mathcal{X}|^K)$ , where  $c_{\text{LB}}$  is number of the calculations required for calculating the likelihood function of each  $\mathbf{l} \in \mathbb{J}^K$  [29]. Although solving  $\mathcal{P}1$  ensures optimal classification performance, large computational efforts are required due to the finite-alphabet search-space  $\mathbb{J}^K$  and the expectation over all possible  $\mathbf{x}_t$  in (10), which is inefficient for future IoT system.

As an alternative approach, FB classification method has been considered in the conventional modulation classification problems for its low computational complexity compared with LB method [25]. These researches normally explore the moments of the statistical distributions of the received signal and manually design the features for different modulation schemes. However, these approaches meet the obstacle in NOMA case because of the superimposed received signals and different fading states of the transmitters. We consider a 6-user NOMA system with 6 signatures, which leads to  $6^6 = 46656$  combinatorial possibilities and is thus non-trivial to manually design the features for NOMA signature classification. Fortunately, the recent booming of DNN provides a new methodology to automatically extract efficient features in a data-driven fashion. The following section will develop a deep learning framework for efficient signature classification.

#### IV. DEEPClassifier: A MODEL-BASED SIGNATURE CLASSIFICATION FRAMEWORK VIA DNN

The universal approximation property of DNN ensures it can perfectly approximate any measurable function with any desired degree of accuracy [39]. We regard the likelihood decision variable  $\mathcal{D}(\mathbf{Y}, \mathbf{H}, \mathbf{l})$  as the function to be approximated and generalize the universal approximation property into signature classification. Therefore, a DNN classifier with enough size can achieve the same performance as the optimal LB method, which is formally expressed in the following **Theorem 2**.

**Theorem 2** (Universal approximation property): The decision variable  $\mathcal{D}(\mathbf{Y}, \mathbf{H}, \mathbf{l})$  in (13), can be accurately approximated by a DNN in the following sense:  $\forall \epsilon > 0$ , there exists a DNN, denoted as  $\mathcal{D}^{\text{DNN}}(\mathbf{Y}, \mathbf{H}, \mathbf{l})$ , such that

$$|\mathcal{D}(\mathbf{Y}, \mathbf{H}, \mathbf{l}) - \mathcal{D}^{\text{DNN}}(\mathbf{Y}, \mathbf{H}, \mathbf{l})| < \epsilon. \quad (18)$$

*Proof:* The proof is straightforward following the universal function approximation property of DNN [39]. ■

Although **Theorem 2** indicates the existence of an optimal DNN, the algorithmic learnability is not ensured. A straightforward DNN with a practical network size performs way worse than the LB method, as we will show in Section V. Existing researches have shown that model-driven design can exploit the communication-domain expertise and thus enhance the

learning efficiency [35]. So this section will firstly gain design insights from the LB algorithmic model and then develop a DNN for efficient signature classification, which will be named from now on, *DeepClassifier*. The objective of DeepClassifier is to extract the likelihood function features and obtain the approximate optimal classification result based on maximum likelihood method.

#### A. Design Insights

Recall the LB method proposed in Section III-A which consists of two processing stages. In the first stage, the immediate decision variable  $\mathcal{D}_t$  is calculated based on  $\mathbf{y}_t$  for each  $t$ . Then in the second stage,  $\mathcal{D}_t$ s are aggregated to obtain the final decision. Correspondingly, DeepClassifier will also consist of two stages, i.e., a feed-forward stage and a recurrent stage, to mimic the algorithmic model of the LB method. The first stage recasts the iterative structure described in (20) to obtain immediate features for each NOMA block. Then the second stage aggregates the features of multiple signal blocks to obtain final classification results by taking the advantage of a deep recurrent structure.

Now we focus on the first stage. For the  $t$ -th NOMA block, we firstly simplify the expectation over  $\mathbf{X}$  in (10) according to Max-Log-MAP algorithm as follows

$$\begin{aligned} & \ln [P_{\mathbf{y}|\mathbf{l},\mathbf{h}}(\mathbf{y}_t|\mathbf{l},\mathbf{h}_t)] \\ &= \ln \left[ \mathbb{E}_{\mathbf{x}_t} \left[ \exp \left( -\frac{1}{\sigma^2} \|\mathbf{y}_t - \mathbf{h}_t^\top \mathcal{F}_1(\mathbf{x}_t)\|_2^2 \right) \right] \right] \quad (19) \\ &\approx \max_{\mathbf{x}_t} \left[ -\|\mathbf{y}_t - \mathbf{h}_t^\top \mathcal{F}_1(\mathbf{x}_t)\|_2^2 \right]. \end{aligned}$$

A general communication-theoretic paradigm to solve NP-hard problem is to develop an iterative framework [38]. In this way, we can formulate the classification result of the  $t$ -th block in the  $(i+1)$ -th iteration as  $\hat{\mathbf{l}}_t^{(i+1)}$

$$\hat{\mathbf{l}}_t^{(i+1)} = \Pi \left[ \hat{\mathbf{l}}_t^{(i)} - \delta^{(i)} \Delta \hat{\mathbf{l}}_t^{(i)} \right], \quad (20)$$

where  $\hat{\mathbf{l}}_t^{(i)}$  is the estimated result for the  $i$ -th iteration,  $\Delta \hat{\mathbf{l}}_t^{(i)}$  is the adjustment value on  $\hat{\mathbf{l}}_t^{(i)}$  and  $\delta_t^{(i)}$  is the step size for the adjustment.  $\Pi[\cdot]$  refers to a non-linear projection operator to ensure that  $\hat{\mathbf{l}}_t^{(i+1)}$  holds effective values. The gradient descent method is normally used to generate  $\Delta \hat{\mathbf{l}}_t^{(i)}$  for each iteration

$$\begin{aligned} \Delta \hat{\mathbf{l}}_t^{(i)} &\approx \max_{\mathbf{x}_t} \left[ \frac{\partial \|\mathbf{y}_t - \mathbf{h}_t^\top \mathcal{F}_1(\mathbf{x}_t)\|_2^2}{\partial \mathbf{l}} \Big|_{\hat{\mathbf{l}}_t^{(i)}} \right] \\ &\propto \max_{\mathbf{x}_t} \left[ \frac{\partial \mathcal{F}_1^\top(\mathbf{x}_t)}{\partial \mathbf{l}} \mathbf{h}_t (\mathcal{F}_1^\top(\mathbf{x}_t) \mathbf{h}_t - \mathbf{y}_t^\top) \Big|_{\hat{\mathbf{l}}_t^{(i)}} \right]. \quad (21) \end{aligned}$$

With conventional mathematical tools, it is neither easy to find the optimal  $\mathbf{x}_t$  in (19), nor is it easy to derive the gradients in (21). With deep learning, however, this iterative structure becomes feasible since the operations in each iteration can be automatically learned by a DNN, as in our proposed design of the feed-forward stage.

After obtaining the estimated results for each NOMA block, the second stage then aggregates them to derive the final classification results according to (14). The recurrent stage of DeepClassifier is designed to accomplish the same goal.



For the ease of discussion, in this section, we encode each index  $l_k \in \mathbb{J}$  into the one-hot representation  $\mathbf{l}_k = [l_{k,1}, \dots, l_{k,j}, \dots, l_{k,J}] \in \{0, 1\}^J$ , where  $l_{k,j}$  is given by

$$l_{k,j} = \begin{cases} 1, & \text{if } j = l_k, \\ 0, & \text{otherwise.} \end{cases} \quad (22)$$

The representation  $\mathbf{l}_k$  is used as the label for the  $k$ -th IoT transmitter. We also note that, the following part uses the complex domain to describe the network model for simplicity of notation, while each complex value can be represented by its real and image parts during implementation. The following parts elaborate the designs of feed-forward and the recurrent stages of DeepClassifier, respectively.

### B. Feed-Forward Stage of DeepClassifier

We derive two hints from Section IV-A to design the feed-forward stage of DeepClassifier:

- First of all, an iterative structure shall be considered where each iteration provides a rough classification result, and this result is then utilized as the side information in the next iteration to achieve more accurate classification, as illustrated in (19).
- Secondly, the main ingredients shall include not only the received signal  $\mathbf{Y}_t$  and the channel coefficient  $\mathbf{h}_t$ , but also the knowledge of NOMA signature pool, i.e.,  $\mathcal{U}$ , as illustrated in (20).

The first hint leads to an iterative layered network structure, as elaborated in Section IV-B1, which is similar to Learned-AMP [38] or DetNet [40]. The second hint stimulates us to explicitly include the signature pool within the network, which, however, has not been exploited before according to our best knowledge. Inspired by the recently proposed neural Turing machine (NTM) [41], we explicitly construct a cache module within the feed-forward stage, such that each iteration can take advantage the knowledge of the signature pool, as detailed in Section IV-B2.

1) *Iterative Layered Network Structure*: The feed-forward stage of the proposed DeepClassifier consists of  $L$  layers to mimic the iterative framework as described in Section IV-A. We employ fully connected DNN (FC-DNN) to construct the iterative layered network to process the low-dimensional raw signal data, which are generated by the channel coefficients and the received NOMA signals. During propagation, the classification results of previous layers are sent to the next layer to iteratively improve the classification accuracy. The  $(i+1)$ -th layer is depicted in Fig. 3 and detailed as follows.

Before training, in the  $t$ -th NOMA block, the channel coefficients  $\mathbf{h}_t = [h_t^1, h_t^2, \dots, h_t^K] \in \mathbb{C}^K$  and the received signals  $\mathbf{y}_t = [y_t^1, y_t^2, \dots, y_t^N] \in \mathbb{C}^N$  of all users are concatenated into one vector. Then the complex vector is decomposed into the real part and the imaginary part to generate the in-phase and quadrature (IQ) data, which is denoted as  $\mathbf{I}_t = [\text{Re}\{\mathbf{h}_t\}, \text{Im}\{\mathbf{h}_t\}, \text{Re}\{\mathbf{y}_t\}, \text{Im}\{\mathbf{y}_t\}] \in \mathbb{R}^{2(K+N)}$ . Our proposed layered DNN then processes  $\mathbf{I}_t$  to perform automatic feature extraction and signal classification for the  $t$ -th NOMA block. After the processing of the  $i$ -th layer, we denote the vector  $\hat{\mathbf{a}}_{k,t}^{(i)} \in [0, 1]^J$ ,  $1 \leq k \leq K$ , as the obtained classification result for the  $k$ -th user, where  $\hat{\mathbf{a}}_{k,t}^{(i)}(j)$ ,  $1 \leq j \leq J$ , is the  $j$ -th

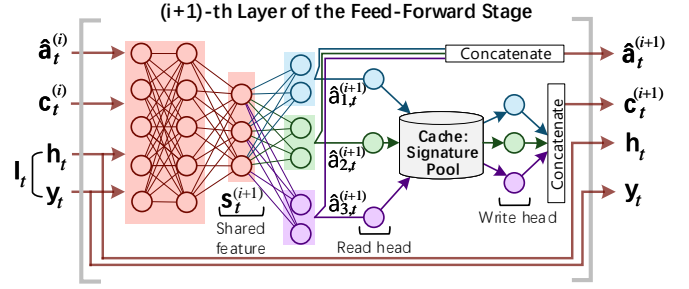


Fig. 3. Flow diagram of the  $(i+1)$ -th layer of the feed-forward stage of DeepClassifier, taking the three-user NOMA system as an example.

element of  $\hat{\mathbf{a}}_{k,t}^{(i)}$  representing the estimated probability of the  $k$ -th user using the  $j$ -th signature, with  $\sum_{j=1}^J \hat{\mathbf{a}}_{k,t}^{(i)}(j) = 1$ . We denote  $\hat{\mathbf{a}}_t^{(i)} = [\hat{\mathbf{a}}_{1,t}^{(i)}, \dots, \hat{\mathbf{a}}_{k,t}^{(i)}, \dots, \hat{\mathbf{a}}_{K,t}^{(i)}]$  as the collection of the classification results for  $K$  users in the  $i$ -th layer. Besides, we denote  $\hat{\mathbf{c}}_t^{(i)} = [\hat{\mathbf{c}}_{1,t}^{(i)}, \dots, \hat{\mathbf{c}}_{k,t}^{(i)}, \dots, \hat{\mathbf{c}}_{K,t}^{(i)}]$  as the collection of the outputs of the cache module in the  $i$ -th layer, where  $\hat{\mathbf{c}}_{k,t}^{(i)}$  corresponds to the  $k$ -th transmitter. The  $(i+1)$ -th layer then takes  $\mathbf{I}_t$ ,  $\hat{\mathbf{a}}_t^{(i)}$  and  $\hat{\mathbf{c}}_t^{(i)}$  as the input to extract the shared feature  $\mathbf{s}_t^{(i+1)}$  via a feed-forward DNN  $\mathbf{g}_{\text{Pub}}^{\text{FF-(i)}}(\cdot)$ , as follows

$$\mathbf{s}_t^{(i+1)} = \mathbf{g}_{\text{Pub}}^{\text{FF-(i+1)}}\left(\mathbf{I}_t, \hat{\mathbf{c}}_t^{(i)}, \hat{\mathbf{a}}_t^{(i)}; \mathbf{W}_{\text{Pub}}\right), \quad (23)$$

where  $\mathbf{W}_{\text{Pub}}$  corresponds to all trainable parameters related to this network. Note that, the superscript “FF” refers to feed-forward DNN and the subscript “Pub” refers to the fact that this network is a public part for  $K$  classification tasks. Note that, the original inputs  $\mathbf{y}_t$  and  $\mathbf{h}_t$  are fed to all layers, inspired by the shortcut connections introduced in ResNet [42].

Normally,  $\mathbf{g}_{\text{Pub}}^{\text{FF-(i+1)}}(\cdot)$  is a DNN constituted by multiple nested non-linear transformation, where each transformation maps  $\mathbf{In}$  to  $\mathbf{Out}$  as follows

$$\mathbf{Out} = \sigma(\mathbf{wIn} + \mathbf{b}), \quad (24)$$

where  $\mathbf{w}$  and  $\mathbf{b}$  denote the parameter matrix and the bias associated with this DNN, respectively. The mapping  $\sigma(\cdot)$  corresponds to the activation function of this DNN, which can be either identity, sigmoid, tanh, softmax or rectified linear unit (ReLU) functions. In the rest of the paper, without ambiguity, we will omit the notation of the parameter set related to all DNNs for simplicity.

With the shared feature  $\mathbf{s}_t^{(i+1)}$ , we then deploy  $K$  private DNNs to obtain immediate classification results  $\mathbf{a}_{k,t}^{(i+1)}$ , where its  $j$ -th element is the probability that the hypothesis of the  $j$ -th signature holds true. For the  $k$ -th transmitter, a DNN, denoted as  $\mathbf{g}_{\text{Pri-}k}^{\text{FF-(i+1)}}(\cdot)$ , is employed to extract  $\mathbf{a}_{k,t}^{(i+1)}$  from  $\mathbf{s}_t^{(i+1)}$

$$\mathbf{a}_{k,t}^{(i+1)} = \mathbf{g}_{\text{Pri-}k}^{\text{FF-(i+1)}}\left(\mathbf{s}_t^{(i+1)}\right), \quad (25)$$

where the subscript “Pri- $k$ ” reflects that  $\mathbf{g}_{\text{Pri-}k}^{\text{FF}}(\cdot)$  is a private network of the  $k$ -th transmitter. Note that, a softmax layer shall be used as the output layer to ensure that each element of  $\mathbf{a}_{k,t}$  is greater than zero and the sum of all elements equals to one.

2) *Memory-Extended Structure*: To explicitly utilize the knowledge of NOMA signature pool, we employ a memory-extended structure by introducing a cache module which contains all the elements of the signature pool.

Define  $\mathfrak{F} \in \mathbb{C}^{J|\mathcal{X}| \times J|\mathcal{X}|}$  as all complex values that can be generated by the signature pool  $\mathcal{U}$  and source symbol alphabet  $\mathcal{X}$ . The vector  $\mathfrak{F}$  thus contains the entire knowledge related to encoding procedure at the IoT transmitters. Therefore, for the  $k$ -th transmitter, the cache module takes  $\hat{\mathbf{a}}_{k,t}^{(i+1)}$  as the input through a *read head*, denoted as  $\mathfrak{W}_{\text{Re}} \in \mathbb{C}^{J|\mathcal{X}| \times J}$ , and then generate  $\hat{\mathbf{c}}_{k,t}^{(i+1)}$  through a *write head*, denoted as  $\mathfrak{W}_{\text{Wr}} \in \mathbb{C}^{N \times J|\mathcal{X}|}$ . This procedure can be represented by

$$\begin{aligned} \mathbf{c}_{k,t}^{(i+1)} &= \mathbf{g}_{\text{Cache-}k}^{\text{FF-}(i+1)} \left( \hat{\mathbf{a}}_{k,t}^{(i+1)} | \mathcal{F}_1(\cdot) \right) \\ &= \sigma \left( \mathfrak{W}_{\text{Wr}} \mathfrak{F} \mathfrak{W}_{\text{Re}} \mathbf{a}_{k,t} \right), \end{aligned} \quad (26)$$

where  $\mathfrak{W}_{\text{Wr}}$  and  $\mathfrak{W}_{\text{Re}}$  are trained to learn what to read and write, respectively, according to  $\mathbf{a}_{k,t}$ . Nonetheless, the size of the vector  $\mathfrak{F}$  is large, which may leads to large read/write heads. In a special case where the linear spreading-based NOMA scheme is considered to achieve low implementation complexity [15], [44], [45], we note that, the cache module can take a very simple structure. With the assumption of linear spreading signature,  $\mathbf{y}_t - \mathbf{h}_t^\top \mathcal{F}_1(\mathbf{x}_t)$  can be written as

$$\mathbf{y}_t - \mathbf{h}_t^\top \mathcal{F}_1(\mathbf{x}_t) = \mathbf{y}_t - \sum_{s=1}^K h_{s,t} x_{s,t} \mathbf{F} \mathbf{I}_s, \quad (27)$$

where  $\mathbf{F} \in \mathbb{C}^{N \times J}$  is the collection of the  $J$  linear spreading signatures with length  $N$ . Therefore, we can calculate the gradient with respect to  $\mathbf{l}_k$  using  $\hat{\mathbf{a}}_{k,t}^{(i+1)}$  as follows

$$\begin{aligned} & \max_{\mathbf{x}_t} \left[ \frac{\partial \left\| \mathbf{y}_t - \sum_{s=1}^K h_{s,t} x_{s,t} \mathbf{F} \mathbf{I}_s \right\|_2^2}{\partial \mathbf{l}_k} \right]_{\mathbf{l}_k = \hat{\mathbf{a}}_{k,t}^{(i)}} \\ &= \max_{\mathbf{x}_t} \left[ (h_{k,t} x_{k,t} \mathbf{F})^\top \left( \sum_{s=1}^K h_{s,t} x_{s,t} \mathbf{F} \mathbf{I}_s - \mathbf{y}_t \right) \right]_{\mathbf{l}_k = \hat{\mathbf{a}}_{k,t}^{(i)}} \\ &\stackrel{(a)}{\propto} \max_{x_{k,t}} \left[ (h_{k,t} x_{k,t} \mathbf{F})^\top h_{k,t} x_{k,t} \mathbf{F} \mathbf{l}_k \right]_{\mathbf{l}_k = \hat{\mathbf{a}}_{k,t}^{(i)}} \\ &\stackrel{(b)}{=} (h_{k,t} \mathbf{F})^\top h_{k,t} \mathbf{F} \hat{\mathbf{a}}_{k,t}^{(i)}, \end{aligned} \quad (28)$$

where in step-(a) we only focus on the terms related to the  $k$ -th transmitter, and step-(b) is due to  $\|x_{k,t}\|^2 = 1, \forall k, t$ . According to (28), the gradient is approximately irrelevant to the choice of  $x_{k,t}$ . As a result, the output of cache module is derived as

$$\mathbf{c}_{k,t}^{(i+1)} = \sigma \left( \mathbf{W}_{\text{Wr}} \mathbf{F} \mathbf{W}_{\text{Re}} \mathbf{a}_{k,t} \right) \quad (29)$$

where  $\mathbf{W}_{\text{Wr}} \in \mathbb{C}^{J \times J}$  and  $\mathbf{W}_{\text{Re}} \in \mathbb{C}^{J \times J}$  are the read and write heads, respectively, and they are initialized as identical matrix to cope with the formulation of (28).

For simplicity of notation, we write the mapping function  $\mathbf{g}^{\text{FF-}(i+1)}(\cdot)$  of the  $(i+1)$ -th layer as

$$\left[ \hat{\mathbf{c}}_t^{(i+1)}, \hat{\mathbf{a}}_t^{(i+1)} \right] = \mathbf{g}^{\text{FF-}(i+1)} \left( \mathbf{y}_t, \mathbf{h}_t, \hat{\mathbf{c}}_t^{(i)}, \hat{\mathbf{a}}_t^{(i)} \right). \quad (30)$$

Assume that the feed-forward stage consists of  $L$  layer. The final classification results are then denoted as  $\hat{\mathbf{a}}_t^{(L)}$ .

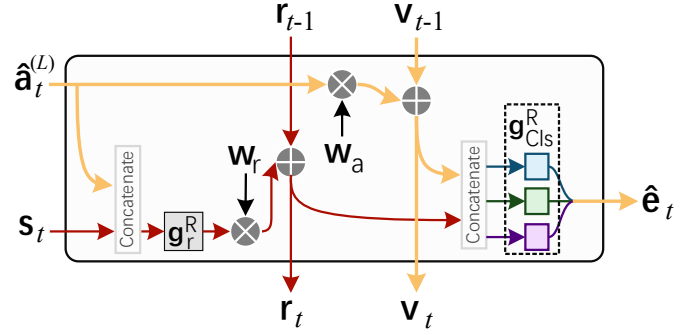


Fig. 4. Detailed network structure of the proposed RANN of DeepClassifier.

### C. Recurrent Stage of DeepClassifier

If we obtain optimal decision variables in the feed-forward stage for all NOMA blocks, then a simple accumulation of these variables can lead to the optimal classification, according to (13). Nonetheless, a limited network size is normally considered in DeepClassifier to control the implementation complexity, so the bias between true decision variables and the obtained ones is inevitable. In this case, using a simple accumulation of  $\hat{\mathbf{a}}_t^{(L)}$ s,  $\forall t$ , may result in the error accumulation. To overcome this obstacle, a deep recurrent structure shall be considered to take the advantage of the sequentially extracted features  $s_t$ s,  $\forall t$ , for classification accuracy enhancement.

RNN is a well-known recurrent network which aims to deal with sequential inputs. In the inference procedure of RNN, the outputs will largely depend on more recent inputs and the effect of the inputs in previous time stamps tends to fade away as time goes by. Its variant LSTM also encounters the similar problem owing to the non-linear accumulation. According to (13), for the signature classification, each NOMA sample should have equal influence on the final decision. Therefore, RNN and LSTM can not meet the hypothesis of the signature classification.

To cope with our problem, we propose a specially designed recurrent structure, namely deep recurrent accumulation neural network (RANN), as illustrated in Fig. 4. The theoretical hypothesis of RANN model originates from the fact that each received sample contributes equally to the final classification decision. The core idea is to construct two straight-through channels to run straight down the entire chains, corresponding to the extracted features and the estimated classification results, denoted as  $\mathbf{r}_{t-1}$  and  $\mathbf{v}_{t-1}$ , respectively. The straight-through channels help avoid the information loss existed in RNN and LSTM. By taking the state  $\mathbf{r}_{t-1}$  and  $\mathbf{v}_{t-1}$  generated in the previous time stamp, the input features  $s_t^{(L)}$  and the immediate classification result  $\hat{\mathbf{a}}_t^{(L)}$  as input, the proposed RANN, denoted as  $\mathbf{g}^{\text{R}}(\cdot)$ , generates the new state vectors  $\mathbf{r}_t$  and  $\mathbf{v}_t$  as well as the instantaneous output  $\hat{\mathbf{e}}_t$ , as follows

$$[\mathbf{r}_t, \mathbf{v}_t, \hat{\mathbf{e}}_t] = \mathbf{g}^{\text{R}} \left( \mathbf{r}_{t-1}, \mathbf{v}_{t-1}, s_t^{(L)}, \hat{\mathbf{a}}_t^{(L)} \right), \quad (31)$$

where the superscript ‘‘R’’ refers to RANN, and  $\hat{\mathbf{e}}_t = [\hat{\mathbf{e}}_{1,t}, \dots, \hat{\mathbf{e}}_{k,t}, \dots, \hat{\mathbf{e}}_{K,t}]$  is the collection of all classification results after this time stamp with  $\hat{\mathbf{e}}_{k,t} \in [0, 1]^J$ . Specifically,

---

**Algorithm 1** Training Algorithm of DeepClassifier
 

---

**Require:** Dataset  $\mathfrak{D}$ .

**Ensure:** DNNs  $\mathbf{g}^{\text{FF}(i)}(\cdot)$ ,  $1 \leq i \leq L$ , and  $\mathbf{g}^{\text{R}}(\cdot)$ .

- 1: **repeat**
  - 2: Draw a mini-batch  $\mathfrak{D}'$  out of  $\mathfrak{D}$
  - 3:  $\mathbf{g}^{\text{FF}(i)}(\cdot), \forall i \leftarrow \text{SGD}(\mathcal{L}_{\text{FF}}, \mathfrak{D}', \mathbf{g}^{\text{FF}(i)}(\cdot), \forall i)$
  - 4: **until** Stop criterion is met
  - 5: **repeat**
  - 6: Draw a mini-batch  $\mathfrak{D}'$  out of  $\mathfrak{D}$
  - 7:  $\mathbf{g}^{\text{R}}(\cdot), \mathbf{g}^{\text{FF}(i)}(\cdot), \forall i \leftarrow \text{SGD}(\mathcal{L}_{\text{R}}, \mathfrak{D}', \mathbf{g}^{\text{R}}(\cdot), \mathbf{g}^{\text{FF}(i)}(\cdot), \forall i)$
  - 8: **until** Stop criterion is met sphere decoding
- 

 $\mathbf{r}_t$ ,  $\mathbf{v}_t$  and  $\hat{\mathbf{e}}_t$  are respectively given by

$$\begin{cases} \mathbf{r}_t = \mathbf{r}_{t-1} + \mathbf{W}_r \cdot \mathbf{g}_r^{\text{R}}(\hat{\mathbf{a}}_t^{(L)}, \mathbf{s}_t), \\ \mathbf{v}_t = \mathbf{v}_{t-1} + \mathbf{W}_a \cdot \hat{\mathbf{a}}_t^{(L)}, \\ \hat{\mathbf{e}}_t = \mathbf{g}_{\text{Cls}}^{\text{R}}(\mathbf{r}_t, \mathbf{v}_t), \end{cases} \quad (32)$$

where  $\mathbf{g}_r^{\text{R}}(\cdot)$  denotes a DNN used to aggregate the input features,  $\mathbf{W}_r$  and  $\mathbf{W}_a$  are the weight parameters for two straight-through channels, and  $\mathbf{g}_r^{\text{R}}(\cdot)$  denotes the classification layer used to generate classification results. Different from LSTM which utilizes the instant inputs and non-linear gates to control the long-term memory in straight-through channels, RANN uses the input-independent weights for accumulation, i.e.,  $\mathbf{W}_r$  and  $\mathbf{W}_v$ , based on the intuition of (13). Therefore, RANN requires far less number of trainable parameters.

Fig. 5 illustrates the overall architecture of the proposed DeepClassifier. The feed-forward stage constructs an iterative layered structure to process the classification of the single NOMA block. Each iteration provides a rough classification result as the side information in the next iteration. The network parameters of different blocks can be shared. After  $L$  iterations, the feed-forward network outputs the classification result  $\hat{\mathbf{a}}_t$  and the corresponding feature  $\mathbf{s}_t$  of the  $t$ -th NOMA block. Then RANN aggregates these results and features by the straight-through channels, which can equally utilize the observations of all blocks. At the end of the  $T$  NOMA transmission blocks, we obtain  $\hat{\mathbf{e}}_T$  as the final estimation of the signature classification results.

#### D. Dataset, Loss Function and Training Algorithm

We generate a synthetic dataset  $\mathfrak{D}$  to train DeepClassifier

$$\mathfrak{D} = \left\{ (\mathbf{Y}, \mathbf{H}, \mathbf{l}_1, \dots, \mathbf{l}_k, \dots, \mathbf{l}_K)^{(m)} \right\}_{m=1}^M, \quad (33)$$

where  $(\mathbf{Y}, \mathbf{H}, \mathbf{l}_1, \dots, \mathbf{l}_k, \dots, \mathbf{l}_K)^{(m)}$  is the  $m$ -th instance in the dataset, and  $M$  is the number of instances in the dataset. For each instance,  $\mathbf{l}_k$ s,  $\forall k$ , are firstly randomly generated according to uniform distribution, then  $\mathbf{H}$  is randomly generated according to a predefined channel model, and finally  $\mathbf{Y}$  is generated according to (5).

Now we analyze the loss functions related to DeepClassifier. For the feed-forward stage, since we are dealing with a typical classification problem, it is natural to deploy cross-entropy (CE) loss. The CE loss between  $\hat{\mathbf{a}}_{k,t}^{(i)}$  and  $\mathbf{l}_k$  in the  $i$ -th layer is defined as

$$\mathcal{L}_{\text{FF-}k,t}^{(i)} = \mathbf{l}_k^\top \log(\hat{\mathbf{a}}_{k,t}^{(i)}), \quad (34)$$

where FF- $k$  refers to the CE loss of the  $k$ -th classification task of the feed-forward stage. The total loss, denoted as  $\mathcal{L}_{\text{FF}}$ , related to the feed-forward stage is then given by

$$\mathcal{L}_{\text{FF}} = \sum_{i=1}^L w_i \sum_{t=1}^T \sum_{k=1}^K \mathcal{L}_{\text{FF-}k,t}^{(i)}, \quad (35)$$

where  $w_i$ ,  $1 \leq i \leq L$ , is the weight coefficient of  $L$  layers. Similarly, for the recurrent stage, the CE loss between  $\hat{\mathbf{e}}_{k,T}$  and  $\mathbf{l}_k$  is given by

$$\mathcal{L}_{\text{R-}k} = \mathbf{l}_k^\top \log(\hat{\mathbf{e}}_{k,T}). \quad (36)$$

Since we only use the output of the final time stamp in the recurrent stage, so the total loss, denoted as  $\mathcal{L}_{\text{R}}$ , is given by

$$\mathcal{L}_{\text{R}} = \sum_{k=1}^K \mathcal{L}_{\text{R-}k}. \quad (37)$$

The CE loss measures the difference between the true index and the estimated index, so DeepClassifier shall be trained to minimize the CE loss. The pseudocode of the entire training process is shown in Algorithm 1. In the first step, we train the feed-forward stage, i.e.,  $\mathbf{g}^{\text{FF}(i)}(\cdot)$ ,  $1 \leq i \leq L$ . The second step takes the trained parameters as a good initialization to fine-tune the recurrent stage, i.e.,  $\mathbf{g}^{\text{R}}(\cdot)$ . For each step, a mini-batch of data is randomly drawn out of the dataset. The optimization step is then performed using stochastic gradient descent (SGD) or a variant [43], by taking the loss function, the mini-batch and the DNN to be trained as the inputs.

#### E. List Detection-Aided DeepClassifier

To approach near-optimal performance, we further consider a list detection-aided DeepClassifier method. The output of the recurrent stage is first used to generate a list of candidate hypotheses, and then an exhaustive search is conducted to determine the most possible hypothesis.

Given the final classification output  $\hat{\mathbf{e}}_T$ , the DeepClassifier-based list detection aims to solve the following problem

$$\begin{aligned} \text{P2: } \hat{\mathbf{l}}_k, \forall k &= \underset{\mathbf{l}_k, \forall k}{\text{argmax}} \ln [P_{\mathbf{l}_k, \forall k | \mathbf{Y}, \mathbf{H}}(\mathbf{l}_k, \forall k | \mathbf{Y}, \mathbf{H})], \\ \text{s.t., (a)} & \mathbf{l} \in \mathbb{J}^K, \\ \text{(b)} & \mathbf{l} \in \{\hat{\mathbf{e}}_{k,T}\}_{\max, D_0}, D_0 \in \mathbb{N}^+, \end{aligned} \quad (38)$$

and (c)  $\mathbf{l}_k$  are generated by  $\mathbf{l}$  according to (22),

where  $D_0$  denotes the search radius of a hyper-sphere,  $\{\hat{\mathbf{e}}_{k,T}\}_{\max, D_0}$  denotes the indexes of  $D_0$ -largest elements in  $\hat{\mathbf{e}}_{k,T}$ , and thus (38b) determines the list of hypotheses to be searched. Note that, since  $D_0$  is smaller than  $J$ , hence  $D_0^K$  would be much smaller than  $J^K$ . Therefore, using the output of DeepClassifier as search center can greatly reduce the computational complexity compared with the LB method.

Finally, we analyze the computational complexity order of the proposed DeepClassifier. Assume that  $c_{\text{FF}}$  is the parameter number of each layer in the feed-forward stage, and  $c_{\text{R}}$  is the parameter number of the recurrent stage. Therefore, the complexity order is given by  $\mathcal{O}(c_{\text{FF}}TL + c_{\text{R}}T)$ . Considering in the complexity involved in the list detection, the total complexity is about  $\mathcal{O}(c_{\text{FF}}TL + c_{\text{R}}T + c_{\text{LB}}TD_0^K|\mathcal{X}|^K)$ . It is much less than the computational complexity of LB method, which grows exponentially with the number of the available NOMA signatures.



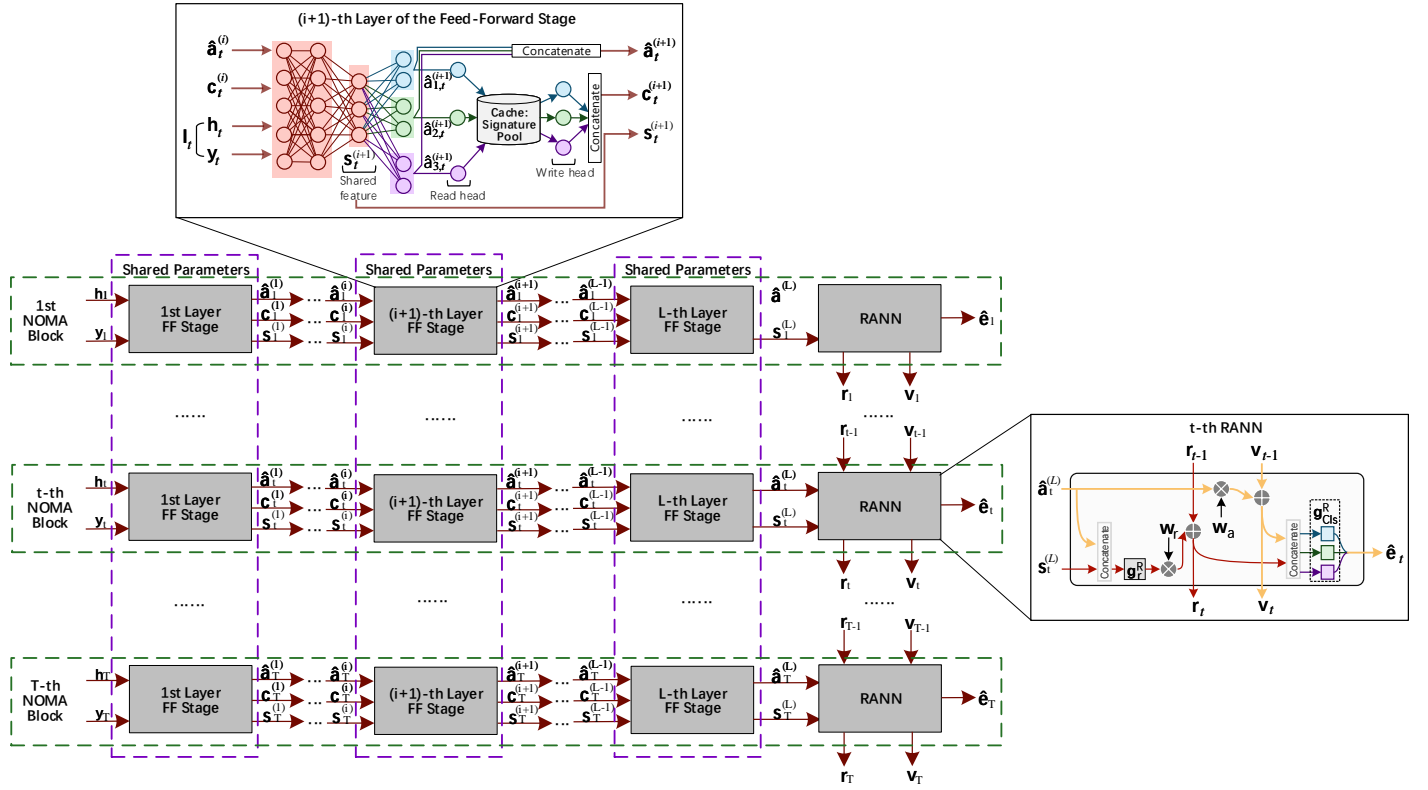


Fig. 5. The overall architecture of DeepClassifier.

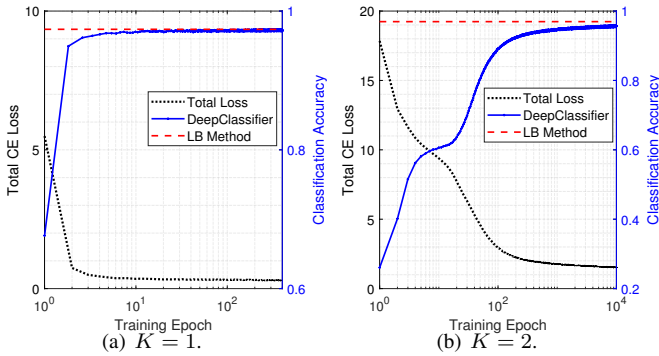


Fig. 6. Network training performance of the feed-forward stage of DeepClassifier with  $J = 6$ ,  $L = 1$ , and  $K = 1, 2$ .

## V. EXPERIMENT RESULTS

In this section, we use the Tensorflow framework to implement the proposed DeepClassifier. The depth and the width of DNN adopted in this section are empirically determined. For comparisons, the performances of other straightforward DNN methods are evaluated. Meanwhile, Monte-Carlo simulations are also conducted based on the optimal LB method.

### A. Performance of the Feed-Forward Stage

In this section, we emphasize on the training performance of the feed-forward stage of DeepClassifier. During experiments, the signature pool  $\mathcal{U}$  is assumed to be composed of the equiangular tight frames (ETF) based signatures, as one recommendation in 3GPP Study Item of NOMA [15]. Each

TABLE I  
EXPERIMENTAL SETUP OF NETWORK TRAINING.

Experimental Setup	Values or Assumptions
User Numbers	1, 2
Signature Pool	6
Training SNR	14 dB
Channel Model	Rayleigh fading channel
Dataset Size	1000000
Network layer	5
Network Width	128
Batch Size	128
Learning Rate	0.0003
Optimizer	Adam

complex spreading signature is generated by solving Grassmannian line packing problem with the parameter  $|\mathcal{U}| = J$ .

Fig. 6 presents the total loss and classification accuracy performance versus training epoch under Rayleigh fading channel with  $K = 1$  and 2. The sizes of the signature pool and the transmit blocks are set as  $J = 6$  and  $T = 1$ , respectively. Besides, Rayleigh fading channel model is considered and the training signal-to-noise ratio (SNR) is set as 14 dB. The size of the dataset  $\mathcal{D}$  is set as  $M = 10^6$ . During both training and testing stages, each complex data sample is decomposed into the real part and the imaginary part as the input to DeepClassifier. To achieve good performance in this case, we simply use the feed-forward stage of DeepClassifier with  $L = 1$ , where

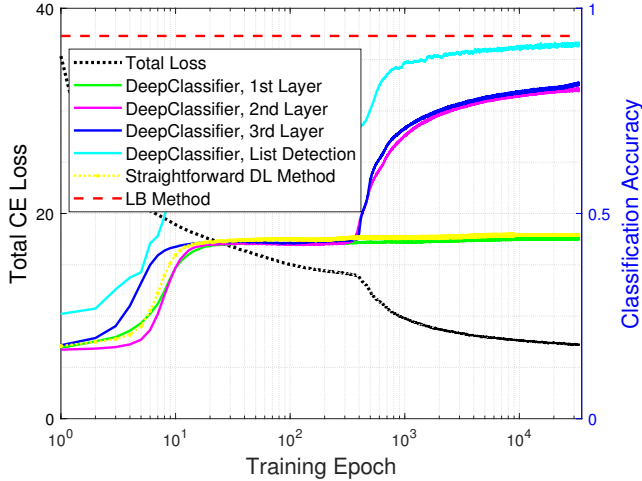


Fig. 7. Network training performance of multiple layers in the feed-forward stage of DeepClassifier with  $J = 6$ ,  $L = 3$ ,  $K = 3$ , and  $T = 1$ .

$\mathbf{g}_{\text{Pub}}^{\text{FF-(1)}}(\cdot)$  is parameterized by a 5-layer fully-connected DNN with the width of 128 neurons. Adam optimizer is used for SGD. The corresponding experimental setup is summarized in Table I. The network parameter settings are empirically determined. And we note that, the proposed DeepClassifier has good robustness and is not sensitive to the hyper-parameters. As shown in Fig. 6, the total CE loss decreases during training, and the proposed method can approach the optimal LB method after convergence with respect to classification accuracy. In Table II, we display the classification accuracy between the optimal LB-based method and DeepClassifier with various SNRs and  $J$ . These results indicate that the optimal LB method can be perfectly mimicked by DNNs, which are consistent with *Theorem 2*.

Fig. 7 considers the network training performance with  $K = 3$  overlapped IoT transmitters, and displays classification accuracy of  $\hat{\mathbf{a}}_t^{(i)}$ ,  $i = 1, \dots, L$ . Other experiment settings are the same as those in Fig. 6, except that the feed-forward stage is implemented with  $L = 3$  and weight coefficients  $w_i = 1$ ,  $i = 1, \dots, L$ . Comparing the performances achieved by the 1st layer of DeepClassifier and the optimal LB method, we see that, the optimal LB method cannot be approached with a limited network size, especially when the number of the IoT devices enlarges. This validates the discussion about *Theorem 2* that the proof of existence does not ensure algorithmic learnability, and sophisticated design on network structure is required. As shown in Fig. 7, the layers with larger index  $i$  takes the side information from previous layer and thus achieves better accuracy. In this case, simply using a straightforward DNN performs way worse than the proposed DL, which validates the effectiveness of the proposed iterative layered structure. To approach the performance limit, we perform list detection based on the estimates of DeepClassifier, i.e.,  $\hat{\mathbf{a}}_t^{(3)}$ . We set a search radius  $D_0$  such that the search space of the list detection constitutes less than 10% of the full search space  $\mathbb{J}^K$ , and then display the achieved classification accuracy. It is seen that, the proposed method nearly approaches the optimal LB method with greatly reduced computational delay.

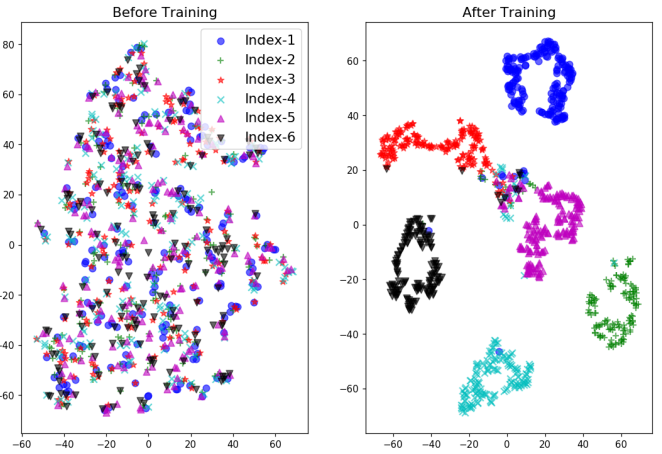


Fig. 8. Visualization of the hidden feature  $\mathbf{S}_t^{(1)}$  with t-SNE technique. Different shapes represent the indexes of different signatures in  $\mathcal{U}$ .

To validate the argument that useful features are automatically extracted during training, in Fig. 8, we visually illustrate the hidden representations  $\mathbf{s}_t^{(i)}$  before and after training with  $K = 1$  and  $J = 6$ . The visualization of  $\mathbf{s}_t^{(i)}$  is realized by t-SNE technique, which converts the high-dimensional NOMA signature features into a matrix of pairwise similarities [46]. With t-SNE technique, the complex features of different signature classes at various stages can be clearly presented. After training, we observe that the separability of the hidden features becomes more prominent. This reflects that meaningful features are automatically extracted via data-driven training.

### B. Performance of the Recurrent Stage

In Fig. 9, we numerically evaluate the advantages of the proposed recurrent structure in DeepClassifier. All the settings are the same as those in Section V-A except that  $T = 2$  and RANN is applied to aggregate the results of multiple time stamps. As illustrated in Algorithm 1, we only train the feed-forward stage for the first 2000 epochs, and then jointly train the feed-forward and the recurrent stages for the rest of the times. For comparison, we display the performance of a straightforward method, which simply accumulates the decision variables obtained by the feed-forward stage of  $T$  observations, just as (13). The performance of the optimal LB method is also provided as an reference. As observed in Fig. 9, combining the  $T$  observations can always provide better classification accuracy than directly using the results obtained by the 3rd layer of the feed-forward stage. Specifically, deploying the proposed RANN can achieve significantly better classification accuracy than the straightforward method. This indicates that the hidden features are well exploited in RANN for better classification performance. Using the list detection can further approach the performance limit.

### C. Evaluations on Classification Accuracy

In the following, we evaluate the classification accuracy performance of the proposed method with a typical NOMA system which assumes  $K = 6$ ,  $J = 6$ , and  $|\mathcal{X}| = 4$  [15].

TABLE II  
COMPARISONS OF CLASSIFICATION ERROR PROBABILITY BETWEEN LB-BASED METHOD AND DEEPCLASSIFIER.

Number of Users	Size of Signature Pool	0 dB		5 dB		10 dB		15 dB	
		LB	DL	LB	DL	LB	DL	LB	DL
K=1	J=2	0.1730	0.1735	0.0719	0.0722	0.0250	0.0252	0.0080	0.0081
	J=4	0.3108	0.3117	0.1361	0.1368	0.0499	0.0499	0.0170	0.0173
	J=6	0.3802	0.3810	0.1697	0.1699	0.0637	0.0639	0.0217	0.0218
K=2	J=2	0.2375	0.2381	0.1194	0.1201	0.0471	0.0476	0.0165	0.0168
	J=4	0.3935	0.3951	0.1996	0.2005	0.0796	0.0804	0.0279	0.0285
	J=6	0.4696	0.4717	0.2419	0.2433	0.0953	0.0961	0.0334	0.0347

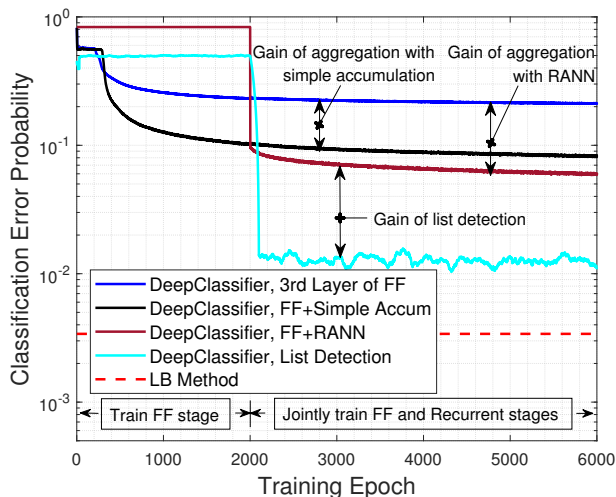


Fig. 9. Performance of DeepClassifier with  $J = 6$ ,  $K = 3$ ,  $L = 3$ , and  $T = 2$ .

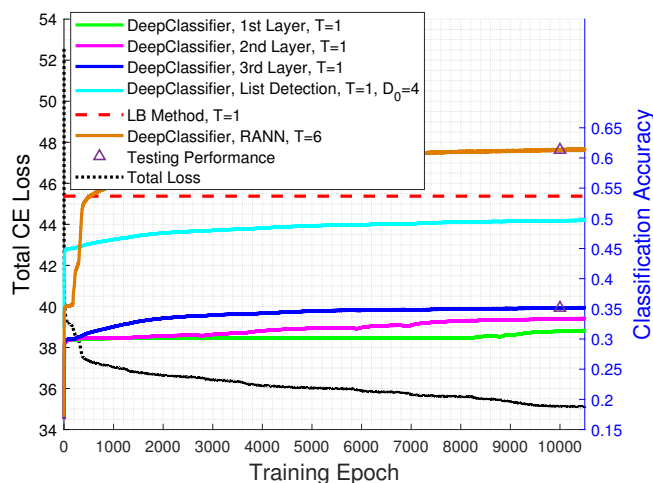


Fig. 10. Performance of DeepClassifier with  $J = 6$ ,  $K = 6$ ,  $L = 3$ , and  $T = 1, 6$ .

The network configurations and the settings of the dataset are the same as those in section V-B. Fig. 10 displays the network training performance of the proposed method with  $T = 1$  and 6. Similar to the observations in Fig. 7, for the case of  $T = 1$ , the classification accuracy is enhanced

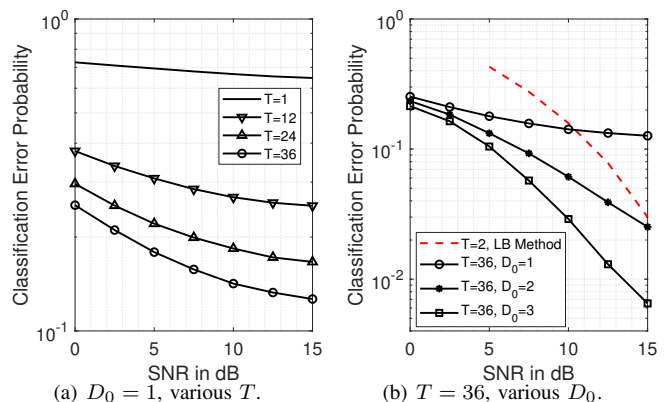


Fig. 11. Average classification error probability under various SNRs with  $K = 6$  and  $J = 6$ .

during training and deeper layer performs better than shallow layers in the feed-forward stage. Using list detection with  $D_0 = 4$  can approach the performance limit achieved by LB method with only  $\frac{4^6}{6^6} < 10\%$  computational complexity. When  $T = 6$ , owing to the aggregation of multiple observations, using RANN significantly promotes the classification accuracy and even outperforms LB method with  $T = 1$ . The testing performance, which coincides the training performance, is also illustrated in Fig. 10 to show that the trained network is efficient for deployment.

The above DeepClassifier trained with  $T = 6$  is then deployed under various SNRs and various  $T$ s, as shown in Fig. 11. As observed in Fig. 11(a), although the network is trained for  $T = 6$ , its performance promotes with the enlargement of  $T$  due to the accumulation of the information conveyed by  $T$  signal blocks. This also validates that the offline trained DeepClassifier has the flexibility and generalization ability to cope with different transmission settings during online deployment. Then we analyze the effect of the search radius  $D_0$  of the list detection. We note that, with the spreading factor of  $N = 4$ , one resource region assumed for NOMA transmission normally contains 36 NOMA signal blocks [15]. Therefore, we assume the block size  $T = 36$  and vary  $D_0$  in Fig. 11(b) to evaluate the achievable classification accuracy of the proposed method. With 10 dB SNR and  $D_0 = 2$ , the proposed scheme achieves 90% classification accuracy. With 15 dB SNR and  $D_0 = 3$ , an classification accuracy better than 99% can also be achieved. Since IoT services such

as machine-type communications normally accept the block error rate (BLER) at the the level of 10% per transmission [15], the achieved level of accuracy is acceptable for practical deployment. As a reference, we show the performance of the optimal LB method with  $T = 2$ , where 97% classification accuracy can be achieved with 15 dB. Nonetheless, even with  $T = 2$  NOMA blocks, the LB method leads to a computational delay 100 times larger than the proposed deep learning scheme with  $T = 36$  and  $D_0 = 3$  when inferencing in Python. In this case, DeepClassifier realizes the reduction of 90% deployment complexity compared with LB method. As for the training complexity, DeepClassifier uses  $731.3K$  training parameters and requires  $2.91M$  floating point operations (FLOPs), which is lower than many existing classification network models [47]–[50] as illustrated in Table III. These results reflect that the proposed DeepClassifier is an efficient method to achieve an acceptable signature classification accuracy with acceptable computational complexity and delay. Owing to the above properties, DeepClassifier can also be applied into other emerging systems with massive connectivity and high energy efficiency requirements, e.g., reconfigurable intelligent surface (RIS) [51] and holographic multiple-input-multiple-output surfaces (HMIMOS) [52].

TABLE III  
THE COMPARISONS OF NETWORK COMPLEXITY

Network Model	Reference	MFLOPs
DeepClassifier		2.91
NAS-based AMC	[47]	4.82
CRFN-CSS	[47]	20.87
MobileNetV3	[47]	44.37
AlexNet	[48]	43.20
DistAMC	[49]	4.42
VGG	[50]	8.34

## VI. CONCLUSION

This paper considered the blind signature classification problem for NOMA in future IoT networks. An optimal LB method was proposed and then proven to achieve perfect classification given enough received signal blocks. To achieve a good tradeoff between computational complexity and classification accuracy, we proposed a AI-driven deep learning method, i.e., DeepClassifier, to automatically extract efficient features for signature classification. Sophisticated network structures were designed for DeepClassifier by embedding the communication domain-expertise, and the effectiveness of the proposed designs were validated by extensive experiments. The proposed method could achieve 99% classification accuracy under a typical NOMA transmission configuration with rather low computational complexity and delay, compared with the optimal LB method. In the future, the proposed network structures can be extended to other emerging systems with non-orthogonal connectivity such as massive MIMO, RIS, and HMIMOS.

## APPENDIX A PROOF OF LEMMA 1

Given  $\sigma \rightarrow 0$ , we use mathematical induction to prove this Lemma.

(1) Consider the single-user case with  $K = 1$  transmitters. According to Assumption 2, i.e.,  $\mathbf{f}_j(x) \neq \mathbf{f}_c(s)$  if either  $j \neq c$  or  $x \neq s$ , we readily see that ambiguity cannot happen since (15) does not hold.

(2) Consider the two-user case with  $K = 2$  transmitters. Assume that there exists the ambiguity, i.e., there exist a non-identical mapping  $\pi$  and  $x_i, x_j, x_u, x_v \in \mathcal{X}$ , such that

$$h_1 \underbrace{(\mathbf{f}_{l_1}(x_i) - \mathbf{f}_{\pi(l_1)}(x_u))}_I + h_2 \underbrace{(\mathbf{f}_{l_2}(x_j) - \mathbf{f}_{\pi(l_2)}(x_v))}_II = 0. \quad (39)$$

Note that at least one term out of I and II does not equal to zero, since the opposite will makes  $\pi$  a identical mapping.

Therefore we have two situations:

a). Only one term among I and II does not equal to zero. Assume that  $I \neq 0$  and  $II = 0$ . Then (39) holds only when  $h_1 = 0$ .

b). Both terms equal to zero, i.e.,  $I = 0$  and  $II = 0$ . Then (39) holds only when  $h_1 = h_2 = 0$  or  $h_1/h_2 = -I/II$ .

Since  $P_h(h)$  is a non-discrete distribution, both cases happen with probability zero, according to measure theory.

(3) Consider the case with arbitrary number  $K$  of the transmitters. We can regard  $K - 1$  transmitters among the  $K$  transmitters as one single transmitter. Then the  $K$ -user case can be generalized into the two-user case. Therefore, we have Lemma 1.

As a straightforward extension, we consider the case where  $\sigma \neq 0$  and the probability density function of  $P_{\mathbf{y}, \mathbf{h}|\mathbf{l}^*}(\mathbf{y}_t|\mathbf{h}_t, \mathbf{l})$  follows the Gaussian mixture model, where the means of each component Gaussian model is defined by the  $\mathbf{h}_t^\top \mathcal{F}_{\mathbf{l}^*}(x_{1,t})$ . According to Lemma 1, we have the following proposition:

$$\text{Prob} [P_{\mathbf{y}, \mathbf{h}|\mathbf{l}}(\mathbf{y}_t|\mathbf{h}_t, \mathbf{l}) = P_{\mathbf{y}, \mathbf{h}|\mathbf{l}^*}(\mathbf{y}_t|\mathbf{h}_t, \mathbf{l})] = 0, \quad (40)$$

since the means of the Gaussian mixture models are unequal in probability.

## APPENDIX B PROOF OF THEOREM 1

Recall that the decision variable  $\mathcal{D}_t(\mathbf{y}_t, \mathbf{h}_t, \mathbf{l})$  defined as

$$\mathcal{D}_t(\mathbf{y}_t, \mathbf{h}_t, \mathbf{l}) = \ln [P_{\mathbf{y}, \mathbf{h}|\mathbf{l}}(\mathbf{y}_t, \mathbf{h}_t|\mathbf{l})]. \quad (41)$$

Observe that the received signal samples are i.i.d. To prove that (14) holds, it is equivalent to prove that

$$\mathbb{E}_{\mathbf{y}_t, \mathbf{h}_t} [\mathcal{D}_t(\mathbf{y}_t, \mathbf{h}_t, \mathbf{l}^*) - \mathcal{D}_t(\mathbf{y}_t, \mathbf{h}_t, \mathbf{l})] > 0. \quad (42)$$

The left-hand-side of (42) can be written as

$$\begin{aligned} & \text{LHS of (42)} \\ &= \int [\mathcal{D}_t(\mathbf{y}_t, \mathbf{h}_t, \mathbf{l}^*) - \mathcal{D}_t(\mathbf{y}_t, \mathbf{h}_t, \mathbf{l})] P_{\mathbf{y}, \mathbf{h}|\mathbf{l}}(\mathbf{y}_t, \mathbf{h}_t|\mathbf{l}^*) d\mathbf{y}_t d\mathbf{h}_t \\ &= \int \ln \frac{P_{\mathbf{y}, \mathbf{h}|\mathbf{l}}(\mathbf{y}_t, \mathbf{h}_t|\mathbf{l}^*)}{P_{\mathbf{y}, \mathbf{h}|\mathbf{l}}(\mathbf{y}_t, \mathbf{h}_t|\mathbf{l})} P_{\mathbf{y}, \mathbf{h}|\mathbf{l}}(\mathbf{y}_t, \mathbf{h}_t|\mathbf{l}^*) d\mathbf{y}_t d\mathbf{h}_t. \end{aligned} \quad (43)$$

We see that (43) follows the definition of Kullback-Leibler divergence (KLD), i.e.,

$$(43) = \text{KLD}(P_{\mathbf{y}, \mathbf{h}|\mathbf{l}}(\mathbf{y}_t, \mathbf{h}_t|\mathbf{l}^*) || P_{\mathbf{y}, \mathbf{h}|\mathbf{l}}(\mathbf{y}_t, \mathbf{h}_t|\mathbf{l})) \geq 0, \quad (44)$$

where the equal sign holds only when  $P_{\mathbf{y}, \mathbf{h}|\mathbf{l}}(\mathbf{y}_t, \mathbf{h}_t|\mathbf{l}^*) = P_{\mathbf{y}, \mathbf{h}|\mathbf{l}}(\mathbf{y}_t, \mathbf{h}_t|\mathbf{l})$ . Now we show that the equal sign does not hold. According to Lemma 1,

$$\text{Prob}[P_{\mathbf{y}, \mathbf{h}|\mathbf{l}}(\mathbf{y}_t|\mathbf{h}_t, \mathbf{l}) = P_{\mathbf{y}, \mathbf{h}|\mathbf{l}^*}(\mathbf{y}_t|\mathbf{h}_t, \mathbf{l})] = 0. \quad (45)$$

Therefore, (42) holds.

## REFERENCES

- [1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb. 2014.
- [2] T. Lv, Y. Ma, J. Zeng, and P. T. Mathiopoulos, "Millimeter-wave NOMA transmission in cellular M2M communications for Internet of Things," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1989–2000, June 2018.
- [3] S. Liao, J. Wu, S. Mumtaz, J. Li, R. Morello, and M. Guizani, "Cognitive balance for fog computing resource in Internet of Things: An edge learning approach," *IEEE Transactions on Mobile Computing*. Early Access.
- [4] Z. Zhou *et al.*, "Learning-based URLLC-aware task offloading for Internet of Health Things," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 396–410, Feb. 2020.
- [5] K. Zrar Ghafoor *et al.*, "Millimeter-wave communication for Internet of Vehicles: Status, challenges, and perspectives," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8525–8546, Sept. 2020.
- [6] S. Sun, M. Kadoch, L. Gong, and B. Rong, "Integrating network function virtualization with SDR and SDN for 4G/5G networks," *IEEE Network*, vol. 29, no. 3, pp. 54–59, May-June 2015.
- [7] M. Liu, J. Yang, and G. Gui, "DSF-NOMA: UAV-assisted emergency communication technology in a heterogeneous Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5508–5519, June 2019.
- [8] Z. Shi *et al.*, "Zero-forcing based downlink virtual MIMO-NOMA communications in IoT Networks," *IEEE Internet of Things Journal*. Early Access.
- [9] Y. Wu, B. Rong, K. Salehian, and G. Gagnon, "Cloud transmission: A new spectrum-reuse friendly digital terrestrial broadcasting transmission system," *IEEE Transactions on Broadcasting*, vol. 58, no. 3, pp. 329–337, Sept. 2012.
- [10] G. Liu, Z. Wang, J. Hu, Z. Ding, and P. Fan, "Cooperative NOMA broadcasting/multicasting for low-latency and high-reliability 5G cellular V2X communications," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7828–7838, Oct. 2019.
- [11] M. Mohammadkarimi, M. A. Raza, and O. A. Dobre, "Signature-based nonorthogonal massive multiple access for future wireless networks: Uplink massive connectivity for machine-type communications," *IEEE Vehicular Technology Magazine*, vol. 13, no. 4, pp. 40–50, Dec. 2018.
- [12] N. Ye, H. Han, L. Zhao, and A. Wang, "Uplink nonorthogonal multiple access technologies toward 5G: A survey," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 6187580, 26 pages, 2018.
- [13] S. Moon, H. Lee, and J. Lee, "SARA: Sparse code multiple access-applied random access for IoT devices," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3160–3174, Aug. 2018.
- [14] M. Jia, L. Wang, Q. Guo, X. Gu, and W. Xiang, "A low complexity detection algorithm for fixed up-link SCMA system in mission critical scenario," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3289–3297, Oct. 2018.
- [15] 3GPP, TR 38.812, Study on Non-Orthogonal Multiple Access (NOMA) for NR (Release 16).
- [16] K. Yang, N. Yang, N. Ye, M. Jia, Z. Gao, and R. Fan, *et al.*, "Non-orthogonal multiple access: Achieving sustainable future radio access," *IEEE Communication Magazine*, vol. 57, no. 2, pp. 116–121, Feb. 2019.
- [17] N. Ye, X. Li, H. Yu, A. Wang, W. Liu, and X. Hou, "On constellation rotation of NOMA with SIC receiver," *IEEE Communications Letters*, vol. 22, no. 3, pp. 514–517, Mar. 2018.
- [18] Z. Ding *et al.*, "Application of non-orthogonal multiple access in LTE and 5G Networks," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 185–191, Feb. 2017.
- [19] B. Rong, Y. Qian, K. Lu, H. Chen, and M. Guizani, "Call admission control optimization in WiMAX networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2509–2522, July 2008.
- [20] N. Zhang, K. Cheng, and G. Kang, "A machine-learning-based blind detection on interference modulation order in NOMA systems," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2463–2466, Dec. 2018.
- [21] M. Choi, D. Yoon, and J. Kim, "Blind signal classification for non-orthogonal multiple access in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9722–9734, Oct. 2019.
- [22] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Communications*, vol. 1, pp. 137–156, Apr. 2007.
- [23] Wen Wei and J. M. Mendel, "Maximum-likelihood classification for digital amplitude-phase modulations," *IEEE Transactions on Communications*, vol. 48, no. 2, pp. 189–193, Feb. 2000.
- [24] A. Ramezani-Kebrya, I. Kim, D. I. Kim, F. Chan, and R. Inkol, "Likelihood-based modulation classification for multiple-antenna receiver," *IEEE Transactions on Communications*, vol. 61, no. 9, pp. 3816–3829, Sept. 2013.
- [25] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Transactions on Communications*, vol. 48, pp. 416–429, Mar. 2000.
- [26] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, Sept. 2018.
- [27] T. J. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Dec. 2017.
- [28] A. Ali and F. Yangyu, "Automatic modulation classification using deep learning based on sparse autoencoders with nonnegativity constraints," *IEEE Signal Processing Letters*, vol. 24, no. 11, pp. 1626–1630, Nov. 2017.
- [29] F. Meng, P. Chen, L. Wu, and X. Wang, "Automatic modulation classification: A deep learning enabled approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10760–10772, Nov. 2018.
- [30] S. Peng *et al.*, "Modulation classification based on signal constellation diagrams and deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 718–727, Mar. 2019.
- [31] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, Feb. 2018.
- [32] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 4074–4077, Apr. 2019.
- [33] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [34] S. Zhou, Z. Wu, Z. Yin, R. Zhang, and Z. Yang, "Blind modulation classification for overlapped co-channel signals using Capsule networks," *IEEE Communications Letters*, vol. 23, no. 10, pp. 1849–1852, Oct. 2019.
- [35] H. He, S. Jin, C. Wen, F. Gao, G. Ye Li, and Z. Xu, "Model-driven deep learning for physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 77–83, Oct. 2019.
- [36] N. Ye, X. Li, H. Yu, A. Wang, W. Liu, and X. Hou, "Deep learning aided grant-free NOMA towards reliable low-latency access in tactile Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2995–3005, May 2019.
- [37] R. Abbas, M. Shirvanimoghaddam, Y. Li, and B. Vucetic, "A novel analytical framework for massive grant-free NOMA," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2436–2449, Mar. 2019.
- [38] M. Borgerding, P. Schniter, and S. Rangan, "AMP-inspired deep networks for sparse linear inverse problems," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4293–4308, Aug. 2017.
- [39] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Network*, vol. 2, no. 5, pp. 359–366, 1989.
- [40] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *Proc. 2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Sapporo, 2017, pp. 1–5.
- [41] A. Graves, G. Wayne, and I. Danihelka, "Neural Turing machines," *CoRR*, vol. abs/1410.5401, 2014.
- [42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 770–778.
- [43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: <http://www.deeplearningbook.org>
- [44] Z. Yuan *et al.*, "Multi-user shared access for Internet of Things," in *Proc. IEEE 83rd VTC Spring*, Nanjing, 2016, pp. 1–5.



- [45] W. Liu, X. Hou, and L. Chen, "Enhanced uplink non-orthogonal multiple access for 5G and beyond systems", *Front. Inf. Technol. & Elec. Eng.*, vol. 19, no. 3, pp. 340–356, Mar. 2018.
- [46] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
- [47] X. Wei, W. Luo, X. Zhang, J. Yang, G. Gui and T. Ohtsuki, "Differentiable Architecture Search-Based Automatic Modulation Classification," in *Proc. 2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1-6.
- [48] Y. Tu and Y. Lin, "Deep Neural Network Compression Technique Towards Efficient Digital Signal Modulation Recognition in Edge Device," *IEEE Access*, vol. 7, pp. 58113-58119, Apr. 2019.
- [49] Y. Wang et al., "Distributed Learning for Automatic Modulation Classification in Edge Devices," *IEEE Wireless Communications Letters*, vol. 9, no. 12, pp. 2177-2181, Dec. 2020.
- [50] Z. Ke and H. Vikalo, "Real-Time Radio Modulation Classification With An LSTM Auto-Encoder," in *Proc. 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 4935-4939.
- [51] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah and C. Yuen, "Reconfigurable Intelligent Surfaces for Energy Efficiency in Wireless Communication," *IEEE Transactions on Wireless Communications*, vol. 18, no. 8, pp. 4157-4170, Aug. 2019.
- [52] C. Huang et al., "Holographic MIMO Surfaces for 6G Wireless Networks: Opportunities, Challenges, and Trends," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 118-125, Oct. 2020.

2022-01-31

# AI-driven blind signature classification for IoT connectivity: a deep learning approach

Pan, Jianxiong

IEEE

---

Pan J, Ye N, Yu H, et al., (2022) AI-driven blind signature classification for IoT connectivity: a deep learning approach, IEEE Transactions on Wireless Communications, Volume 21, Number 8, August 2022, pp. 6033-6047

<https://doi.org/10.1109/TWC.2022.3145399>

*Downloaded from Cranfield Library Services E-Repository*