IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 21, 2020, accepted November 13, 2020, date of publication November 20, 2020,
date of current version December 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3039543

Automatic Modulation Classification Based on Hierarchical Recurrent Neural Networks With Grouped Auxiliary Memory

KE ZANG AND ZHENGUO MA
College of Biomedical Engineering and Instrument Science, Yuquan Campus, Zhejiang University, Hangzhou 310027, China
Corresponding author: Zhenguo Ma (850501@zju.edu.cn)

ABSTRACT As a valuable topic in wireless communication systems, automatic modulation classification has been studied for many years. In recent years, recurrent neural networks (RNNs), such as long short-term memory (LSTM), have been used in this area and have achieved good results. However, these models often suffer from the vanishing gradient problem when the temporal depth and spatial depth increases, which diminishes the ability to latch long-term memories. In this paper, we propose a new hierarchical RNN architecture with grouped auxiliary memory to better capture long-term dependencies. The proposed model is compared with LSTM and gated recurrent unit (GRU) on the RadioML 2016.10a dataset, which is widely used as a benchmark in modulation classification. The results show that the proposed network yields a higher average classification accuracy under varying signal-to-noise ratio (SNR) conditions ranging from 0 dB to 20 dB, even with much fewer parameters. The performance superiority is also confirmed using a dataset with variable lengths of signals.

INDEX TERMS Automatic modulation classification (AMC), recurrent neural networks (RNNs), hierarchical recurrent structure, long-term memory.

I. INTRODUCTION

Automatic modulation classification (AMC) is the process of deciding the modulation to be used by transmitter, based on observations of the received signal. It is becoming increasingly important in cooperative communications, especially since the advent of the software-defined autonomous radio [1]. Furthermore, AMC plays an crucial role in many applications, such as the identification of interference signals and jammers, tracking the activities of specific users [2]–[5].

An automatic modulation classifier can be defined as a system that automatically identifies the modulation type of the received signal, given that the signal exists and that its parameters are distributed in a known range. This needs a universal modulation recognizer capable of classifying a comprehensive list of modulation schemes. Quite a few scholars have presented a variety of excellent approaches, which can be roughly divided into two categories: likelihood-based (LB) approaches and feature-based (FB) approaches [6]. LB methods are based on the likelihood function of the received signal, wherein the decision is made by comparing the likelihood ratio against a threshold [2]. Even though LB methods usually exhibit high accuracy and minimize the probability of mismatches, such methods suffer from high latency or require complete priori knowledge like the clock frequency offset. LB approaches are also prone to mismatching when applying the theoretical system model to the actual scene. FB approaches usually extract certain features from the received signals; then, selected classifiers are used to classify different modulation signals [6]. Traditional feature methods mainly include instantaneous time features, statistical features (moments, cumulants, and cyclostationarity), and transform features. These features, with an efficient classifier for AMC, have achieved satisfactory performances.

In recent years, deep neural networks have received much attention in many application domains, such as computer vision [7], natural language processing [8] in which recurrent neural networks (RNNs) occupy an extremely important position in sequence processing and classification because they are efficient feature extractor and classifier. In most cases, the features extracted by the neural network are more effective than those extracted manually. Moreover, manually

The associate editor coordinating the review of this manuscript and approving it for publication was Matti Hämäläinen.

extracting features from data may cause the loss of information that is necessary for the classifiers [9]. Therefore, many researchers try to use deep neural networks such as convolutional neural networks (CNNs) and long short-term memory (LSTM) [10] for AMC problems.

CNNs exploit spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. CNNs share weights among all neurons in a particular feature map and each neuron is connected only to a subset of the input. This helps to reduce the number of parameters in the whole system and makes the computation more efficient. Despite its great achievements in spatial feature extraction, CNNs do not perform well in modeling time-series changes. To learn the changes in time series, error signals must travel for a long temporal distance when backpropagating through time. The difficulties arising from the large temporal lengths of RNNs are significantly alleviated by LSTMs [11]. However, training LSTMs to work deep in both time and space still poses a challenge, and there are shortcomings, which hinder the learning of long-term temporal changes, such as the vanishing gradient problem.

In this paper, we propose a modified hierarchical recurrent neural networks with an grouped auxiliary memory (GAM-HRNN). We use a hierarchical structure by stacking the layers with shortcut connection from grouped auxiliary memory(GAM [11]) to each layer. Recurrent structures such as LSTM, gated recurrent unit (GRU) [12], and update gate recurrent neural network (UGRNN) [13] can be used in HRNN. The contributions of this paper are as follows: firstly, we develop a new recurrent-model-based deep learning solution, wherein the experimental results show high accuracy on a standard dataset compared with LSTM, GRU, Recurrent highway networks (RHNs) [14] and recurrent highway networks with grouped auxiliary memory (GAM-RHNs) [11]. Secondly, even with fewer parameters, the model achieves comparable and decent results. Thirdly, we explore the norm of gradients of the mentioned models during the training process. Finally, we find that the proposed model also provides an advantage when dealing with signals of unfixed lengths. The rest of the paper is organized as follows. Section II introduces related studies. Section III formulates the problem and introduces the standard and modified datasets. We introduce the LSTM model, GRU model and RHNs in Section IV and explains our GAM-HRNN in Section V. Section VI presents the experiments and analysis, and Section VII concludes the paper.

II. RELATED WORKS

In this section, a brief introduction of the works using traditional methods is provided first. Then, we review the work that relates to our method in detail.

A. TRADITIONAL METHODS

Traditional methods can be primarily divided into two categories: LB and FB approaches. Chavali *et al.*used LB algorithms for modulation classification in fading non-Gaussian channels [15]. Then, they modeled the additive Gaussian noise with a Gaussian mixture distribution model. FB methods use instantaneous time features, statistical features, transform features, and other features, including constellation shape. Yuan *et al.* [16] developed an algorithm using wavelet transform and pattern recognition for analog and digital modulation classification. Wavelet transform was used to estimate the symbol rate of the received signals to separate analog signals from digital signals. Ananthram *et al.* proposed a method based on elementary fourth-order cumulants [17]. The cumulant-based classification is particularly effective when used in a hierarchical scheme. In [18], the authors used various statistical moments of the signal amplitude, phase, and frequency with a fuzzy classifier; their technique performed well at low signal-to-noise ratios (SNRs).

Researchers also used artificial neural networks (ANNs) as a classifier. In [19], a common ANN achieved good performance dealing with analog and digital modulation-type classification. Although ANN had achieved success in modulation recognition, its overdependence on sample training data and easily settling into a local optimum solution restricted its performance and application. In [20], the authors proposed a hierarchical support vector machine(SVM)-based structure and used higher order moments and cumulants for AMC. It improved the performance of the recognizer efficiently. Aslam *et al.* [21] explored the use of genetic programming (GP) in combination with K-nearest neighbor (KNN) for AMC. KNN was used to evaluate fitness of GP individuals during the training phase. As we can see, each FB method had its own advantages, and all traditional machine learning methods, such as KNN, SVM, and ANN, had been used as a classifier. As mentioned previously, in many domains, the features learned automatically were more effective than those extracted manually. The separation of the feature extraction and classifier always led to information loss. This led researchers to consider deep neural network methods.

B. DEEP NEURAL NETWORK METHODS

With the experimental conditions generally standardized, many researchers used deep neural network methods, which mainly included CNNs and LSTMs for the AMC problem and achieved excellent performance. Deep neural network methods depend on sample training data. O'Shea *et al.* built the RadioML 2016.10a dataset in 2016 [22], and achieved a good performance on it with a simple CNN model. Peng *et al.* indicated that their proposed CNN-based model achieved good accuracy without the necessity of manual feature selection compared with SVM for AMC [23]. They also used two famous CNN-based models, AlexNet and GoogLeNet for AMC and converted complex signals into data formats in a grid-like topology, e.g., images that facilitated the use of prevalent deep neural network models and frameworks for classification [24]. The experiments indicated that CNN models show a significant performance advantage and application feasibility. Teacher and student networks were used to shrink the size of the model, with a slight accuracy decrease in [25].

Huang *et al.* used compressive CNN for modulation classification [5]. Furthermore, CNN models were also used to classify the modulation types in an orthogonal frequency-division multiplexing system, wherein the modulation classification accuracy was limited [26].

Temporal dependencies are important in AMC problems, and an LSTM can learn those features effectively. Rajendran *et al.* proposed an LSTM for AMC on the standard RadioML 2016.10a dataset, without requiring expert features like higher order cyclic moments [27]. The simple LSTM model yielded an average classification accuracy of approximately 90%, under varying SNR conditions, ranging from 0 dB to 20 dB, which will be compared herein. They showed that an LSTM-based model can learn good representations of the time-domain sequences for the AMC problem. However, the forget bias of the model should be set to 1.0 manually, and to achieve high accuracy, many parameters were necessary. Pascanu *et al.* presented the methods for constructing deep RNNs [28]. Simply increasing the recurrence depth yielded RHNs [14], which settled the vanishing gradient problem in spatial depth to some extent, but the representation capability was low. For hierarchical structure, each layer had its own states, which promoted the variety and improved the representation capability of the model. However, LSTM also faced problems that degraded its performance, for example, the vanishing gradient problem as the model grew deeper in both time and space. In [29], neurons in the same layer are independent of each other and they connected across layers. Hu and Zheng tried to modify the memory update method of LSTM network and achieved good results on prediction tasks. [30] Tensorized LSTM are employed to model the temporal patterns and an adaptive shared memory was used to help the networks learn the relatedness among tasks in [31].

Combining the CNN with LSTM in parallel mode and serial mode achieved better performance compared with independent networks [9]. Huang *et al.* applied a few data augmentation methods, such as rotation, flip, and Gaussian noise, on the RadioML Dataset. Their experiments showed that the rotation method yielded the best accuracy [32], and needed fewer data to achieve relatively good results. For the convenience of application, exploring lightweight networks is also very important for AMC problems. Wang *et al.* proposed a pruning method for networks and obtained a model with fewer parameters and comparable accuracy [33].

III. PROBLEM FORMULATION AND DATASET INTRODUCTION

A. COMMUNICATION SIGNAL FORMULATION

Consider digital modulation as an example. At the receiver side, the received signal $y(t)$ can be formulated as follows:

$$y(t) = x(t) * c(t) + n(t). \tag{1}$$

Here, $x(t)$ is the modulated signal from the transmitter, $c(t)$ represents the time-varying impulse response of the transmitted wireless channel. $n(t)$ denote the additive white Gaussian noise. The generation of $x(t)$ is shown in Fig. 1,

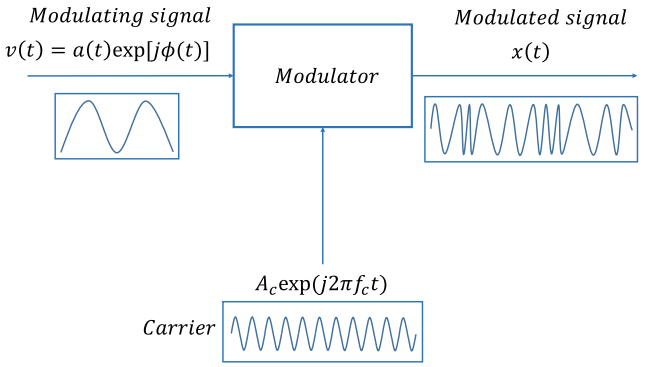


FIGURE 1. Generalized modulation system.

TABLE 1. Modulation parameters.

Modulation types	BPSK,QPSK,8PSK,16QAM, 64QAM,BFSK,CPFSK,PAM4, WB-FM,AM-SSB,AM-DSB.
Sample length	128
Samples per symbol	8
SNR range	[-20, 18] step 2dB.
Training sequences number	110000.
Testing sequences number	110000.

The modulating signal and carrier signal are mixed to obtain the modulated signal $x(t)$, which is formulated as follows:

$$\begin{aligned} x(t) &= \Re(v(t)A_c exp(j2\pi f_c t)) \\ &= A_c a(t) cos(2\pi f_c t + \phi(t)) \end{aligned} \tag{2}$$

Here, $v(t)$ denotes the modulating signal, A_c represents the amplitude of the carrier, and f_c denotes the carrier frequency. The aim of AMC is to use the received signal $y(t)$ to maximize the value of $P(x(t) \in N_i|y(t))$, where N_i is the i_{th} category of all the modulation types.

B. DATASET INTRODUCTION

1) STANDARD DATASET

The standard RadioML 2016.10a dataset [22] consists of 11 modulations: 8 digital and 3 analog modulations, with 4 samples/symbol and a sample length of 128 samples. All are widely used in wireless communications systems globally. Radio channel effects are relatively well-characterized. Realistic non-ideal effects, such as thermal noise, oscillator drift, symbol timing offset, sample rate offset, carrier frequency offset, and phase difference, are reflected in the data. These parameters are shown in Table 1.

2) VARIABLE LENGTH SEQUENCES

To explore the model's ability to handle the modulation of different parameters with variable lengths, we used the RML2018.01a, which was first used in [34] to obtain variable length signals. In this dataset, each sequence had 1024 samples, with SNR ranging from −20 dB to 30 dB. We took the first 11 modulation types in the dataset. Then, we randomly selected the sequences to split them into lengths of 128, 256,

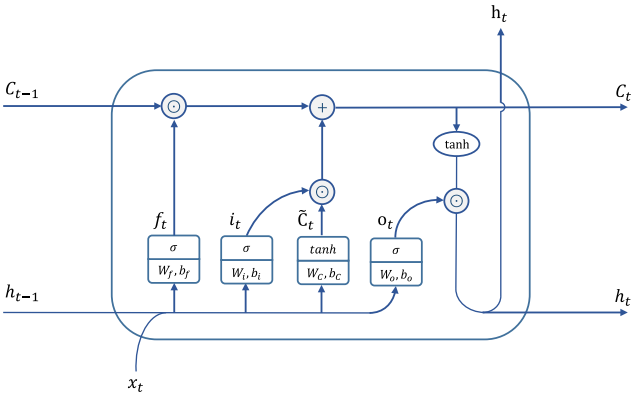


FIGURE 2. Long short-term memory.

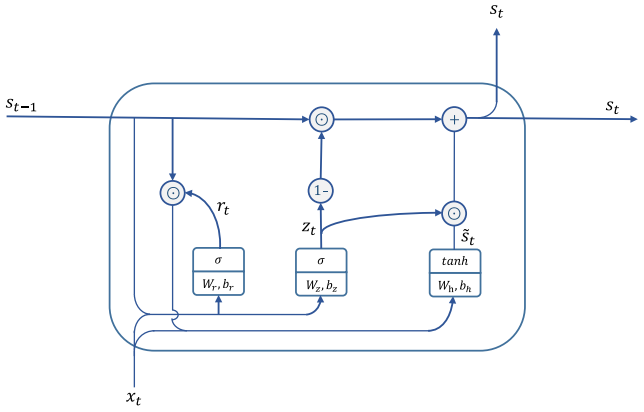


FIGURE 3. Gated recurrent unit.

and 512 with SNR ranging from −20 dB to 20 dB. The number of samples was 99,000 in both the training and testing sequences.

IV. RELATED MODELS

NOTATION

Boldface letter are used for vectors and matrices;**1** denotes vectors of ones. σ denotes the sigmoid activation function, tanh denotes the hyperbolic function, and $\odot$ represents element-wise multiplication.

A. LONG SHORT-TERM MEMORY

LSTM is one of the most popular models of RNNs. It was first proposed by Hocreiter and Schmidhuber in 1997 [10]. The case idea of LSTM is to protect the integrity of messages with the control of writing memories. LSTM uses three mechanisms to achieve this: write control, read control, and forget control. Write control uses some units to cancel out some useless information, read control cancels out the irrelevant information, and forget control selectively forgets the least relevant old information. The mechanisms are realized by three gates shown in Fig. 2, and formulated as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{h}_{t-1} + \mathbf{U}_i\mathbf{x}_t + \mathbf{b}_i). \tag{3a}$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_\mathbf{o}\mathbf{h}_{t-1} + \mathbf{U}_o\mathbf{x}_t + \mathbf{b}_o). \tag{3b}$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{h}_{t-1} + \mathbf{U}_f\mathbf{x}_t + \mathbf{b}_f). \tag{3c}$$

Candidate values:

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C\mathbf{h}_{t-1} + \mathbf{U}_C\mathbf{x}_t + \mathbf{b}_C). \tag{4}$$

Drop the useless information and add new information:

$$\mathbf{C}_t = \mathbf{C}_{t-1} \odot \mathbf{f}_t + \tilde{\mathbf{C}}_t \odot \mathbf{i}_t. \tag{5}$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t). \tag{6}$$

LSTM first gives the candidate write $\tilde{C}_t$, then uses the forget gate and input gate to update the state. Finally, it uses the output gate to provide the output of the model. However, LSTM still has problems, such as write conflicts and read conflicts [10]. This hinders the model from keeping the memory for long time steps.

B. GATED RECURRENT UNIT

GRU, which was first introduced by Chung *et al.* in 2014, is another popular model of RNNs [12]. GRU explicitly links the state, coordinates the writes, and forgets, as presented in Fig. 3. The formulation is as follows:

$$\mathbf{r}_t = \sigma(\mathbf{W}_r\mathbf{s}_{t-1} + \mathbf{U}_r\mathbf{x}_t + \mathbf{b}_r). \tag{7a}$$
$$\mathbf{z}_t = \sigma(\mathbf{W}_z\mathbf{s}_{t-1} + \mathbf{U}_z\mathbf{x}_t + \mathbf{b}_z). \tag{7b}$$
$$\tilde{\mathbf{s}}_t = \tanh(\mathbf{W}_s(\mathbf{r}_t \odot \mathbf{s}_{t-1} + \mathbf{U}_s\mathbf{x}_t + \mathbf{b}_s)). \tag{8}$$
$$\mathbf{s}_t = \mathbf{z}_t \odot \mathbf{s}_{t-1} + (\mathbf{1} - \mathbf{z}_t) \odot \tilde{\mathbf{s}}_t. \tag{9}$$

Instead of performing selective writes and selective forgets, GRU foregoes some expressiveness and selectively overwrites by setting the forget gate equal to 1 minus the write gate. The update gate z_t is the same as the forget gate from the prototype LSTM, f_t and the input gate is calculated by $1 - z_t$. This works because it turns s_t into an element-wise weighted average of s_{t-1} and $\tilde{s}_t$, which is bounded if both s_{t-1} and $\tilde{s}_t$ are bounded. GRU is an alternative to the LSTM, but GRU outperformed LSTM on nearly all tasks except language modeling with the naive initialization [35]

C. RECURRENT HIGHWAY NETWORK

Recurrent highway network (RHN) was first proposed in 2017 by Zilly et al [14]. Many Sequential processing tasks require complex nonlinear transitions from one step to the next, and recurrent neural networks with deep transition functions remain difficult to train, even when using LSTM networks. RHN extends the LSTM architecture to allow step-to-step transition depths larger than one. It can use generic RNN structures such as UGRNN, LSTM, and GRU for the networks.

For example, as mentioned above, LSTM receives c_{t-1} and h_{t-1} from its last time step to compute the current c_{t-1} and h_{t-1} and hands over them to the next time. However, as shown in Fig. 11, when we use a LSTM kernel for an L-layer RHN model, c_t^1 and h_t^1 are initialed to zero for the first layer and x_t is the input of the first layer. For the second layer, the c_{t-1}^2 and h_{t-1}^2 are taken from the last layer (c_t^1 and h_t^1) at the same current t. Then, c_t^2 and h_t^2 are calculated and delivered to the

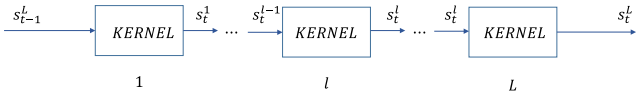


FIGURE 4. Recurrent Highway Neural Networks.

next layer. At the last layer, c_t^L and h_t^L are passed to the next time. The following is the detailed formulation (LSTM used as an example):

$$\mathbf{c}_t^\ell = \mathbf{h}_{t-1}^{\ell-1} \odot \mathbf{f}_t^\ell + \tilde{\mathbf{C}}_t^\ell \odot \mathbf{i}_t^\ell. \tag{10a}$$
$$\mathbf{h}_t^\ell = \mathbf{o}_t^\ell \odot tanh(\mathbf{C}_t^\ell) \tag{10b}$$

where ℓ denotes the current ℓ layer. In a general way,

$$\mathbf{s}_t^L = H_{KERNEL\times L}(\mathbf{s}_{t-1}, \{\tilde{\mathbf{x}}_{t\,\ell=1}^{\ell\,L}) \tag{11}$$

V. HIERARCHICAL RECURRENT NEURAL NETWORKS WITH GROUPED AUXILIARY MEMORY

In this section, we propose an HRNN with grouped auxiliary memory named GAM-HRNN. The main body of the model was built with a hierarchical structure using the generic kernel mentioned above. Shortcut reading blocks used the output of the last layer as the key to read from the auxiliary memory to obtain information as the input to the next layer. The details are presented in Fig. 5. Group auxiliary memory (GAM) denotes the auxiliary module. K represents generic RNN structures, such as LSTM, GRU, and UGRNN. Input x, previous states, and auxiliary memory were first written into GAM. The memory module m_t was partitioned equally into N groups, which is favorable for dealing with long short-term information, and each of these groups is a length vector S (The structure of GAM is shown in Fig. 6.): $m_t = (m_t^1, \cdots, m_t^N), m_t^i \in R^S (1 \le i \le N), N_m = S \times N$. We denote the softmax over groups activation as $\zeta_{S\times N} : R_{N_m} \mapsto R_{N_m}$.

Update m_t:

$$\mathbf{m}_t = (\mathbf{1} - \mathbf{a}_t^w) \odot \mathbf{m}_{t-1} + \mathbf{a}_t^w \odot (\mathbf{U}_{S\times N}\overline{\mathbf{m}}_t). \tag{12a}$$
$$\mathbf{a}_t^w = G_w(\mathbf{h}_t^w; \zeta_{S\times N}) \in R^{N_m}. \tag{12b}$$
$$\mathbf{h}_t^w = A(\mathbf{W}_h \mathbf{s}_{t-1}^1 + \mathbf{U}_h \mathbf{x}_t) + \mathbf{b}_h \tag{12c}$$
$$\mathbf{r}_t = \sigma(\mathbf{W}_r \mathbf{s}_{t-1}^1 + \mathbf{U}_r \mathbf{x}_t + \mathbf{b}_r) \tag{12d}$$
$$\overline{\mathbf{m}}_t = \tanh(\mathbf{W}_m[\mathbf{r}_t \odot \mathbf{s}_{t-1}^1, \mathbf{x}_t] + \mathbf{b}_m) \in R_N \tag{12e}$$

Here, $\mathbf{s}_{t-1}^1$ denotes the state of the first layer, the $\overline{m}_t$ serves as a 'candidate' state for the calculation, $G_w(\cdot)$ is an attentional mechanism implemented using the softmax over groups activation ζ, which can be defined as follows:

$$\mathbf{a}_t = \begin{bmatrix} \mathbf{a}_t^1 \\ \vdots \\ \mathbf{a}_t^N \end{bmatrix} = \zeta_{S\times N}(\mathbf{q}_t) = \zeta_{S\times N}(\begin{bmatrix} \mathbf{q}_t^1 \\ \vdots \\ \mathbf{q}_t^N \end{bmatrix}), \tag{13a}$$
$$\mathbf{a}_t^i = softmax(\mathbf{q}_t^i) \in R^S, \quad i = 1, 2, \cdots, N. \tag{13b}$$

Note that $N_m = S \times N$, each $a_t^{i,j} (1 \le i \le N, 1 \le j \le S)$ in a_t is between 0 and 1, and

$$\frac{1}{S \cdot N} \sum_{i=1}^{N} \sum_{j=1}^{S} a_t^{i,j} = \frac{1}{S}. \tag{14}$$

A is an affine transformation. h_t^w is the vector for writing, and will be reused by R^1 through the shortcut connectivity. The duplicating matrix $U_{S\times N} \in R^{N_m \times N}$ in (12a) is given by

$$\mathbf{U}_{S\times N} = \begin{bmatrix} \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_N \end{bmatrix} \in R^{N_m \times N} \tag{15a}$$
$$\mathbf{U}_i = \begin{bmatrix} \mathbf{u}_{i,1} & \cdots & \mathbf{u}_{i,N} \end{bmatrix} \in R^{S\times N} \tag{15b}$$
$$\mathbf{u}_{i,j} = \begin{cases} \mathbf{1}, & i = j \\ \mathbf{0}, & i \ne j \end{cases} \in R^S \tag{15c}$$

in which $1 \le i \le N, 1 \le j \le S$

Then, we updated s_t. This can comprise generic layers of RNN structures, such as LSTM, GRU, and UGRNN.

$$\mathbf{s}_t^\ell = K^\ell(\mathbf{s}_{t-1}^\ell, \tilde{\mathbf{x}}_t^\ell). \tag{16a}$$
$$\begin{aligned} \tilde{\mathbf{x}}_t^\ell &= [R^\ell(\mathbf{m}_t, \tilde{\mathbf{h}_t^\ell}), s_t^{\ell-1}]. \\ &= [V_{S\times N}(\mathbf{a}_t^{r\ell} \odot \mathbf{m}_t), s_t^{\ell-1}] \end{aligned} \tag{16b}$$
$$\tilde{\mathbf{h}_t^\ell} = \begin{cases} \mathbf{h}_t^w, & \ell = 1 \\ \mathbf{s}_t^{\ell-1}, & \ell > 1 \end{cases} \tag{16c}$$
$$\mathbf{a}_t^{r\ell} = \begin{cases} G_r^1(\mathbf{h}_t^w; \zeta_{S\times N}), & \ell = 1 \\ G_r^\ell(\mathbf{s}_t^{\ell-1}; \zeta_{S\times N}), & \ell > 1 \end{cases} \tag{16d}$$

Here, $[\cdot]$ denotes the concatenation of the elements in it, the $V_{S\times N}$ is defined as the transpose of $U_{S\times N}$ given in (15a), K denotes generic RNN kernels. The reading blocks uses $\tilde{\mathbf{h}_t^\ell}$ as the key to get useful information from GAM. In (16c), when $\ell > 1$, previous state s_{t-1}^{l-1} can be an alternative key for the reading blocks.

In the networks, information is sparsely written in the GAM based on the group mechanism and delivered to the next time step directly. This means that only a very limited portion of states in the GAM is overwritten in each time step. Thus, the information can be efficiently maintained. Subsequently, for each layer, the required information is also read from GAM sparsely using the new state of the current layer as the key. Finally, the required information is sent to the next layer, along with the output of the previous layer. In the backpropagation view, the networks offer a shortcut for the error signals that back-propagate in both temporal and spatial dimensions.

VI. EXPERIMENTS AND DISCUSSION

In this section, we evaluate the performance of the proposed model on both RadioML2016.10a dataset and the variable length dataset. As shown in Table 1, we equally divided the standard data set into training and testing sets with the data in IQ format. Using the IQ data, we derived the amplitude sequence and phase vector. The amplitude vectors were

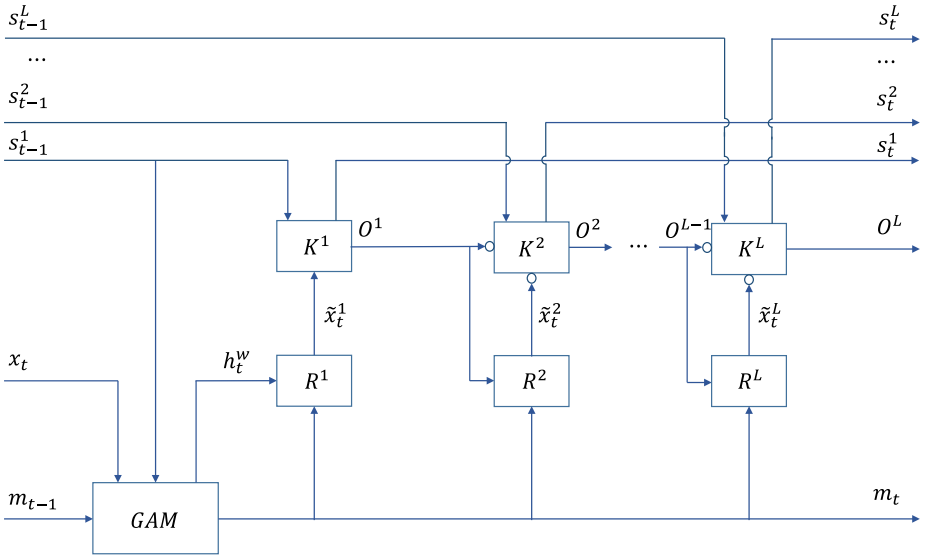


FIGURE 5. Diagram of the proposed hierarchical recurrent neural network with grouped auxiliary memory architecture. Concatenating all the input of a block if there is a ∘ at the place of input.

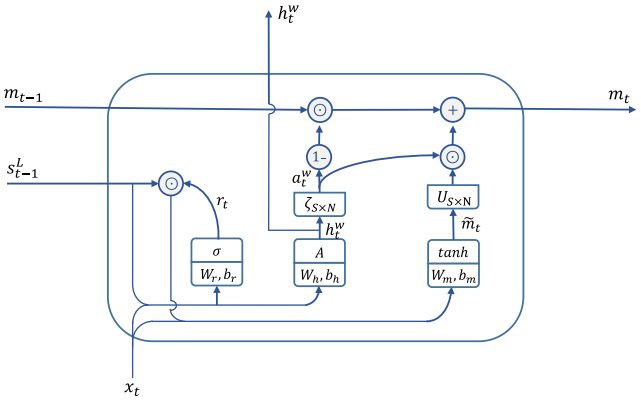


FIGURE 6. Grouped auxiliary memory module.

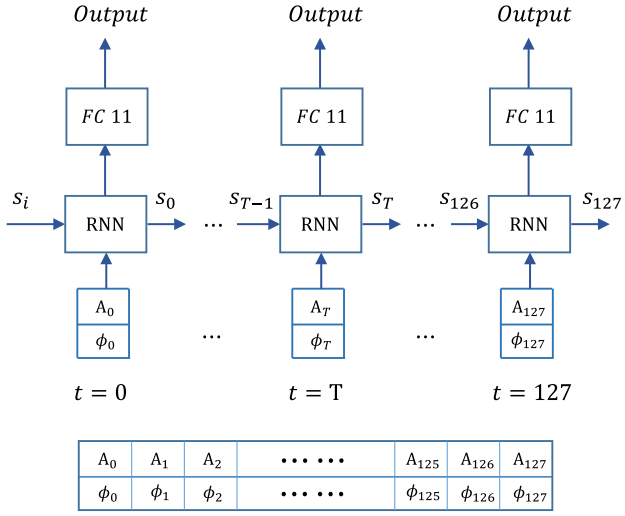


FIGURE 7. Architecture of the whole network with the input rule. RNN denotes the models mentioned above, and FC represented full-connected networks.

L2 normalized and the phase vectors, which were in radians, were normalized between -1 and 1. Finally, we obtained a $R^{2\times128}$ matrix for a single signal sequence. A single group of amplitude and phase was given to the model for each timestep. Details are shown in Fig. 7.

For all the models, we used the Adam optimizer [36] with a minibatchsize of 400 vectors. The learning rate was set to 0.001. Weights were initialized via the Xavier uniform initializer [37], and the models were implemented using Tensorflow [38]. The standard backpropagation through time algorithm was used for the RNN's training. For all the models, dropout of 0.2 was used for each layer. For LSTM, we initialized the forget gate bias to 1.0. This implied that we encouraged the LSTM to write the information into memory at the start. For the GAM–RNN model, a 2 × 20 size was set to GAM, while the size of vector for writing(h_t^w in (12c)) was 20. We also added a dropout of 0.3 for the auxiliary module. GAM-HRNN used 128 units for each layer. And different number of units for each layer were used for different models to keep the parameters of all models are roughly the same. For independently RNN(IndRNN), the recurrent weight is constrained in the range of $|u_n| \in (0, \sqrt[T]{2})$ [29], where T represents the time steps of the sequence. Furthermore, we denoted the GAM-HRNN model with the kernel of GRU as GAM-HRNN-GRU, and this notation was used for all other models with kernels. The initialized forget bias of 1.0 was not used for the GAM-HRNN-LSTM model.

A. CLASSIFICATION ACCURACY ON STANDARD RadioML 2016.10a DATASET

The classification accuracies of all SNRs are presented in Fig. 8 and Fig. 9. The number of model parameters with average accuracy for an SNR range from 0 dB to 20 dB are shown in Table 2. As we can see, 2-layer GAM–GRU exhibited the best accuracy for all SNRs compared to other

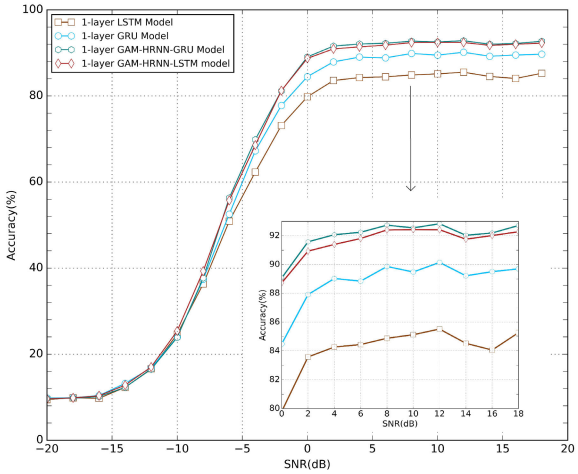


FIGURE 8. Classification accuracy for the proposed 1-layer models at different SNRs.

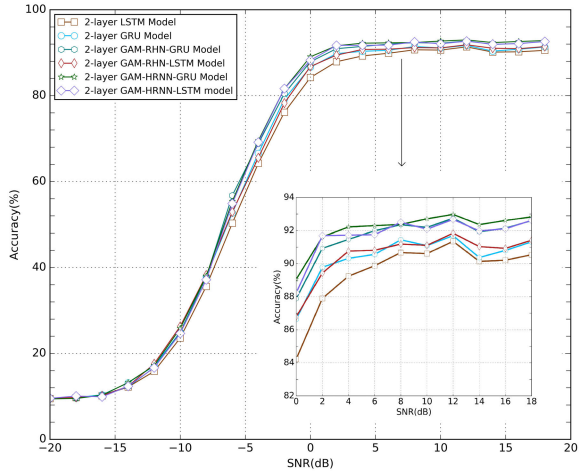


FIGURE 9. Classification accuracy for the proposed 2-layer models at different SNRs.

models, and the 2-layer GRU and LSTM models obtained similar results on the dataset. All the models have insufficient capability for classifying the modulation types at low SNRs. At -18 dB (the lowest SNR), the accuracy is nearly $\frac{1}{11}$, which illustrates that the model returns an almost random category for this SNR (there were a total of 11 categories of various modulation types). The proposed 2-layer GAM-HRNN-GRU model achieved an average accuracy of 92.2 % in the SNR range from 0 dB to 20 dB, with fewer model parameters than the 2-layer LSTM model. Simultaneously, the single-layer GAM-HRNN-GRU obtained 91.6% average accuracy with fewer parameters than the LSTM, which is a positive aspect when considering practical applications. Meanwhile, the accuracy of single-layer GAM-HRNN-GRU is about 8% better than that of the single-layer LSTM. For RHN models, the two-layer RHN-GRU model achieved accuracy that was slightly lower than that of the two-layer GRU. The two-layer RHN-LSTM achieved the worst accuracy, which indicates that the recurrent highway structure was not

TABLE 2. Average accuracy for SNR ranges from 0dB to 20dB results.

Model	Acc.%	Size
1-layer LSTM	84.9%	106091
1-layer GRU	88.8%	80011
1-layer GAM-HRNN-LSTM	91.6%	101335
1-layer GAM-HRNN-GRU	91.6%	82263
2-layer LSTM	90.5%	200075
2-layer GRU	90.6%	192571
2-layer RHN-LSTM	80.1%	249736
2-layer IndRNN	80.3%	210161
2-layer TG-LSTM	90.7%	200075
2-layer RHN-GRU	90.4%	198511
2-layer GAM-RHN-LSTM	85.3%	241391
2-layer GAM-RHN-GRU	91.7%	190091
2-layer GAM-HRNN-LSTM	91.9%	246559
2-layer GAM-HRNN-GRU	92.2%	192031

suitable for this work. The results showed that the hierarchical structures were more efficient than the RHN structures for this problem. Accuracy of 2-layer GAM-HRNN-GRU was higher than that of 2-layer GAM-RHN-GRU with about the same parameters and GAM-HRNN-GRU was more stable. . In summary, simple RNN models, such as LSTM or GRU, could not achieve excellent accuracy, which was probably due to long-term memory problems. The GAM-HRNN model provided the best results, as GAM could efficiently utilize the temporal features of the sequence and then latch information for a long time period. Furthermore, the hierarchical structure with reading shortcuts was more efficient.

We also present confusion matrices at three SNRs to further explain the performance of the proposed model. The results showed that at -8 SNR (Fig. 10.), the model can gradually return the correct category, but the accuracy was still low for practical application. The model shows excellent performance when the SNR is greater than 0 dB (Fig. 11.); nonetheless, the model could not separate AM-DSB and WBFM signals very well, even at high SNRs (Fig. 12.). This is mainly due to the small observation window (0.64 ms of modulated speech per example) and low information rate with frequent silence between words [39]. Distinguishing between QAM16 and QAM64 also suffered from short-time observations over only a few symbols. However, once the constellations were of higher order and shared common points [39], the proposed model performed well. At the three SNRs, compared with QAM16 and QAM64, we could find that the order of the modulation type is an influential factor for the accuracy of the classification. This was true for all investigated methods.

B. GRADIENTS ANALYSIS

To further explain the memory mechanism in GAM-HRNN, we analyzed the gradient changing curves in the training process. The key to being able to learn long-term dependencies is in the control $\frac{\partial L}{\partial s_t}$, which implies keeping the gradients within an appropriate range. In RNNs, the gradient suffers from the vanishing problem in both temporal and spatial dimensions. To further explain the experimental results, we compared $\frac{\partial L}{\partial s_t}$

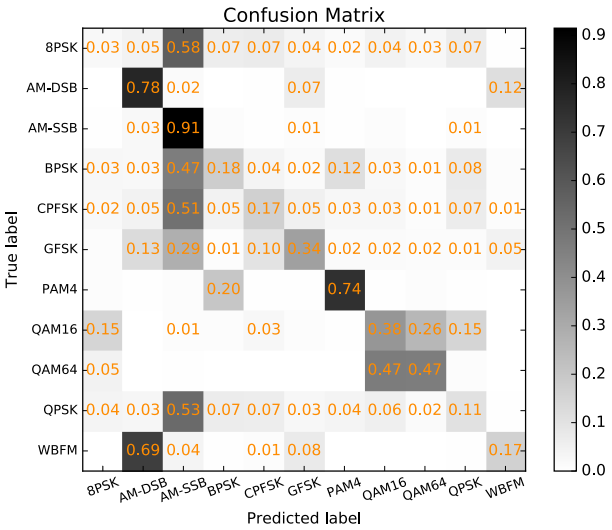


FIGURE 10. Confusion matrix of 2-layer GAM-HRNN-GRU model on RadioML dataset at -8dB signal-to-noise (SNR).

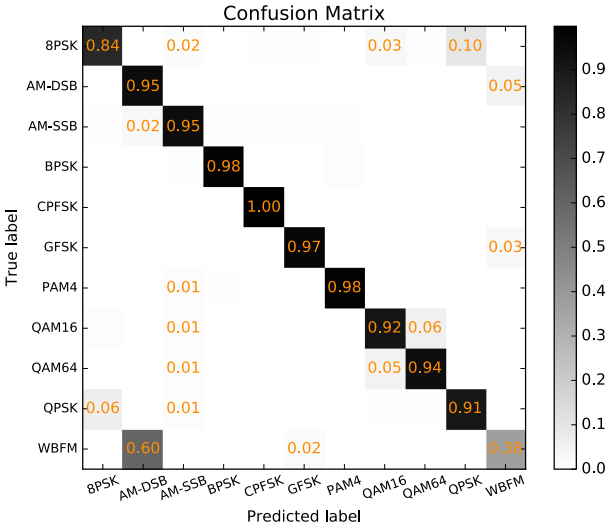


FIGURE 11. Confusion matrix of 2-layer GAM-HRNN-GRU model on RadioML dataset at 0dB signal-to-noise (SNR).

for 1-layer GAM-HRNN-LSTMs and 1-layer LSTM in the training process. The forget biases of LSTM were initialized to 0 for both models.

Curves are presented in Fig. 13. At the beginning of the training, gradients norms of GAM-HRNN-LSTM were at an acceptable level. After 300 updates, GAM-HRNN-LSTM still maintained an appropriate value of gradients along the timesteps, but LSTM suffered from a gradient vanishing problem. Hence, the GAM module can adaptively maintain the gradients at a suitable level for the RNN model. This was good for the training process and the model would eventually obtain a satisfactory result.

C. MODIFIED DATASET

We further evaluated the model on the modified variable length dataset. For AMC modulation classification, the ability

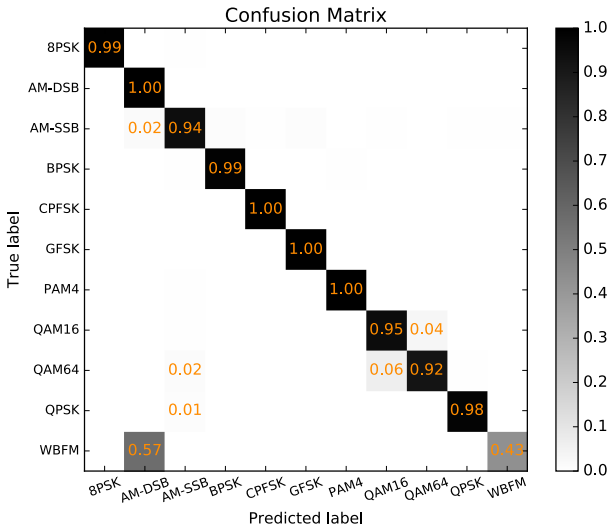


FIGURE 12. Confusion matrix of 2-layer GAM-HRNN-GRU model on RadioML dataset at 18dB signal-to-noise (SNR).

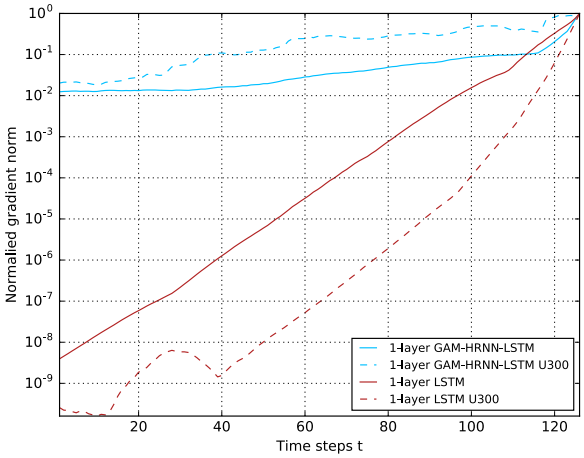


FIGURE 13. Gradients norm of 1-layer models with hidden states at the beginning of training and after 300 updates (U300 means after 300 updates).

of the model to deal with variable length signals with different parameters was also important. In this subsection, we evaluate 2-layer GAM-HRNN-GRU and 2-layer LSTM on the variable length dataset. The details of the dataset are presented above. The configurations of the models were the same as that in subsection A. The models were trained for SNRs ranging from -20 dB to 20 dB, and input sample lengths varied from 128 to 512 samples. The results are presented in Fig. 14.

At low SNRs, both models had poor capability to return the correct criterion for the dataset. With the increases in SNRs, the proposed model performed better than LSTM at all lengths. The accuracy increased as the data obtained by the model accumulated because the model could learn temporal dependencies from the information. Moreover, this requires the ability of the model to retain memory for a long time, which is critical for this problem.

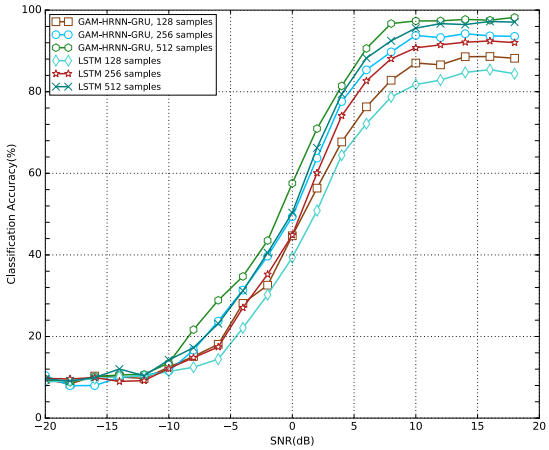


FIGURE 14. Accuracy of variable length of signals of 2-layer GAM-HRNN-GRU and LSTM.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a recurrent structure named GAM-HRNN for AMC problem. Subsequently, we evaluated the GAM-HRNN model on the standard and variable length datasets. Experiments verified that the proposed model exhibited excellent performance on this problem. And our 1-layer model can also give competitive result and beat other models of 2 layers with much fewer parameters. The model had sufficient ability to handle variable length signal inputs with different parameters. We also emphasized on the importance of maintaining the gradient within an appropriate range in the training process for obtaining good results.

There are some limitations to the proposed model. The proposed model has insufficient capability to deal with inputs at low SNRs. In addition, the parameters can be further reduced by modifying the structure of the model or using some pruning methods.

REFERENCES

[1] J. Hamkins and M. K. Simon, "Modulation classification," in *Autonomous Software-Defined Radio Receivers for Deep Space Applications* (Deep Space Communications and Navigation Series). Pasadena, CA, USA: Jet Propulsion Laboratory, California Institute of Technology, 2006, pp. 271–319.

[2] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: Classical approaches and new trends," *IET Commun.*, vol. 1, no. 2, pp. 137–156, Apr. 2007.

[3] S. Huang, Y. Yao, Z. Wei, Z. Feng, and P. Zhang, "Automatic modulation classification of overlapped sources using multiple cumulants," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6089–6101, Jul. 2017.

[4] A. Ali and W. Hamouda, "Advances on spectrum sensing for cognitive radio networks: Theory and applications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1277–1304, 2nd Quart., 2017.

[5] S. Huang, L. Chai, Z. Li, D. Zhang, Y. Yao, Y. Zhang, and Z. Feng, "Automatic modulation classification using compressive convolutional neural network," *IEEE Access*, vol. 7, pp. 79636–79643, 2019.

[6] X. Li, F. Dong, S. Zhang, and W. Guo, "A survey on deep learning techniques in wireless signal recognition," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–12, Feb. 2019.

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[8] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Assoc. Comput. Linguistics, Hum. Lang. Technol. (HLT NAACL)*, vol. 1, 2019, pp. 4171–4186.

[9] D. Zhang, W. Ding, B. Zhang, C. Xie, H. Li, C. Liu, and J. Han, "Automatic modulation classification based on deep learning for unmanned aerial vehicles," *Sensors*, vol. 18, no. 3, p. 924, Mar. 2018.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] W. Luo and F. Yu, "Recurrent highway networks with grouped auxiliary memory," *IEEE Access*, vol. 7, pp. 182037–182049, 2019.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555

[13] J. Collins, J. Sohl-Dickstein, and D. Sussillo, "Capacity and trainability in recurrent neural networks," in *Proc. 5th Int. Conf. Learn. Represent., Conf. Track ((ICLR))*, 2019, pp. 1–17.

[14] J. G. Zilly, R. K. Srivastava, J. Koutnik, and J. Schmidhuber, "Recurrent highway networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 8, pp. 6346–6357, 2017.

[15] V. G. Chavali and C. R. C. M. da Silva, "Classification of digital amplitude-phase modulated signals in time-correlated non-Gaussian channels," *IEEE Trans. Commun.*, vol. 61, no. 6, pp. 2408–2419, Jun. 2013.

[16] J. Yuan, Z. Zhao-Yang, and Q. Pei-Liang, "Communication' signals," in *Proc. IEEE Mil. Commun. Conf. Modulation (MILCOM)*, 2004.

[17] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Trans. Commun.*, vol. 48, no. 3, pp. 416–429, Mar. 2000.

[18] J. Lopatka and M. Pedzisz, "Automatic modulation classification using statistical moments and a fuzzy classifier," in *Proc. 5th Int. Conf. Signal Process.*, 2002, pp. 1500–1506.

[19] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals," *IEEE Trans. Commun.*, vol. 46, no. 4, pp. 431–436, Apr. 1998.

[20] A. E. Shermeh and R. Ghazalian, "Recognition of communication signal types using genetic algorithm and support vector machines based on the higher order statistics," *Digit. Signal Process.*, vol. 20, no. 6, pp. 1748–1757, Dec. 2010, doi: 10.1016/j.dsp.2010.03.003.

[21] M. W. Aslam, S. Member, Z. Zhu, and S. Member, "Automatic modulation classification using combination of genetic programming and KNN," *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 2742–2750, Jun. 2012.

[22] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," *Commun. Comput. Inf. Sci.*, vol. 629, pp. 213–226, Sep. 2016.

[23] S. Peng, H. Jiang, H. Wang, H. Alwageed, and Y.-D. Yao, "Modulation classification using convolutional neural network based deep learning model," in *Proc. 26th Wireless Opt. Commun. Conf. (WOCC)*, no. 1, Apr. 2017, pp. 6–10.

[24] S. Peng, H. Jiang, H. Wang, H. Alwageed, Y. Zhou, M. M. Sebdani, and Y.-D. Yao, "Modulation classification based on signal constellation diagrams and deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 718–727, Mar. 2019.

[25] H. Ma, G. Xu, H. Meng, M. Wang, S. Yang, R. Wu, and W. Wang, "Cross model deep learning scheme for automatic modulation classification," *IEEE Access*, vol. 8, pp. 78923–78931, 2020.

[26] J. Shi, S. Hong, C. Cai, Y. Wang, H. Huang, and G. Gui, "Deep learning-based automatic modulation recognition method in the presence of phase offset," *IEEE Access*, vol. 8, pp. 42831–42847, 2020.

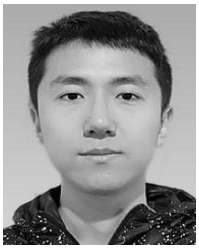
[27] S. Rajendran, S. Member, W. Meert, D. Giustiniano, S. Member, V. Lenders, S. Pollin, and S. Member, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, May 2018.

[28] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," 2013, *arXiv:1312.6026*. [Online]. Available: https://arxiv.org/abs/1312.6026

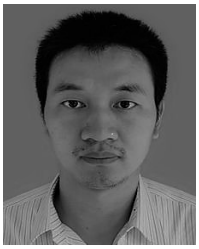
[29] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (IndRNN): Building a longer and deeper RNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, no. 1, Jun. 2018, pp. 5457–5466.

[30] J. Hu and W. Zheng, "Transformation-gated LSTM: Efficient capture of short-term mutation dependencies for multivariate time series prediction tasks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[31] D. Xu, W. Cheng, B. Zong, D. Song, J. Ni, W. Yu, Y. Liu, H. Chen, and X. Zhang, "Tensorized LSTM with Adaptive Shared Memory for Learning Trends in Multivariate Time Series," in *Proc. AAAI*, 2020, pp. 1395–1402. [Online]. Available: http://www.aaai.org
[32] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu, "Data augmentation for deep learning-based radio modulation classification," *IEEE Access*, vol. 8, pp. 1498–1506, 2020.
[33] Y. Wang, J. Yang, M. Liu, and G. Gui, "LightAMC: Lightweight automatic modulation classification via deep learning and compressive sensing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3491–3495, Mar. 2020.
[34] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.
[35] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of Recurrent Network architectures," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, vol. 3, 2015, pp. 2332–2340.
[36] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.
[37] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, May 2010.
[38] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: http://arxiv.org/abs/1603.04467
[39] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.



KE ZANG received the B.S. degree from the Department of Instrument Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2016. He is currently pursuing the Ph.D. degree with Zhejiang University. His research interests include machine learning, and its application in wireless communication and signal processing.



ZHENGUO MA received the B.S. and Ph.D. degrees from the Department of Instrument Science and Technology, Zhejiang University, Hangzhou, China, in 2006 and 2011, respectively. He is currently an Associate Professor with Zhejiang University. His research interests include heterogeneous computing architecture and wireless communication.