## RESEARCH

## **Open Access**

# A novel deep learning automatic modulation classifier with fusion of multichannel information using GRU



Sigi Sun<sup>1</sup> and Yongyu Wang<sup>2\*</sup>

\*Correspondence: 805910302@qq.com

<sup>1</sup> School of Electrical and Electronic Engineering, North China Electric Power University, Beijing, China <sup>2</sup> School of Information and Electronic Engineering, Shandong Technology and Business University, Yantai, China

## Abstract

Automatic modulation classification (AMC) plays a vital role in modern communication systems, which can support wireless communication systems with limited spectrum resource. This paper proposes an AMC method, which integrates gated recurrent unit (GRU) and convolutional neural network (CNN) to utilize the complementary input features of received signals for spatiotemporal feature extraction and classification. Different from other state-of-the-art (SoA) frameworks, the proposed AMC classifier, named as fusion GRU deep learning neural network (FGDNN), aggregates firstly temporal features with GRUs and then extracts spatial features with CNNs. The GRUs can store temporal dynamic features, and facilitate to capture the characteristics of correlation and dependence among input features. The method is tested extensively with comparisons in order to verify its effectiveness. Experiment results show that the recognition rates of our method outperform other deep learning frameworks.

Keywords: Automatic modulation classification, Multichannel, Input features, Temporal dynamics, Gated recurrent unit

## 1 Introduction

Automatic modulation classification (AMC) refers to a signal processing mechanism through which a received signal's most likely modulation scheme is determined using classification with minimal information about the signal configuration [1]. Modulation classification is a major issue in many communication systems with both military and civilian applications, such as spectrum monitoring [2], dynamic spectrum access [3, 4], IoT attack detection [5], and cognitive radio [6, 7]. Traditional AMC methods include likelihood-based (LB) classifiers, which has a high dependence on prior knowledge, and consequently the feature-based automatic modulation classification (FBAMC) method was widely studied. The FBAMC method involves two stages. In the first stage, a received signal's features are extracted and then provided to a labeling stage that functions as the second processing stage. Features extracted in the first stage correspond primarily to the signal's characteristics. These features include signal spectral-based [8], wavelet transform-based, high-order statistics-based, and cyclostationary analysis-based features [6].



© The Author(s) 2023. Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http:// creativecommons.org/licenses/by/4.0/

During the last decade, deep learning (DL) has made great achievements in many fields, such as image processing, natural language processing, wireless communications, and so on. So many scholars tried to apply DL methods to the FBAMC method for improving classification accuracy [9-21]. Most of the current methods based on DL are directly borrowed from the field of image processing or natural language processing. We present the related DL-based AMC works in two aspects. Firstly, we survey related works whose contributions are mainly about achieving a high classification performance through combining different network layers or deepening the network depth.

Meng et al. proposed an earlier DL-based AMC method [9], which is CNN-based AMC and can outperform the FBAMC and has a faster computing speed with parallel computation. An efficient CNN architecture, MCNet, is also proposed in [10], which is concatenated or added by multiple convolution blocks with asymmetric convolution kernel, and can effectively capture the spatial correlation of modulation signals with the increase of the number of convolution blocks. Another optimized CNNAMC network named SBCNN also has a high robustness on a complex dataset by designing an optimal filter size for improving the prediction accuracy [11]. Lin et al. designed a time-frequency attention mechanism for CNNAMC to learn which frequency, channel and time features are more meaningful in networks for modulation classification [12], which outperforms other attention mechanisms. Multiple CNN models are trained in [13] for multitask learning under different SNRs, and each CNN model shares the model weight with other CNN models, which can be applied in realistic noise scenarios and achieves better generalization and robustness. Besides CNN, there are other contributions in the field of image processing that have been applied to AMC. P. Qi et al. exploited a waveform-spectrum multimodal fusion method to realize deep residual network (ResNet)-based AMC method [8], which can efficiently distinguish among sixteen modulated signals.

Moreover, recurrent neural network (RNN) is well-known for its capability to learn from the temporal data. Hu et al. realize the DNN-based modulation classifier using multiple long short-term memory (LSTM) layers and fully connected layers and show that the proposed RNN-based classifier is robust to the uncertain noise conditions [14]. In addition, modulated signals have not only spatial characteristics, but also temporal characteristics. A CNN-LSTM-based dual-stream structure is proposed in [15], which extracts spatial features with CNN and then extracts temporal features with LSTM. Another three-stream deep learning framework is proposed in [16], which has efficient convergence speed and achieves improved recognition accuracy, especially for the signals modulated by higher dimensional schemes such as 16-QAM and 64-QAM. LSTM attains high classification accuracy with fewer signal symbols, but it requires a long training process due to its recurrent structure [17]. Hong et al. [18] proposed a classifier composed of a simple convolution layer and gated recurrent unit (GRU), but the performance gain is not obvious compared with one-layer LSTM.

Secondly, we survey related deep learning works based on input data formats, that include in-phase/quadrature (I/Q)-based, amplitude/phase (A/P)-based, I/Q + A/P-based, constellation-based, and other formats.

- (1) *I/Q* Based: In [19], a seven-layer convolutional neural network (CNN), a seven-layer residual networks (ResCNN), a seven-layer densely connected networks (DenseC-NNs), and convolutional long short-term deep neural network (CLDNN) are evaluated and shown that CLDNN is the most suitable framework for the AMC task. Different from the CLDNN used in [19], where only the last step outputs from long short-term memory (LSTM) layer are used as features, a temporal RNN attention layer over outputs of all time steps is proposed to summarize temporal information in [14]. Though most recent DL-based AMC works are based on real-valued operations and representations, Tu et al. demonstrated the high potential of complex-valued networks for AMC in [22], and their results validate the superior performance in AMC achieved by the complex-valued networks.
- (2) A/P Based: Different from the I/Q, A/P is the polar coordinate counterpart transferred by I/Q. In [1], a novel deep learning and polar transformation framework for an adaptive automatic modulation classification is presented, which is composed of CNN architecture and channel compensation network. Huang et al. [5] proposed a novel gated recurrent residual neural network, which is made up of ResNet and GRU layers.
- (3) *I*/*Q* + *A*/*P* Based: To explore the feature interaction and spatial-temporal properties of raw complex signals, a CNNLSTM-based dual-stream structure (DSCLDNN) is proposed in [15]. As for the fusion operation in DSCLDNN, the outer product is adapted. In [20], Chang et al. proposed multitask learning deep neural network based on the differences and characteristics of *I*/*Q* and *A*/*P*.
- (4) Constellation Based: Peng et al. [21] used two convolutional neural network (CNN)-based DL models, AlexNet and GoogLeNet for AMC task directly. In [23], Lin et al. developed a framework to transform complex-valued signal waveforms into images with statistical significance, termed contour stellar image, which can convey deep level statistical information from the raw wireless signal waveforms while being represented in an image data format. Although the AMC using constellation map can directly benefit from the visual CNN networks, the temporal characteristics of signal sequence is neglected, which results in the degradation of classification accuracy.

However, utilizing multiple formats of a signal is a complex task in practical applications and different input information might affect the performance of the framework. While some conclusions from previous work are interesting and helpful, how to fuse different formats of signals are still required to explore for designing networks in accordance with characteristics of modulated signals. The main contributions of this article are summarized as follows.

(1) Compared with conventional model-driven modulation classification methods, we propose a novel deep learning framework to efficiently fuse multichannel information and explore spatial and temporal characteristics existing in modulated signals. Specially, it has three functional parts: fusion of input features and temporal characteristics mapping, spatial features extraction, and fully connected classifier. Since the classification process consists of offline training and online classification

stages, the designed classifier is capable of identifying the received signal samples corrupted by different channel conditions and is robust to the uncertainties of noise conditions.

(2) As for RNNs, GRU layer is more promising than LSTM for extracting temporal information in signal data [24]. In this paper, we modify the standard GRU model to implement better fusion of multiple input features. The modification to the conventional GRU model involves model parameter sharing and reduction of the number of model parameters. The modified GRU can expand the dimension of the signal features, and the strong temporal correlation stored in stacked GRU layers can help the following CNN layers to extract spatial features more efficiently. Moreover, the designed classifier is flexible in practice as it can process different input lengths during the offline training stage and online classification stage in practice.

The main contribution of this article includes a novel deep learning framework to efficiently fuse multichannel information and explore spatial and temporal characteristics existing in modulated signals. Specially, it has three functional parts: fusion of input features and temporal characteristics mapping, spatial features extraction, and fully connected classifier. The remainder of this article is organized as follows. Section 2 introduces the motivation and presents the deep-learning AMC framework with fusion of multichannel information using GRU. Experiments and discussion are conducted in Sect. 3, and the results are analyzed in detail. Finally, Sect. 4 concludes this article.

## 2 Method

#### 2.1 Motivation

This paper considers a single-input single-output communication system, and the received signal r(t) can be written as in (1). s(t) is the modulated signal from the transmitter, which is a time series signal of either a continuous signal or a series of discrete bits modulated onto a sinusoid with either varying frequency, phase, amplitude, trajectory, or some permutation. h(t) is the channel impulse response, reflecting some path loss or constant gain term on the signal, and n(t) is an additive Gaussian white noise process reflecting thermal noise.

$$r(t) = s(t) * h(t) + n(t)$$
 (1)

The received signal r(t) is sampled *N* times at a rate  $f_s = 1/T_s$  by the analog to digital converter, which generates the discrete-time observed signal r(n). The in-phase component (*I*) and quadrature component (*Q*) of r(n), called I/Q data, are given as

$$I = \{ \operatorname{real}[r(n)] \}_{n=0}^{N-1}, \text{ and } Q = \{ \operatorname{imag}[r(n)] \}_{n=0}^{N-1}$$
(2)

The amplitude/phase (A/P) data are generated by transferring the I/Q data from the cartesian coordinate system to the polar coordinate system [1].

Most of AMC methods are based on mono-modal input information, i.e., only the I/Q data [14, 19] or the A/P data [1, 5] are used. Basically, the I/Q data can be utilized to learn temporal characteristics from raw complex signals, and the A/P data will provide effective temporal and spatial features related to the amplitude and phase of raw complex signals. They are belonging to the waveform features of the received signals.

However, the ability of I/Q data to accurately represent signal characteristics may be affected in low SNRs, and the A/P data can recover some signal features by using the amplitude and the phase.

One intuitive idea is that, with the fusion of multichannel inputs, the extracted feature is more discriminating than the mono-modal representation. Motived by differences and complementary features between I/Q data and A/P data, a deep learning framework using GRUs is proposed to implement the fusion of I/Q and A/P data, as shown in Fig. 1.



**Fig. 1** The structure of the proposed framework. This figure presents the architecture of the proposed framework. It consists of three functional parts: fusion of input features and temporal characteristics mapping, spatial features extraction, and fully connected classifier. Part-A is the stacked GRU layers, which includes two GRU layers. Part-B includes three CNN layers and two max-pooling layers, and the number and kernel size of convolutional filters are given. Part-C includes two fully connected (FC) layers, and the Softmax activation function is adopted at the last FC layer

## 2.2 Method description

The overall process of our proposed classification method is shown in Fig. 2, which consists of three functional parts: fusion of input features and temporal characteristics mapping, spatial features extraction, and fully connected classifier.

Firstly, a preprocessing is implemented to convert signals from I/Q data to A/P data. Secondly, stacked GRU layers are exploited to facilitate fusion of multichannel input features, and the more discriminating feature is obtained by learning the temporal pattern. Then, outputs of the stacked GRU layers are fed to the following CNNs for spatial feature extraction. The spatial feature extraction consists of three CNN layers, which are followed by two fully connected (FC) layers. The Softmax activation function is adopted at the last FC layer to obtain the probability of modulation schemes.

Gated recurrent unit (GRU) is proposed as a replacement for LSTM to solve the vanishing/exploding gradient problem, which has been encountered by recurrent neural networks (RNNs) while dealing with long-term dependencies. GRUs have fewer training parameters than LSTMs and require less memory and training time than LSTMs. The critical difference is that GRUs combine input and forget gates of LSTM into an update gate and discard cell state (see Fig. 3).

In this section, we modify the standard GRU model to implement better fusion of multiple input features. The modification to the conventional GRU model involves



**Fig. 2** The process map. This figure presents the overall process of the proposed classification method. The deep learning AMC can be processed in two processes, i.e., the offline training process and the online test process. In training process, preprocessing receives the signals and then transforms them from *I/Q* data to *A/P* data. After multichannel input fusion and features extraction, classifier learns their modulate type. In test process, the well-trained model receives the signal and decide the modulation scheme



Fig. 3 The architecture of Gated Recurrent Unit. This figure presents the architecture of gated recurrent unit (GRU). GRUs include an update gate and a reset gate, and discard cell state compared with LSTM

model parameter sharing and reduction of the number of model parameters.Specifically,  $u_t$ , $r_t$ , and  $n_t$  share the input-to-hidden parameters, in which the subscripts u, r, and n indicate the update gate, reset gate, and candidate state, respectively. That is, the input term in GRU is expressed by

$$s_t = W x_t \tag{3}$$

where  $x_t$  denotes the current input vector, and W is the shared input-to-hidden parameter for  $u_t$ ,  $r_t$ , and  $n_t$ . Thus, the calculation procedures for GRU are

$$u_t = \sigma_g(Wx_t + U_u h_{t-1}) \tag{4}$$

$$r_t = \sigma_g(Wx_t + U_r h_{t-1}) \tag{5}$$

$$n_t = \sigma_h[Wx_t + r_t \odot (U_n h_{t-1})] \tag{6}$$

$$h_t = (1 - u_t) \odot n_t + u_t \odot h_{t-1} \tag{7}$$

where  $h_{t-1}$  denotes the previous hidden state vector,  $U_u$ ,  $U_r$ ,  $U_n$  are hidden-to-hidden parameters,  $\sigma_g$  is the logistic sigmoid function,  $\sigma_h$  is the hyperbolic tangent function, and  $\odot$  operator represents an element-wise multiplication. As shown in Fig. 3,  $u_t$  is the output of the update gate,  $r_t$  is the output of the reset gate, and  $n_t$  is the output of the hyperbolic tangent function in Fig. 3.

The modified GRU is capable of processing the representative features with the arbitrary length for modulation classification and can expand the dimension of the signal features. As described in Fig. 1, I/Q data and A/P data are fed to the stacked GRU layers, which includes two layers of GRU with m hidden neurons, respectively, and a dropout layer with rate of 0.2. The kernel is initialized using a Glorot uniform initializer, and a recurrent kernel initialized using an orthogonal initializer. The bias is initialized to be zero. Due to the recurrent structure of GRU, the input to first GRU layer is a  $N \times 4$  matrix, where the first dimension N corresponds to the sample length (N = 128, 256, 512, 1024), and the second dimension 4 is the total number of input channel (i.e., I, Q, A and P). The sample length N is regarded as the time step of GRU and this parameter changes dynamically, which determines that the GRU can process data with arbitrary length. The output of the first GRU layer, the input and output of the second GRU layer are all  $N \times m$  matrix.

There are multiple ways to summarize the final output of a RNN layer before feeding into the subsequent layers. The most popular one is to use the output at the final time step as the summary of all temporal information and as the output of this layer [19]. It can effectively reduce the number of parameters in the subsequent neutral network layer. Another way is to use a weighted sum of outputs at all time steps as the final output [14]. Based on high-order attention mechanism, high-order convolutional attention networks for radio signal expression and feature correlation learning are proposed in [25]. How to effectively compute the weight remains an open issue and it may bring more complexity. Both of above methods apparently lose part of the information. In order to fuse the multichannel inputs for further feature extraction and classification, the aggregated information from all time steps will be used in this paper. The outputs at all time steps are concatenated and truncated as output of the stacked GRUs. As a result, temporal correlation data is expanded and reshaped as  $m \times 1 \times 128$  tensors.

Then, three one-dimensional convolution layers are adopted to extract spatial features from the temporal correlation data. For convolution layers, the convolution processing of the feature map by the convolution kernel is the key step of feature extraction. Convolutional filters of larger width (kernel size) can effectively compute higher n-gram features at each time step. This is important for maintaining the training quality of the AMC classifier when it is going through the training process for low-SNR received samples modulated with high-order modulation schemes. The numbers of kernels used in the first, second, and third convolution layer are 32, 64, and 128, respectively, with the same kernel size of  $1 \times 7$ . And the rectified linear unit (ReLU) is used as the activation function. In the CNN, a pooling layer is mainly used to compress features extracted by the convolution layer. Here two maximum pooling layers with stride = 2 are adopted after the first and second convolution layers to further reduce dimension of time series.

A batch normalization process is implemented after every CNN layer to prevent effects of large noise variance present in the extracted feature map, especially while training in the case of low-SNRs. This batch normalization process, which batch normalizes the activations by using empirical estimates of their means and variances, is also helpful to prevent overfitting in the training process.

After learning the features by the three convolution layers,  $128 \times 1$  feature vector is obtained. It will be used as the input of FC layers, which are composed of FC1 and FC2 with 64 and 24 hidden neurons, respectively. Besides, the dropout method is employed on both FC layers with a dropout rate p = 0.5 to avoid overfitting.

Regarding the network configuration reported in Table 1, the network has 259,072 trainable parameters in total. We use categorical cross-entropy as a loss function and Adam optimizer for all frameworks. The learning rate is initialized at 0.01 (dropped 90% after 10 epochs). The batch size for each iteration is set to 64. All experiments are carried out in Pytorch, and the performance is measured on a system equipped by a 3.40-GHz CPU, 16GB RAM, and a single NVIDIA GeForce GTX 1080Ti GPU.

	Output volume	Description	
input	$N \times 4$	4 N = 128, 256, 512, 1024	
GRU1	N×128	128 hidden neurons	
GRU2	128 × 1 × 128	128 hidden neurons	
Conv1	32 × 1 × 62	(32, 1, 7), s = 2, p = 1	
Pooling1	32 × 1 × 31	max-pooling, $s = 2$	
Conv2	64 × 1 × 14	(64, 1, 7), s = 2, p = 1	
Pooling2	64 × 1 × 7	max-pooling, $s = 2$	
Conv3	128 × 1	(128, 1, 7), s = 1, p = 0	
FC1	64	Fully connected with 64 neurons	
FC2	24	Fully connected with 24 neurons	

Table 1 Configuration of the proposed architecture

## 3 Results and discussion

## 3.1 Dataset description

DeepSig: RadioML [4] is one of the most challenging datasets of modulation classification. The newest version includes both synthetic simulated channel effects, such as carrier frequency offset, symbol rate offset, delay spread, thermal noise, and over-the-air measurements of 24 digital and analog modulation formats. The dataset has over 2.5 million 1024-length frames of modulation signal, signal-to-noise ratios (SNR) varying from - 20 to + 30 dB.

In the experiments, a part of RadioML2018.01 dataset is used as training and testing benchmark (SNR ranging from -10 to +20 with 2 dB apart). There are 4096 frames per modulation-SNR combination, where some 80% for training and the remainder for testing.

## 3.2 Classification accuracy

In this section, extensive experiments are carried out to explore the impact of different network parameters and modules combination on the fusion of multichannel information and the performance of the proposed framework FGDNN. Figure 4 illustrates the classification accuracy of the proposed FGDNN with different sample lengths. GRU is capable of dealing with input of arbitrary lengths due to the parameter sharing mechanism. Such characteristics is very suitable for automatic modulation classification when mismatch between the lengths of testing signal sequences and training signal sequences occurs. After the AMC classifier is trained using signal sequences with length N = 128, 256, 512, 1024, respectively, we evaluate the performance of it using signal sequences with N = 128. As Fig. 4 shows, the classification accuracy of our proposed AMC classifier is up with increasing in length of signal sequences. This shows the feasibility and



**Fig. 4** Classification accuracy of FGDNN for different sample lengths.presents the classification accuracy of the proposed FGDNN with different sample lengths, N = 128, 256, 512, 1024. After the AMC classifier is trained using signal sequences with length N = 128, 256, 512, 1024, respectively, we evaluate the performance of it using signal sequences with N = 128. Figure 4 shows that the classification accuracy of the proposed AMC classifier is up with increasing in length of signal sequences



**Fig. 5** Performance comparison of FGDNN with different neurons. This figure presents the performance comparison of GRU with different hidden neurons, m = 16, 32, 64, 128, 256. The sample length is set as 128 here. This figure shows that GRU with 128 hidden neurons outperforms the others and further increasing the number of hidden neurons cannot improve the performance of the network. When m = 16, the Classification Accuracy is the worst

the convenience of the proposed method to classify the modulation scheme of signal sequences with different lengths.

Since different parameters utilized in the FGDNN perform different classification accuracies, the performance comparison of GRU with different hidden neurons is shown in Fig. 5. According to the comparison criterion, the sample length is set as 128 below. As in Fig. 5, it can be seen that GRU with 128 hidden neurons outperforms the others and further increasing the number of hidden neurons cannot improve the performance of the network, but ends with more computational complexity. On the other hand, the network suffers poor generalization when m is small. In other words, the number of hidden neurons in GRU layers can expand the multichannel inputs information and diversify the features for performance improvement. In experiments, we explore the effects of different GRU layers on classification performance to test and verify the effectiveness of two-layer stacked GRUs. The result suggests that one-layer GRU is not as good as several-layer stacked GRUs, and that two-layer stacked GRUs reach a better classification accuracy at high SNRs. When the number of GRU layer continues increasing, the simulation effect does not improve. Furthermore, the number of hidden neurons in each GRU laver plays a critical role in our framework, which is usually big. When more than two GRU layers with big number of hidden neurons is chosen, it dramatically increases the network capacity and the computational complexity. Thus, the framework based on two-layer stacked GRUs is suitable and chosen.

In experiments, we test three different kinds of frameworks, referring to as FGDNN with and without GRU layers, without CNN layers, as shown in Fig. 6. It is clear that FGDNN with GRU layers outperforms the others. The former yields almost 2 dB gain



**Fig. 6** Performance comparison for FGDNN with and without GRU layers, without CNN. This figure presents performance comparison for FGDNN with and without GRU layers, without CNN. It shows that FGDNN with GRU layers outperforms the others. The FGDNN yields almost 2 dB gain over the one without CNN layers

over the one without CNN layers. The obviously degraded recognition performance of the one without GRU layers at high SNR illustrates the importance of temporal modeling of input data. The missing temporal information results in the performance loss since the received signal samples are highly correlated in the time domain due to the channel effects.



**Fig. 7** Classification performance comparison among FGDNN and other SoA frameworks. This figure presents the comparison result among FGDNN and five frameworks from [14–18], here named as LSTM-FC, CNN-LSTM2, CNN-LSTM2, and GRU2, respectively. As of network input, CNN-LSTM2 uses both *I/Q* and *A/P* data, LSTM2 uses only *A/P* data as input, and the others utilize *I/Q* data directly. The FGDNN outperforms the other frameworks, especially at low SNRs. Specifically, FGDNN is better than CNN-LSTM2 and CNN-LSTM by 3% and 2% at + 10 dB SNR. This figure also shows that AMC frameworks using *A/P* data as input get better result in low SNR range

	Trainable parameters (K)	Inference time (ms)
FGDNN	253	0.265
LSTM-FC	321	0.248
CNN-LSTM2	1117	0.477
CNN-LSTM	397	0.311
LSTM2	196	0.201
GRU2	148	0.183

Table 2	Comparison	of com	putation	complexity
---------	------------	--------	----------	------------

Then, we compared the FGDNN with five frameworks from [1, 14, 16–18], here named as LSTM-FC, CNN-LSTM2, CNN-LSTM, LSTM2, and GRU2, respectively. And the comparison result is given in Fig. 7. As of network input, CNN-LSTM2 uses both I/Q and A/P data, LSTM2 uses only A/P data as input, and the others utilize I/Q data directly. With high-relevant temporal information acquired from the fusion of multichannel inputs, FGDNN outperforms the other frameworks, especially at low SNRs. Specifically, FGDNN is superior to the CNN-LSTM2 and CNN-LSTM models in terms of classification accuracy, for instance, better than them by 3% and 2%, respectively, at + 10 dB SNR. It also can be observed from Fig. 7 that AMC frameworks using A/P data as input get better result in low SNR range.

Regarding the computational complexity, the network capacity (the number of trainable parameters) and the average prediction time are summarized in Table 2. As shown in Table 2, the number of trainable parameters in the FGDNN is larger than LSTM2 and GRU2 but smaller than the other models. The tradeoff between improved recognition accuracy and computational complexity is still acceptable.

## 3.3 Information fusion analysis

In the stacked GRU layers, which consists of two layers of GRU, the outputs of the previous layer are sent directly into the second layer, which helps to extract the temporal representations in a hierarchical manner. The inherent non-linearity and recurrent structures makes understanding the information fused by stack GRU layers sort of difficult. In order to obtain better insightful understanding, we use visualization techniques to show feature maps obtained by GRU. These visualizations can help to understand how stack GRU layers behave for multi-channel inputs. Figure 8 presents the input features and the temporal activation of stack GRU layers in the trained model for a 16QAM input signal with 18 dB SNR.

It can be noticed in Fig. 8 that activations of many time steps correspond to the amplitude and phase changes in the input waveform. With aggregating multi-channel inputs, the second layer stores much long term dependencies from the temporal representations obtained by the first layer. So the strong temporal correlation in feature maps can help the following CNN layers to extract spatial features more efficiently.



**Fig. 8** The (top) input and (bottom) output of stacked GRU layers. This figure presents the input features and the output activation of stack GRU layers for a 16QAM modulated signal with 18 dB SNR. The top figure is the input of stacked GRU layers, including in-phase/quadrature (*I/Q*) data, amplitude/phase (*A/P*) data, and the bottom one is the output of the second GRU layer along time step (N = 128). This figure shows that activations of many time steps correspond to the amplitude and phase changes in the input waveform. The visualization techniques can help us to understand how stack GRU layers behave for multichannel inputs

## 4 Conclusion

During the last decade, deep learning (DL) has been applied to the FBAMC method. We present the related DL-based AMC works in two aspects. Firstly, we survey related works whose contributions are mainly about achieving a high classification performance through combining different network layers or deepening the network depth. Secondly, we survey related deep learning works based on input data formats, that include I/Q-based, A/P-based, I/Q + A/P-based, constellation-based, and other formats. Aiming to improve classification accuracy, combination of different network layers and multichannel input information are studied in this paper to learn spatial and temporal

characteristics from each signal sequence. Based on that, a novel data-driven AMC method with fusion of multichannel information using GRUs is proposed. The proposed method utilizes GRU to fuse multichannel inputs and expand the dimension of the signal features, and takes advantage of CNN to extract features and classify. It turns out that the strong temporal correlation in feature maps obtained by GRU can help the following CNN layers to extract spatial features more efficiently. The method is tested extensively with comparisons in order to verify its effectiveness and superiority. The comparison shows that the recognition rates of our method outperform other deep learning frameworks. Specifically, FGDNN is superior to the CNN-LSTM2 and CNN-LSTM models in terms of classification accuracy, for instance, better than them by 3% and 2%, respectively, at + 10 dB SNR. Our future research direction involves the modification of the structure of the proposed FGDNN framework to respectively achieve lower execution latency and higher classification accuracy in order to improve the efficiency of the proposed framework.

#### Abbreviations

AMC	Automatic modulation classification
LB	Likelihood-based
FB	Feature-based
DL	Deep learning
CNN	Convolutional neural network
RNN	Recurrent neural network
LSTM	Long short-term memory
GRU	Gated recurrent unit
SoA	State-of-the-art
IoT	Internet of things
I/Q	In-phase and quadrature
A/P	Amplitude/phase
SNR	Signal-to-noise ratio
QAM	Quadrature amplitude modulation
ResNet	Residual network

#### Author contributions

SQS had contributions to algorithm design, experimental simulations, and paper writing. YYW had contributions to theoretical design and analysis, and paper writing. Both authors read and approved the final manuscript.

#### Funding

This study was supported by the Natural Science Foundation of China (NSFC) under Grant No. 62173290 and the Doctoral Fund of Shandong Technology and Business University [B5201620].

#### Availability of data and materials

The dataset is downloaded from https://www.deepsig.ai/ datasets. The name of dataset is RadioML2018.01.

#### Declarations

#### **Competing interests**

The authors declare that they have no competing interests.

Received: 21 February 2023 Accepted: 17 July 2023 Published online: 22 July 2023

#### References

- 1. P. Ghasemzadeh, S. Banerjee, M. Hempel, H. Sharif, A novel deep learning and polar transformation framework for an adaptive automatic modulation classification. IEEE Trans. Veh. Technol. **69**(11), 13243–13258 (2020)
- Q. Zhou, R.H. Zhang, F.P. Zhang, X.J. Jing, An automatic modulation classification network for IoT terminal spectrum monitoring under zero-sample situations. EURASIP J. Wirel. Commun. Netw. (2022). https://doi.org/10.1186/ s13638-022-02099-2
- T.J. O'Shea, J. Corgan, T.C. Clancy, Convolutional radio modulation recognition networks. Proc. Int. Conf. Eng. Appl. Neural Netw. 213–226 (2016)

- T.J. O'Shea, T. Roy, T. Clancy, Over-the-air deep learning based radio signal classification. IEEE J. Sel. Top. Signal Process. 12(1), 168–179 (2018)
- S. Huang, R. Dai, J.J. Huang et al., Automatic modulation classification using gated recurrent residual network. IEEE Internet Things J. 7(8), 7795–7807 (2020)
- P. Ghasemzadeh, S. Banerjee, M. Hempel, H. Sharif, Performance evaluation of feature-based automatic modulation classification, in *Proceedings of IEEE 12th International Conference on Signal Processing and Communication Systems* (ICSPCS) (2018), pp. 1–5
- P. Ghasemzadeh, M. Hempel, H. Sharif, GS-QRNN: a high-efficiency automatic modulation classifier for cognitive radio IoT. IEEE Internet Things J. 9(12), 9467–9477 (2022)
- P. Qi, X. Zhou, S. Zheng, Z. Li, Automatic modulation classification based on deep residual networks with multimodal information. IEEE Trans. Cogn. Commun. Netw. 7(1), 21–33 (2021)
- F. Meng, P. Chen, L. Wu, X. Wang, Automatic modulation classification: a deep learning enabled approach. IEEE Trans. Veh. Technol. 67(11), 10760–10772 (2018)
- T. Huynh-The, C. Hua, Q. Pham, D. Kim, MCNet: an efficient CNN architecture for robust automatic modulation classification. IEEE Commun. Lett. 24(4), 811–815 (2020)
- S.H. Kim, C.B. Moon, J.W. Kim, D.S. Kim, A hybrid deep learning model for automatic modulation classification. IEEE Wirel. Commun. Lett. 11(2), 313–317 (2022)
- 12. S. Lin, Y. Zeng, Y. Gong, Learning of time-frequency attention mechanism for automatic modulation recognition. IEEE Wirel. Commun. Lett. **11**(4), 707–711 (2022)
- Y. Wang, G. Gui, T. Ohtsuki, F. Adachi, Multi-task learning for generalized automatic modulation classification under non-Gaussian noise with varying SNR conditions. IEEE Trans. Wirel. Commun. 20(6), 3587–3596 (2021)
- S. Hu, Y. Pei, P.P. Liang, Y.C. Liang, Deep neural network for robust modulation classification under uncertain noise conditions. IEEE Trans. Veh. Technol. 69(1), 564–577 (2020)
- Z. Zhang, H. Luo, C. Wang, C. Gan, Y. Xiang, Automatic modulation classification using CNN-LSTM based dual-stream structure. IEEE Trans. Veh. Technol. 69(11), 13521–13531 (2020)
- J. Xu, C. Luo, G. Parr, Y. Luo, A spatiotemporal multichannel learning framework for automatic modulation recognition. IEEE Wirel. Commun. Lett. 9(10), 1629–1632 (2020)
- S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, S. Pollin, Deep learning models for wireless signal classification with distributed low cost spectrum sensors. IEEE Trans. Cogn. Commun. Netw. 4(3), 433–445 (2018)
- 18. D. Hong, Z. Zhang, X. Xu, Automatic modulation classification using recurrent neural networks, in *Proceedings of 3rd IEEE International Conference on Computing and Communications* (2017), pp. 695–700
- X. Liu, D. Yang, A.E. Gamal, Deep neural network architectures for modulation classification, in Proceedings Asilomar Conference on Signals, Systems, and Computers (2017), pp. 915–919
- S. Chang, S. Huang, R. Zhang, Z. Feng, L. Liu, Multi-task learning based deep neural network for automatic modulation classification. IEEE Internet Things J. 9(3), 2192–2206 (2022)
- S.L. Peng, H.Y. Jiang, H.X. Wang et al., Modulation classification based on signal constellation diagrams and deep learning. IEEE Trans. Neural Netw. Learn. Syst. 30(3), 718–727 (2019)
- Y. Tu, Y. Lin, C. Hou, S. Mao, Complex-valued networks for automatic modulation classification. IEEE Trans. Veh. Technol. 69(9), 10085–10089 (2020)
- Y. Lin, Y. Tu, Z. Dou et al., Contour stella image and deep learning for signal recognition in the physical layer. IEEE Trans. Cogn. Commun. Netw. 7(1), 34–46 (2021)
- J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modelling. (2014). https://doi.org/10.48550/arXiv.1412.3555
- 25. D. Zhang, Y. Lu, Y. Li et al., High-order convolutional attention networks for automatic modulation classification in communication. IEEE Trans. Wirel. Commun. **14**(8), 1–12 (2021)

### **Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

#### Submit your next manuscript at > springeropen.com