electronics

MDPI

Article

Automatic Modulation Recognition Based on Multimodal Information Processing: A New Approach and Application

Wenna Zhang [1], Kailiang Xue [1], Aiqin Yao [1] and Yunqiang Sun [1,2,*]

[1] School of Information and Communication Engineering, North University of China, Taiyuan 030051, China; b20210510@st.nuc.edu.cn (W.Z.); b20210508@st.nuc.edu.cn (K.X.); yaoaiqin@nuc.edu.cn (A.Y.)
[2] Key Laboratory of Instrumentation Science & Dynamic Measurement, Ministry of Education, North University of China, Taiyuan 030051, China
* Correspondence: syq@nuc.edu.cn

Abstract: Automatic modulation recognition (AMR) has wide applications in the fields of wireless communications, radar systems, and intelligent sensor networks. The existing deep learning-based modulation recognition models often focus on temporal features while overlooking the interrelations and spatio-temporal relationships among different types of signals. To overcome these limitations, a hybrid neural network based on a multimodal parallel structure, called the multimodal parallel hybrid neural network (MPHNN), is proposed to improve the recognition accuracy. The algorithm first preprocesses the data by parallelly processing the multimodal forms of the modulated signals before inputting them into the network. Subsequently, by combining Convolutional Neural Networks (CNN) and Bidirectional Gated Recurrent Unit (Bi-GRU) models, the CNN is used to extract spatial features of the received signals, while the Bi-GRU transmits previous state information of the time series to the current state to capture temporal features. Finally, the Convolutional Block Attention Module (CBAM) and Multi-Head Self-Attention (MHSA) are introduced as two attention mechanisms to handle the temporal and spatial correlations of the signals through an attention fusion mechanism, achieving the calibration of the signal feature maps. The effectiveness of this method is validated using various datasets, with the experimental results demonstrating that the proposed approach can fully utilize the information of multimodal signals. The experimental results show that the recognition accuracy of MPHNN on multiple datasets reaches 93.1%, and it has lower computational complexity and fewer parameters than other models.

Keywords: automatic modulation recognition; multimodal data; attention mechanism; hybrid neural network; parallel structure

Citation: Zhang, W.; Xue, K.; Yao, A.; Sun, Y. Automatic Modulation Recognition Based on Multimodal Information Processing: A New Approach and Application. *Electronics* **2024**, *13*, 4568. https://doi.org/10.3390/electronics13224568

Academic Editor: Enrique Romero-Cadaval

Received: 18 October 2024
Revised: 15 November 2024
Accepted: 19 November 2024
Published: 20 November 2024

1. Introduction

With the increasing complexity of wireless communication environments, the diversity of electromagnetic signal space is continuously increasing, the amount of information transmitted is growing, and the speed of signal change is also accelerating. Automatic modulation recognition (AMR) plays a crucial decision-making factor in the development of modern non-collaborative communication systems, and is widely used in the fields of wireless communications [1], radar systems [2], and intelligent sensor networks [3]. It provides the basis for the optimization and self-adaptability of various communication systems and plays an important role in radio spectrum management [4], signal monitoring [5], and communication security [6]. In practice applications, signals with different carrier frequencies, phases, and amplitudes may coexist. Therefore, accurately identifying the modulation type to which a received signal belongs is essential for efficient communication. AMR analyzes and interprets the characteristics of the received signal, employs new methods to utilize the information from multiple modalities, extracts useful features, and predicts the modulation types through classification, thereby determining the modulation mode adopted by the signal [7]. Modulation type refers

to the process of changing certain characteristics of the information carrier (e.g., radio waves) according to certain rules in order to transmit or convey information. By accurately identifying the modulation type, the receiver parameters can be dynamically adjusted according to the transmission characteristics of each signal type, enabling a more accurate estimate of critical information such as the carrier bandwidth of the signal, thereby optimizing the subsequent demodulation and decoding processes. This is helpful to improve the reliability of data transmission and spectrum utilization, and enables efficient communication in different environmental interference conditions.

The existing AMR algorithms can generally be classified into two categories: the Maximum Likelihood Method (MLM) based on the likelihood ratio test, and the feature-based method (FBM). The MLM [8] performs modulation recognition by calculating the likelihood ratio of the signal through statistical principles. It is based on modeling the signals of different modulation types and using statistical inference methods to estimate the probability density function of the signals, usually using Gaussian Mixture Models (GMMs) to describe the probability density distribution of the signals. By comparing the likelihood ratios of different modulation types one by one, the most probable modulation type can be determined. In practical applications, the MLM requires pre-establishing probability density function models for various modulation types and training the model parameters [9]. The method has high requirements for signal modeling, which needs to accurately describe the probability density distribution of the signal and has high computational complexity. However, in environments with high noise and low signal–noise ratio (SNR), the MLM outperforms the FBM. The FBM [10] primarily relies on the feature parameters of the signal for modulation recognition. It extracts and utilizes statistical characteristics [11], spectral information [12], time-domain features [13], spatial features [14], transient features [15], etc., from the original signals, and according to the relevance and importance of the features, it performs feature selection and dimensionality reduction operations to reduce feature dimensionality and improve the computational efficiency while retaining the information that is the most relevant, representative, and discriminative for the classification or recognition task; the selected features and corresponding labeled data are used to train the classification or recognition model. The FBM is widely used in various pattern recognition tasks, such as image recognition [16], speech recognition [17], text classification [18], object detection [19], etc. It allows for the selection of appropriate features according to the task requirements and provides intuitive interpretation and understanding. However, this method also faces the challenges of feature selection and extraction, as well as constraints in feature representation. Therefore, it is necessary to select appropriate features and algorithms according to the specific circumstances, and conduct sufficient experiments and optimization in practical applications. These two methods have their own advantages and disadvantages. The MLM requires the establishment of accurate mathematical models for signals, resulting in higher computational complexity. The FBM, on the other hand, relies more on feature extraction and classifier design, requiring appropriate feature selection and classifier training, but it offers faster computation speed. The recognition accuracy of traditional methods is low in complex channel environments or high noise interference; especially in the case of serious signal distortion, the traditional methods are often unable to fully extract effective features or deal with the nonlinear characteristics of the signal. Therefore, to overcome these challenges, deep learning (DL) methods have been widely applied to modulation recognition tasks in recent years. DL can automatically learn features and extract complex patterns in large-scale data, which improves robustness and recognition accuracy under harsh channel conditions.

In recent years, DL techniques have been widely applied in AMR tasks. DL techniques [20–22] are able to automatically learn and extract features from the original signal data by constructing a deep neural network model, thus realizing AMR. This method has better performance and robustness compared to traditional methods based on manual feature engineering. In 2016, O'Shea and his team [23] combined deep learning techniques with the recognition requirements for modulation patterns, and successfully identified

11 different analog and digital modulation modes through Convolutional Neural Networks (CNNs). Immediately after that, O'Shea et al. [24] improved the CNN and introduced an updated version of CNN2. At the same time, Wang et al. [25] proposed a feature selection method based on distance measurements by calculating the information entropy to extract features such as power spectrum, wavelet energy spectrum, singular spectrum, and Rayleigh spectrum of the signals, and utilized the neural network to complete the modulation mode recognition. Ettefagh et al. [26] designed a feature extraction and neural network adaptive system capable of recognizing various modulation modes such as AM, ASK, etc. In 2018, Zhang et al. [27] proposed a heterogeneous deep model fusion (HDMF) method that combines CNN and long short-term memory (LSTM) in two different ways to acquire a large database of single carrier-modulated signals and fading channels with 11 different noises at different signal noise ratios (SNRs). This demonstrates better performance than standalone networks. Rajendran et al. [28], on the other hand, improved the speed and accuracy of modulation pattern recognition by combining deep learning with the Fast Fourier Transform (FFT). Tayakout et al. [29] used Support Vector Machines (SVMs) and the Bayesian approach to improve the recognition accuracy through a simplified distributed spatio-temporal coding method. In 2020, Wang et al. [30] significantly improved recognition accuracy with a lightweight neural network of SK-ResNet18. They modified the traditional convolution to parallel grouped convolution and added a feed-forward self-attention mechanism. This algorithm achieved an overall recognition accuracy of 65.19% on the RML2016.10a dataset, indicating that their method has achieved some success in the field of modulation recognition and demonstrated high accuracy. In the same year, Liu et al. [31] used graph convolutional networks for modulation pattern recognition and demonstrated superior performance under low SNR conditions. In 2021, Jafar et al. [32] achieved 99% accuracy in recognizing QAM64 using the normality test and spectral in-phase and orthogonally modulated constellation diagram technique, achieving extremely high recognition accuracy under specific conditions. In the same year, Li et al. [33] solved the problem of feature correlation by combining one-dimensional CNN features with handcrafted features. Wang et al. [34], considering the conditions of time-varying signal–noise ratio, proposed a new generalized AMC method based on multitask learning (MTL), which can extract general features from datasets with different noise scenes and has higher robustness and generalization. Shi et al. [35] combined a multi-convolutional deep network overlay attention mechanism with a shallow network that mitigated misjudgments to achieve improved classification performance. Ansari et al. [36] achieved the classification of digital signals through various machine learning algorithms. The neural architecture search method of Zhang et al. [37] improved the flexibility of model search and overcame the difficulty of gradient propagation.

Despite the significant advancements made by many deep learning-based methods in the field of signal processing, most of these approaches focus solely on temporal features while neglecting the inherent correlations and spatio-temporal relationships between multimodal signals [38,39]. This results in poor adaptability when dealing with scenarios involving different modulation types and complex signal characteristics. For instance, traditional methods often struggle to effectively handle the feature differences in signals such as frequency modulation (FM) and phase modulation (PM). Additionally, some methods require substantial computational resources and long training times, limiting their applicability in real-time systems. To address these issues, this paper proposes a multimodal parallel hybrid neural network (MPHNN) based on a spatio-temporal attention mechanism. The proposed method is capable of processing information from multiple signal modalities simultaneously, fully exploiting their correlations. By using parallel convolutional layers to extract local features from each input modality and a feature fusion layer to combine the features from different modalities, the MPHNN significantly improves the accuracy and efficiency of modulation type classification.

In order to make full use of the spatio-temporal information, a spatio-temporal attention mechanism layer is employed to weigh the features at different time steps. Finally, the

global feature representation is extracted by a global feature extraction layer and a classifier layer is used for the classification prediction of modulation types. The network is used to recognize signals of a single modulation mode, taking original I/Q (in-phase/quadrature) time-domain data along with their instantaneous amplitude, phase, and frequency as input, and splitting and shaping the input data to generate a 128×2 vector, where 128 represents the length of each input signal. The output of the network is a one-hot encoded category corresponding to the 11 modulation modes of the input signals. The main contributions of this paper are as follows:

(1) Multimodal Information Extraction: This paper calculates the information of multiple modes using the received modulated signals. These modes include the original IQ signal, instantaneous amplitude, and phase, as well as frequency. Among them, the first two modes exhibit the temporal features of the modulated signal, while the latter mode provides the frequency-domain features of the modulated signal.

(2) Data Preprocessing and Network Design with Multi-Stream structure: In this paper, the input data are efficiently preprocessed by splitting it into vectors of size 128×2 to adapt to the network structure. Meanwhile, after extracting the features of multimodal information, a feature fusion strategy is utilized to learn the joint feature representation, and the spatio-temporal features are extracted by CNN and Bidirectional Gated Cycling Unit (Bi-GRU); the parallel architecture is designed to reduce the computational complexity so as to meet the needs of real-time processing. A suitable network architecture and output layer are designed to achieve the effective classification and recognition of modulation modes.

(3) Application of Attention Mechanisms: Two different attention mechanisms are applied in this paper. These attention mechanisms are able to assign weights to features in the channel and spatial dimensions, capturing long-range dependencies between features. Additionally, the attention mechanism highlights significant parts of signal variations to optimize network performance.

Through the aforementioned contributions, this paper provides an effective method and network architecture for the recognition of single modulation mode signals, and proposes a comprehensive approach that integrates the utilization of multimodal information, multi-stream structure, and attention mechanisms [40] to better address the problem of adaptive modulation recognition.

2. Multimodal Data

For calculating multimodal information, different types of input vectors can be obtained by considering both the time-domain and the frequency-domain of the received signal and by combining the features from time-domain and frequency-domain. Each type of vector contains a numerical representation of the corresponding feature, which is used to describe the instantaneous features of the received signal.

2.1. Mathematical Model of the Signal

In general, the channel of a wireless communication signal can be modeled as a time-varying linear filter. In this model, the input signal undergoes effects such as multipath propagation, noise interference, and fading after passing through the wireless channel, leading to variations in the signal such as delay, frequency offset, and amplitude attenuation.

After a certain period of transmission, it can fall entirely within the bandwidth of the receiver, then the mathematical model corresponding to the receiving signal $R(t)$ can be expressed as follows:

$$R(t) = S(t) * H(t) + N(t), \tag{1}$$

where $S(t)$ represents the input signal transmitted into the channel.

$H(t)$ represents the channel's response function to the input signal, which is typically time-dependent. The multipath effects and time-variation in the wireless channel can cause the channel's impulse response to dynamically change. The shape of the impulse response depends on the characteristics of the channel, such as multipath propagation, fading patterns, and others. Multipath propagation refers to the phenomenon where the

transmitted signal encounters various obstacles or objects during propagation, arriving at the receiver via different paths. These paths introduce different delays, frequency shifts, and amplitude attenuations, causing the signal to overlap at the receiver, resulting in multipath interference.

During transmission, N(t) is typically additive and independent, with the most common noise model being Additive White Gaussian Noise (AWGN). The strength of the noise is commonly measured by the signal–noise ratio (SNR). Noise can originate from various sources, including environmental interference, electromagnetic waves, and equipment thermal noise.

2.2. Instantaneous Features of the Signal

A complex baseband signal refers to a baseband signal with complex values, and its mathematical model can be expressed as follows:

$$S(t) = I(t) + jQ(t) = A(t)e^{j\theta(t)}, \tag{2}$$

where $I(t)$ and $\mathrm{Q}(t)$ represent the two independent components of the signal on the orthogonal channels, corresponding to the real and imaginary parts of the baseband signal, respectively. Therefore, the instantaneous features of the signal correspond to the following:

Instantaneous amplitude is obtained by performing analytic processing on the signal to extract the amplitude information of the signal at different time points.

$$A(t) = \sqrt{I^2(t) + Q^2(t)}, \tag{3}$$

Instantaneous phase indicates the phase delay or offset of the signal at different times.

$$\theta(t) = \begin{cases} arctan[\frac{Q(t)}{I(t)}] & I(t) > 0 \\ arctan[\frac{Q(t)}{I(t)}] - \pi & I(t) < 0, Q(t) \leq 0 \\ arctan[\frac{Q(t)}{I(t)}] + \pi & I(t) < 0, Q(t) > 0 \\ -\frac{\pi}{2} & I(t) = 0, Q(t) \leq 0 \\ \frac{\pi}{2} & I(t) = 0, Q(t) > 0 \end{cases}, \tag{4}$$

Instantaneous frequency refers to the frequency value at a particular moment in the time series. The phase change $\Delta\varphi(\mathrm{t})$ between adjacent time points can be calculated by differentiating the instantaneous phase.

$$\Delta\varphi(t0) = \varphi(t + t0) - \varphi(t), \tag{5}$$

$$f(t0) = \frac{\Delta\varphi(t0)}{2\pi\Delta t}, \tag{6}$$

Among them, $\Delta\varphi(\mathrm{t0})$ represents the phase change at time t0, and Δt represents the time difference between adjacent time points. In order to better visualize the features of these vectors, the instantaneous amplitude, instantaneous phase, instantaneous frequency, and IQ time-domain plots from the publicly available dataset RML2016.10A [16] are used for visualization. In Figure 1, each row of subplots represents the features corresponding to a modulation mode: from left to right, instantaneous amplitude, instantaneous phase, instantaneous frequency, and IQ time-domain plots.

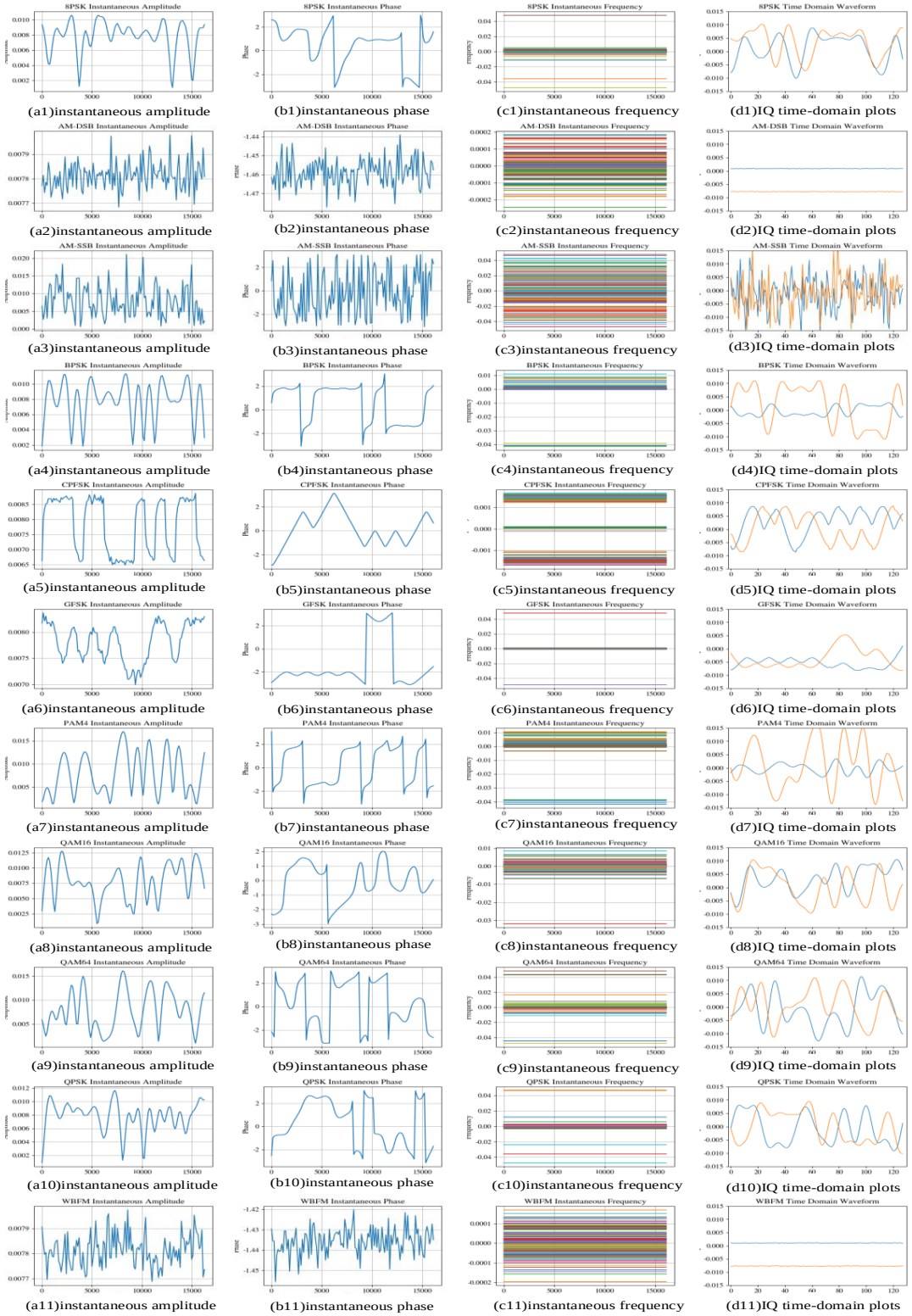


Figure 1. Visualization of instantaneous amplitude, instantaneous phase, instantaneous frequency, and IQ time-domain plots for 11 modulation modes.

3. Proposed Model

The multimodal parallel hybrid neural network (MPHNN) model proposed in this paper is specifically designed for AMR. A multi-input multi-output neural network model is constructed using TensorFlow and Keras. It mainly consists of an input module, a feature

extraction module, attention mechanisms, a Bi-GRU, and a global average pooling layer. The input module accepts multiple types of input data, including original data signals and amplitude, frequency, and phase features, which are preprocessed and fed into the model for further processing. The feature extraction module partially uses four convolutional layers to extract features from each input, obtaining convolutional results for different types of features. These convolutional layers help in extracting the localized features in the input data, thus better capturing the information within the input data. The attention mechanism part includes the Convolutional Block Attention Module (CBAM) and Multi-Head Self-Attention (MHSA), which uses a network of attention fusion mechanisms to concatenate the feature tensors obtained from the attention mechanisms with the convolutional results. This enhances the model's focus on key information. The feature tensor after attention fusion is then input into the Bi-GRU, which helps to preserve the contextual information in the sequence data and extracts richer feature representations to better capture the dependencies within the sequence. The global average pooling layer converts the sequence features into fixed-length feature vectors. The output layer maps the pooled features to the output categories of the model through a fully connected layer, which is used to predict the output results of the model. The overall architecture of the MPHNN is shown in Figure 2.

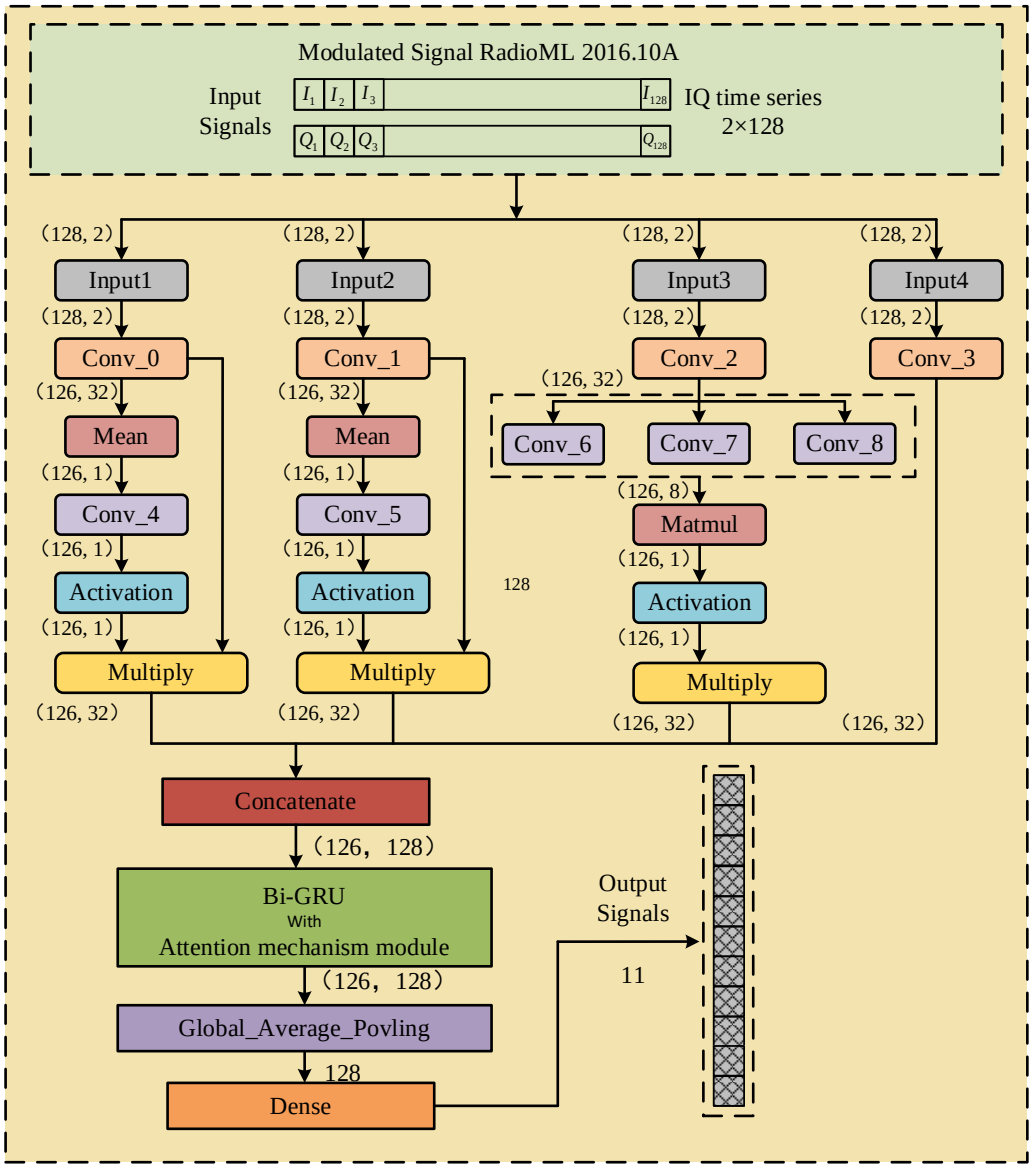


Figure 2. Overall architecture of the MPHNN.

The MPHNN model, through the aforementioned design, can effectively utilize the attention mechanism, convolutional feature extraction, Bi-GRU, and other techniques to achieve efficient processing and accurate predictions of AMR tasks.

3.1. Multi-Stream Structure

The multi-stream input and processing structure of the preprocessing section in Figure 2 captures the features at different scales and efficiently extends the dataset by utilizing complementary information such as original data, amplitude, frequency, and phase. Subsequently, each input stream is processed through a different convolutional block to extract dual-stream and single-stream features, respectively. With this multi-stream architecture, rich features in the input data can be better extracted. By using a concatenate function to splice various feature tensors along the channel dimension, the dimensions of the four input tensors are (126, 32), and the output dimension after concatenation is (126, 128). This concatenated output is then passed to the next convolutional block, preserving the feature information of each branch to collect multimodal information from multiple streams.

3.2. Attention Mechanism Module

The introduction of attention mechanisms enables the model to handle sequential data more flexibly and to focus on important information at different locations adaptively. This is particularly beneficial for dealing with long sequential data, and can improve the balance between long-term dependencies and local information, thus improving the model's performance. This paper uses two types of attention mechanisms to enable the model to pay more attention to important inputs.

3.2.1. CBAM

As shown in Figure 3, the CBAM is structured as C × H × W. Based on the given feature map, it shows that it can generate attention feature map information in both channel and spatial dimensions, and multiply it with the original input feature map to achieve adaptive feature correction and generate the final feature map information. It is able to suppress unnecessary regional responses by focusing on important features of the image. The CBAM is a lightweight module that can be embedded into any backbone network to improve its performance. The main purpose of the attention mechanism is to improve network performance through precise attention mechanisms and the suppression of irrelevant noise information.

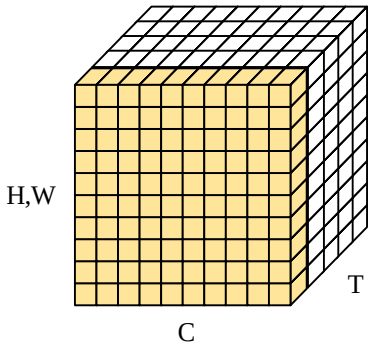


Figure 3. Structure of CBAM.

The working mechanism of CBAM is shown in Figure 4, which consists of two parts: channel attention mechanism and space attention mechanism.

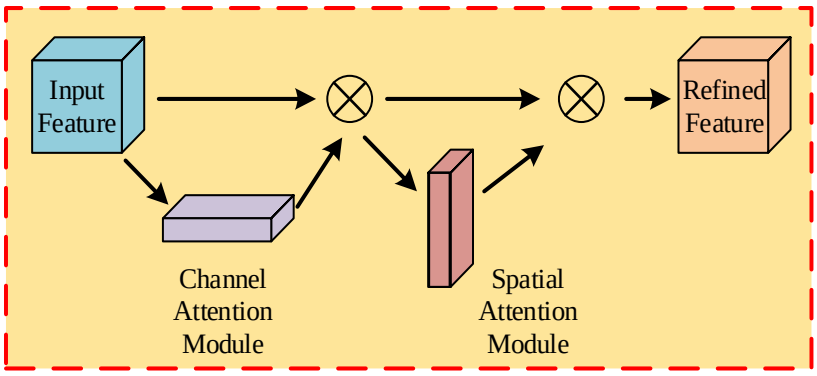


Figure 4. Working mechanism of the CBAM.

Firstly, the channel attention mechanism focuses on the channel features through capturing the relationships among the features within the same channel. By applying the average pooling $F_{avg}^{c} \in R^{C\times1\times1}$ to compress and reduce the spatial dimensions on the feature map $F \in R^{C\times H\times W}$ generated by the backbone network, the degree information of the target object can be learned and the holistic information within each channel can be captured. These global feature descriptions are fed into a shared Multi-Layer Perceptron (MLP) network consisting of multiple fully connected layers and activation functions to generate the final 1D channel attention feature map $M_c \in R^{C\times1\times1}$. Each channel in the channel attention feature map corresponds to each channel in the input feature map, which can be regarded as a feature detector, with the value of each channel indicating the weight of the importance of that channel. In summary, the channel attention calculation can be described by the following equation:

$$M_c(F) = sigmoid\Big(MLP(AvgPool(F))\Big) = sigmoid\Big(W_1\Big(W_0F_{avg}^{c}\Big)\Big), \tag{7}$$

where W_1 and W_0 represent the weights of MLP and average pooling, respectively, and $F_{avg}^{c} \in R^{C\times1\times1}$ represents the feature map obtained after performing spatial average pooling, with dimensions of $C \times 1 \times 1$.

Secondly, the spatial attention mechanism focuses on the effective information within the spatial dimension of the feature map. Unlike the channel attention mechanism, the spatial attention mechanism identifies where the valid information is located on the feature map. By applying average pooling along the channel dimension, the input feature map is downscaled to capture global feature representations across channels, obtaining $F_{avg}^{s} \in R^{1\times H\times W}$.

This global feature representation is then fed into a convolutional operation to generate a 2D spatial attention feature map $M_s \in R^{1\times H\times W}$. The 2D feature map undergoes appropriate processing, such as applying activation functions, to obtain the final spatial attention feature map. Each element in the spatial attention feature map corresponds to a position on the input feature map, with its value representing the weight, indicating the importance of that position. Through the spatial attention mechanism, the network can focus on the positions of effective information on the feature map, thus improving the network's sensory field and attention to spatial locations. It can be summarized as follows:

$$M_s(F') = sigmoid\Big(F^{7*7}(AvgPool(F'))\Big) = sigmoid\Big(F^{7*7}\Big(F_{avg}^{s}\Big)\Big), \tag{8}$$

The whole process can, thus, be described in the following equation:

$$F'' = M_s(F') \otimes M_c(F) \otimes F, \tag{9}$$

where $\otimes$ denotes element-by-element multiplication with broadcasting employed to adapt the dimensions and match the shapes of the inputs.

3.2.2. MHSA

The multi-head attention mechanism is an extension of the self-attention mechanism that incorporates multiple attention heads. By performing linear transformations on the input, multiple sets of different query, key, and value mappings are obtained; multiple attention representations are computed in parallel; and then the final multi-head attention representation is obtained through linear combination or concatenation, as shown in Figure 5. Each attention head can focus on different aspects of the input sequence, realizing attention in different subspaces and allowing the model to capture richer information. This design enhances the expressive ability of the model, enabling it to handle more complex tasks and improve generalization, making it more adaptive when dealing with different data and contexts.

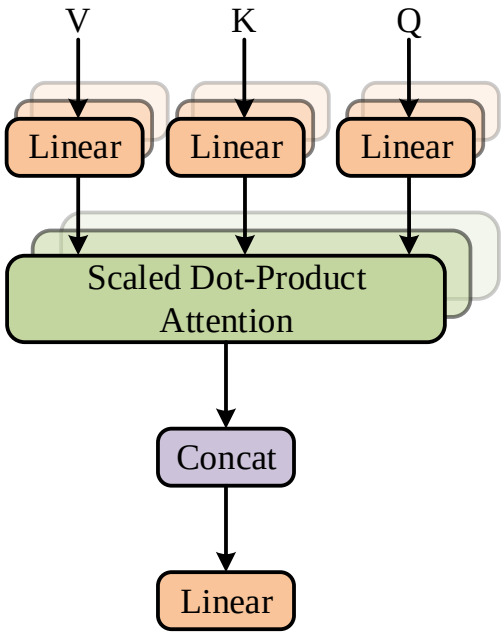


Figure 5. Structure of the Multi-Head Self-Attention (MHSA) module.

With the attention model, given an element in the target as a query, the attention mechanism calculates the similarity or correlation between the query and each key, obtaining the weight of the corresponding value for each key. Specifically, in multi-head attention, the query, key, and value are first linearly mapped h times (where h represents the number of heads), each mapping using different parameter matrices W^Q, W^K and W^V. Then, each head performs a scaled dot-product attention calculation, yielding h attention outcomes.

$$h_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right), \tag{10}$$

Finally, the h attention outcomes are concatenated together, and pass through an additional linear transformation to obtain the final multi-head attention result.

$$MH(Q, K, V) = Concat(h_1, h_2, h_3, \ldots, h_h)W^O, \tag{11}$$

The design of this multi-head attention structure allows the model to simultaneously utilize different representation subspaces to learn relevant information, thus enriching the representation capability of the model. Among them, the calculation of scaled dot-product attention is illustrated in Figure 6.

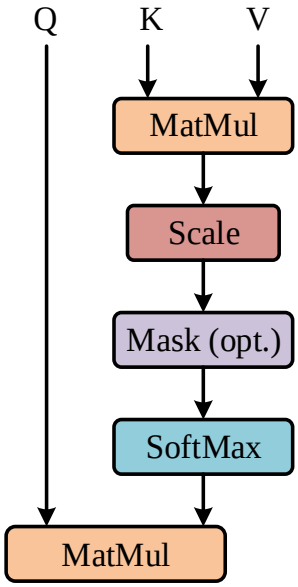


Figure 6. Scaled dot-product attention.

As shown in Equation (11), first, the "Query (Q), Kkey (K), and Value (V)" projections are constructed for the constituent elements in the source. Second, the score of the query vector dot-product with the transpose of the key vector, i.e., QK^T, is computed, which determines the degree of attention that the current element pays to the other elements in the input sequence. The score is divided by the scaling factor $\sqrt{d_k}$, which plays a regulating role and controls the size of the dot-product so that it is not too large or too small. If the scaling factor is small, then the result of the dot-product will also be small, in which case, the difference between the dot-product attention and additive attention is not very significant. However, if the scaling factor is large, the result of the dot-product will be large. If the scaling is not carried out, the problem of becoming almost 0 or 1 after softmax may easily occur, resulting in unreasonable attention distribution. In addition, excessive dot-product results also lead to the problem of gradient disappearance during backpropagation. Therefore, it is usually necessary to scale in order to balance the gradient with the direction of propagation. Then, these weights undergo softmax normalization to become a probability distribution of the attention of an element on other elements. After multiplying each value vector by softmax, the desired value to be attended to remains unchanged while irrelevant elements are suppressed. Finally, the weighted sum of the weights and corresponding values is computed to produce the output of the attention model.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{12}$$

3.2.3. Fusion of Two Attention Mechanisms

Embedding the CBAM and MHSA as parallel attentional modules into the neural network is important. This approach allows the neural network model to concurrently focus on both inter-channel correlations and intro-sequence importance during the feature extraction process. By retaining key information while reducing noise, the model's representation and generalization capabilities for the input data are enhanced. In this study, a specialized attention fusion mechanism is designed, as shown in Figure 7, for dynamically determining the weights of the outputs from the CBAM and MHSA modules. The outputs from the CBAM and MHSA modules are connected to the network of the attention mechanism. Through training, the model dynamically learns the weights of each output, using these attention weights to combine the outputs of the two modules. This mechanism allows the model to adaptively adjust the contribution of the outputs of the

two modules in different situations. Finally, the outputs of the two modules are connected to act on the signal output features of the Bi-GRU module, which are then fed inside the global average pooling layer for further processing to integrate the information from the two attention mechanisms.

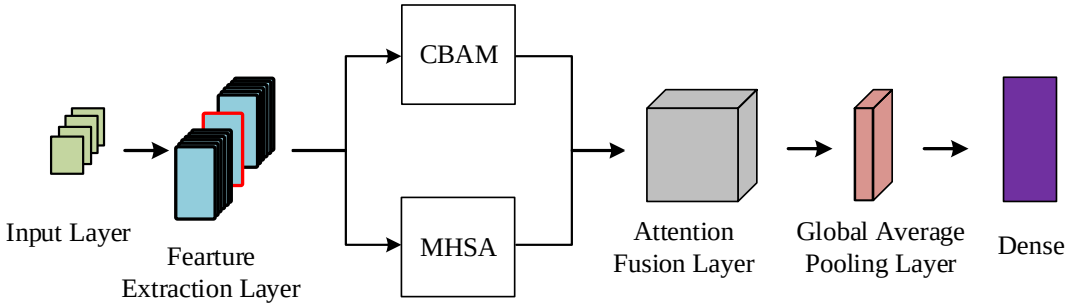


Figure 7. Structure of attention fusion mechanism.

3.3. Bi-GRU

Bi-GRU (Bidirectional Gated Cycling Unit) is a variant of recurrent neural network (RNN), as shown in Figure 8, which combines the flow of information in both the forward and backward directions.

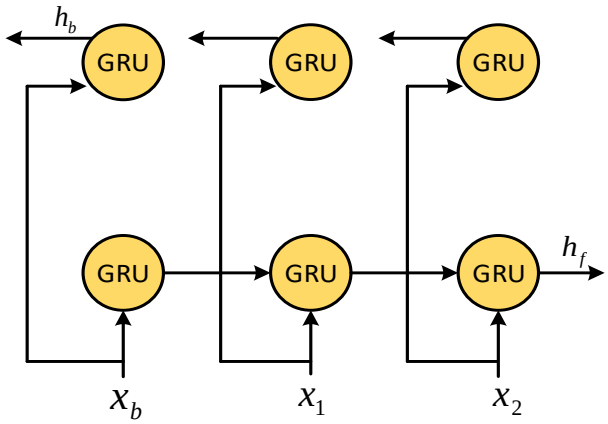


Figure 8. Bi-GRU information flow transfer diagram.

A GRU is a kind of gated recurrent unit, similar to the long short-term memory network (LSTM), used for processing sequential data. The GRU has update gates and reset gates, which control whether to update and reset the current state. Compared to LSTM, GRU reduces the number of gates, making the calculation simpler. The Bi-GRU consists of two GRUs, a forward GRU and a reverse GRU. The forward GRU processes the input sequence in the forward direction, while the reverse GRU processes the input sequence in the reverse direction. In this way, the Bi-GRU can utilize contextual information from both the past and the future, capturing more comprehensive sequence features.

$$\overrightarrow{h_{tf}} = GRU\left(\overrightarrow{h_{t-1}}, x_t\right), \tag{13}$$

$$\overrightarrow{h_{tb}} = GRU\left(x_t, \overrightarrow{h_{t+1}}\right), \tag{14}$$

where $\overrightarrow{h_{tf}}$ represents the hidden state from left to right, which corresponds to the forward computation. $\overrightarrow{h_{tb}}$ indicates the hidden state from right to left, which corresponds to the reverse computation. x_t represents the t-th element of the input sequence. By concatenating the hidden states from both the forward and reverse directions, the final hidden state h_t is obtained.

$$h_t = \left[\overrightarrow{h_{tf}}\,;\,\overrightarrow{h_{tb}}\right], \tag{15}$$

Passing the hidden state h_t to the fully connected layer provides the output y_t:

$$y_t = softmax(Wh_t + b), \tag{16}$$

where W and b are the weight and bias of the fully connected layer, respectively.

4. Experimental Setup and Result Analysis

4.1. Experimental Dataset and Implementation Details

The standard dataset RadioML2016.10A, generated by GNU Radio, is used in the experiment as a verification platform for AMR tasks, mainly based on its advantages of diversity, signal–noise ratio coverage, data quality, and wide application. The dataset contains a variety of modulation modes, which can effectively evaluate the generality and robustness of the method under different modulation schemes. It covers low to high SNR conditions, which provides a challenge for performance evaluation in low SNR environments, and the data collection is standardized and of high quality, which ensures the accuracy of the experimental results. In addition, RadioML2016.10A is widely used in academic and industrial fields, which can provide a unified platform for cross-domain comparison and benchmarking, making the advantages of the method easier to demonstrate and closely related to actual communication requirements, which has the following characteristics:

(1) It includes multiple signal–noise ratio (SNR) levels ranging from −20 dB to 18 dB with an interval of 2 dB;
(2) It contains severe channel fading accompanied by intermediate frequency offset, time offset, sampling rate offset, multipath, and Additive White Gaussian Noise (AWGN) effects;
(3) The dataset consists of single I/Q complex input samples with a dimension of [2 × 128];
(4) It includes simplified samples of 220,000 signals and eleven modulation modes used in practical applications, including BPSK, QPSK, 8 PSK, 16-QAM, 64-QAM, CPFSK, GFSK, 4-PAM, WBFM, AM-SSB, and AM-DSB. These modulation schemes are common modulation types in wireless communications and are widely used in different communication systems. Their selection not only reflects the diversity in practical communication, but also covers different modulation techniques and implementation ways, which is highly representative. In the task of automatic modulation recognition, different modulation modes have different signal characteristics. Testing the performance of a method on multiple modulation modes can effectively evaluate its generality and robustness, especially in complex environments. The dataset is divided into three subsets for training, validation, and testing in an appropriate ratio of 6:2:2. A summary of the dataset is shown in Table 1.

In the experimental setup, all the networks were trained for a total of 200 epochs, which is a common practice in deep learning experiments. However, considering the potential impact of the number of epochs chosen and the early stopping criterion on model performance and convergence, an early stopping strategy based on validation loss was adopted. Specifically, if the validation loss is not reduced for five consecutive epochs, the training would be terminated, thus reducing the risk of overfitting. Furthermore, to optimize the training efficiency and ensure reproducibility, the batch size was kept consistent at 1024 and the learning rate of all the models was initialized to 0.001. The Adam optimizer, which has better robustness and effectiveness, was used as the network optimizer, and the categorical cross-entropy was used as the loss function during training. Regarding the activation function in the neural network structure, the ReLU (Rectified Linear Unit) activation function was used for all the layers except the output layer, while the softmax activation function was used for the multi-class classification task. For experimentation, the Keras platform with the TensorFlow backend was utilized, leveraging computing acceleration provided by two NVIDIA CUDA-supported GeForce GTX 2070 GPUs(manufactured by NVIDIA Corporation, which is headquartered in Santa Clara, CA, USA). These resources ensured the effective training and testing of the models under consistent conditions.

Table 1. Analysis of the dataset.

Dataset	RadioML2016.10A
Number of Modulation Modes	11
Categorization	8 Digital Modulations; 3 Analog Modulations
Digital Modulations	QPSK, PAM 4, 8PSK, BPSK, CPFSK, BFSK QAM64, and QAM16
Analog Modulations	AM-DSB, AM-SSB, and WB-FM
Sample Size	220,000
Sampling Frequency	1 M/s
Sampling Interval	128 μs
Sample format	IQ format
Sample Input Size	128×2
Samples/Symbols	8
Signal–Noise Ratio (dB)	−20:2:18
Training Data	132,000 Samples (60%)
Validation Data	44,000 Samples (20%)
Test Data	44,000 Samples (20%)
Channel Environment	Carrier Frequency Offset, Symbol speed offset, Delay, Thermal Noise, etc.

Learning rate, batch size, and epochs are the core hyperparameters in deep learning training, which directly affect the stability, convergence speed, and performance of the model. The learning rate determines the parameter update step size: too large will cause oscillation, and too small will converge too slowly. The experiments show that choosing a 0.001 learning rate can achieve a better balance between stability and convergence speed. The batch size affects the gradient update frequency: a large batch size improves the computational efficiency but may reduce the generalization ability, and a small batch size helps to break through the local optimum, but the training efficiency is low. We chose 1024 as the batch size to balance stability and computational efficiency. The number of epochs affects the degree of model learning: too few may lead to underfitting, and too many may lead to overfitting. With cross-validation and early stopping techniques, we selected 200 epochs, ensuring efficient learning and avoiding overfitting.

4.2. Performance Evaluation

The training loss reflects the fitting degree of the model to the training data, while the validation loss reflects the fitting degree of the model to new data. If the validation loss is much greater than the training loss, there may be an overfitting problem. When both the training and validation losses decrease, it indicates that the model has good fitting effects. In Figure 9, we conduct a comprehensive performance evaluation of the proposed model. By recording the loss and accuracy values for both the training set and the test set, we obtain the epoch–loss and epoch–accuracy curves as the number of training iterations increases. The results show that in the proposed method, there is no significant difference between the training loss and validation loss with the increase in training iterations, and both of them decrease with the increase in epochs. Meanwhile, the training accuracy and verification accuracy are gradually improved. In addition, the accuracy on the training set is slightly higher than that on the verification set, indicating that there has been no overfitting phenomenon during the training process. Overall, the proposed model performs stably in terms of training and validation loss, which indicates that the algorithm has good generalization ability and performance. Additionally, the training in this paper targets models under datasets with multiple spans of different SNRs, and in order to accurately evaluate the overall generalization ability of the model, the iterative training accuracy is calculated as the average accuracy of the datasets under each SNR condition. This approach facilitates a more comprehensive assessment of the overall performance of the model under

various SNR conditions and provides more insight into the robustness of the model in the face of different challenges.

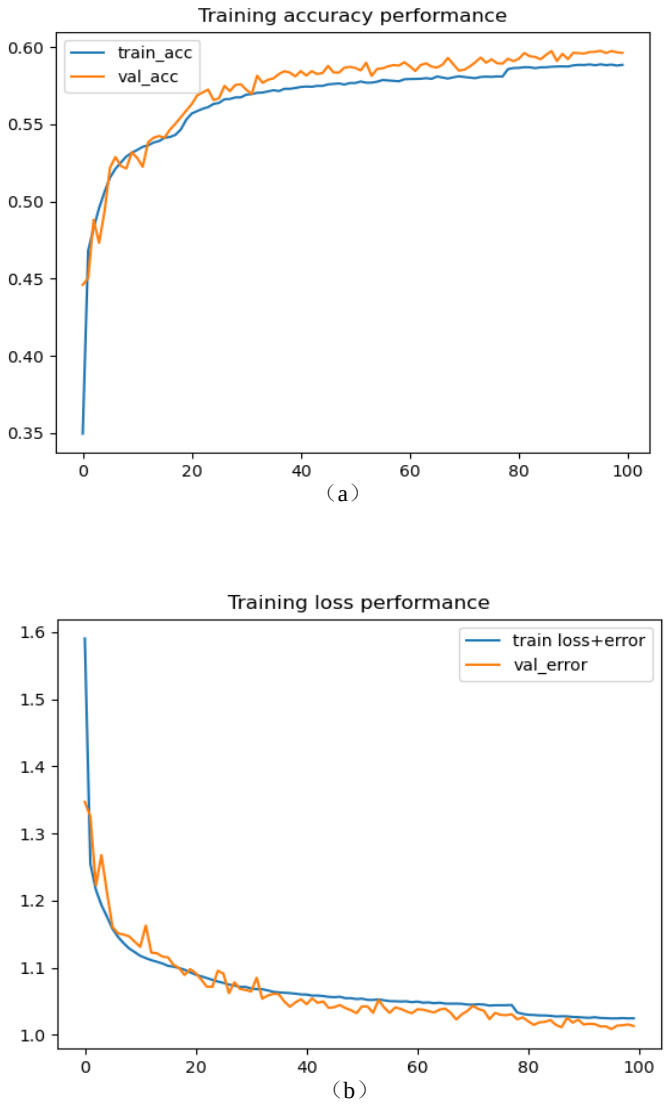


Figure 9. Changes during training: (**a**) accuracy and (**b**) loss values.

The specific metric of recognition accuracy refers to the proportion of correctly identified classifications by the model out of the total instances in the test dataset. It evaluates the signals within the dataset, providing an overall indication of the model's correctness that surpasses the training and validation losses. This metric allows for a more thorough assessment of the model's effectiveness in real-world scenarios.

4.3. Performance Comparison with Existing Work

In the SNR range of −20 dB to 18 dB, the proposed MPHNN was compared with six existing baseline models in terms of recognition accuracy and computational complexity, as shown in Figure 10. The models are named as follows:

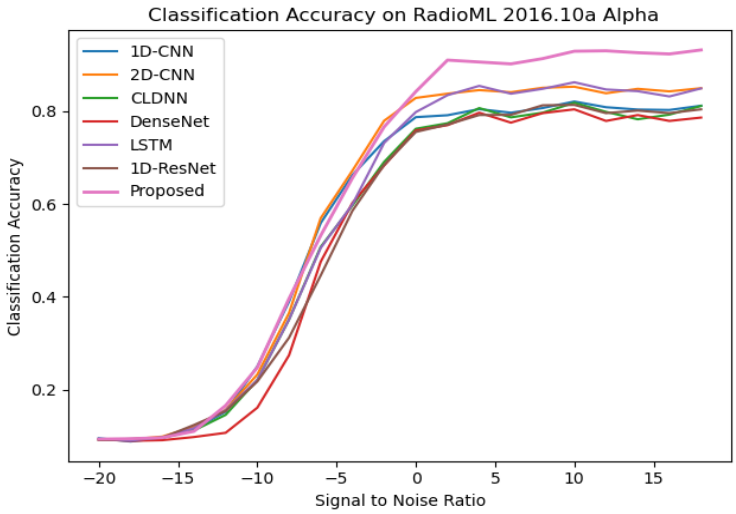


Figure 10. Recognition accuracy of the dataset RadioML2016.10A on several models.

(a) 1D-CNN [41]: The model employs a Convolutional Neural Network (CNN) architecture consisting of convolutional layers, max-pooling layers, and dense layers. Hyperparameters include filters set to 64, a kernel size of 3, and a ReLU activation function. The model utilizes a one-dimensional convolution operation to capture local patterns and features in temporal data.

(b) 2D-CNN [42]: The model consists of 2D convolutional layers, 1×2 max-pooling layers, and dense layers. Hyperparameters include 256, 128, or 64 filters; a kernel size of 2×8; a ReLU activation function; and the use of Glorot uniform distribution for initializing the convolutional kernel weights. Image classification is trained on the RadioML dataset. When converting the signal sequence into two-dimensional images, different representations can be considered, such as using spectrograms or conventional grayscale image representation, which is very useful for processing spatially structured data by using two-dimensional convolution operations on images to effectively extract high-level features and patterns in images.

(c) CLDNN [43]: This model contains three key components: CNN, LSTM, and DNN. The specific architecture includes an input layer, a zero-padding layer, three convolutional layers, a dropout layer, a concatenation layer, a reshape layer, an LSTM layer, two fully connected layers, and a softmax layer. The hyperparameters include 50 convolutional kernels, a kernel size of 1×8, a dropout rate of 0.5, 50 LSTM units, and 256 and 11 neurons in the two fully connected layers, respectively. Convolutional operations are used to extract local features from input features to capture the local relationships and spectral features of the input data. LSTM is employed to model the temporal information of the features, capturing contextual dependencies and helping the model understand the temporal information in speech signals. DNN connects the convolutional layers, and recurrent layers and other auxiliary layers to combine features between different layers and perform the final classification or regression task.

(d) DenseNet [44]: This model incorporates multiple convolutional layers, a dropout layer, fully connected layers, and a softmax layer. The hyperparameters include a dropout rate of 0.6, an input data shape of (2, 128), and an output class count of 11. The model introduces the concept of dense connections. By connecting all the feature maps from the previous layer to the current layer, the progressive construction allows the network to build high-level features from low-level features, enabling direct information flow between different layers.

(e) LSTM [45]: The model is a recurrent neural network consisting of two LSTM layers and one fully connected layer. The hyperparameters include an input data shape of (128, 2) and an output class count of 11. The model introduces gating mechanisms, and can effectively deal with long-term dependencies and the gradient vanishing problem in sequences, allowing the model to better capture and memorize important information in the sequence.

(f) ResNet [46]: The model comprises convolutional layers, residual connections, dropout layers, and fully connected layers. The hyperparameters include an input data shape of (2, 128) and an output class count of 11. The model addresses the issues of gradient vanishing and model degradation when training deep neural networks by introducing residual connections.

(g) Proposed model, which is referred to as the model developed in this study.

The comparison in Figure 10 evaluates the performance of the MPHNN against these baseline models in terms of recognition accuracy and computational complexity within the specified SNR range.

Figure 10 illustrates the recognition accuracy curves of the seven methods at all the SNRs under the same experimental conditions. As the SNR increases, the recognition accuracies of all the models are gradually improved. Upon comparison, it is found that the proposed model achieves the highest accuracy under most SNR conditions. Specifically, in low SNR scenarios, the recognition accuracy of the proposed model is similar to that of the other baseline networks. When the SNR is -8 dB, the inflection point of the recognition accuracy curve of the proposed model exhibits a noticeable improvement, and after an SNR of -2 dB, the proposed model begins to demonstrate its advantages. At an SNR of 2 dB, the recognition accuracy of the proposed model is still around 91%, while the accuracy of the other networks is below 90%. Under an SNR of 18 dB, the proposed model achieves a peak accuracy of 93.1%, which is 12% better than that of 1D-CNN, 8.1% better than that of 2D-CNN, 12% better than that of CLDNN, 14.5% better than that of DenseNet, 8.2% better than that of LSTM, and 12.8% better than that of ResNet. It can be observed that the overall recognition rate of the model with the addition of the attention mechanism has an advantage over the other baseline models due to the use of CNN and Bi-GRU to extract the spatial and temporal correlation features, performing weighted processing on the features with different temporal steps and spatial correlations of the signals.

The confusion matrix is an important metric for evaluating the classification performance of an algorithm. It shows the recognition accuracy of each modulation method in the form of a matrix. Take the confusion matrix at 18 dB SNR as an example to illustrate. The vertical axis represents the actual labels, while the horizontal axis represents the predicted labels. The values on the diagonal of the matrix represent the number of samples with accurate prediction as a percentage of the total number of samples. Figure 11 illustrates the confusion matrix for the six baseline models and the proposed model when the SNR is 18 dB. In the RadioML 2016.10A dataset, the distinction between AM-DSB and WBFM becomes difficult due to the small observation window and low information rate, resulting in the frequent occurrences of silence between data symbols. In this dataset, there is confusion between the higher-order QAM16 and QAM64 digital modulations as they share common features in the constellation diagrams, and this confusion is mainly affected by the shorter observation time. However, as shown in Figure 11g, the proposed model is able to identify the different modulations more accurately under the condition of higher SNRs, significantly reducing the confusion between QAM16 and QAM64, with recognition accuracies of 93% and 91%, respectively, at an SNR of 18 dB. This is due to the fact that the proposed model extracts spatial and temporal features that more accurately capture the periodic internal trends corresponding to the modulation types. By utilizing these features, different modulation types can be better distinguished, leading to improved recognition accuracy.

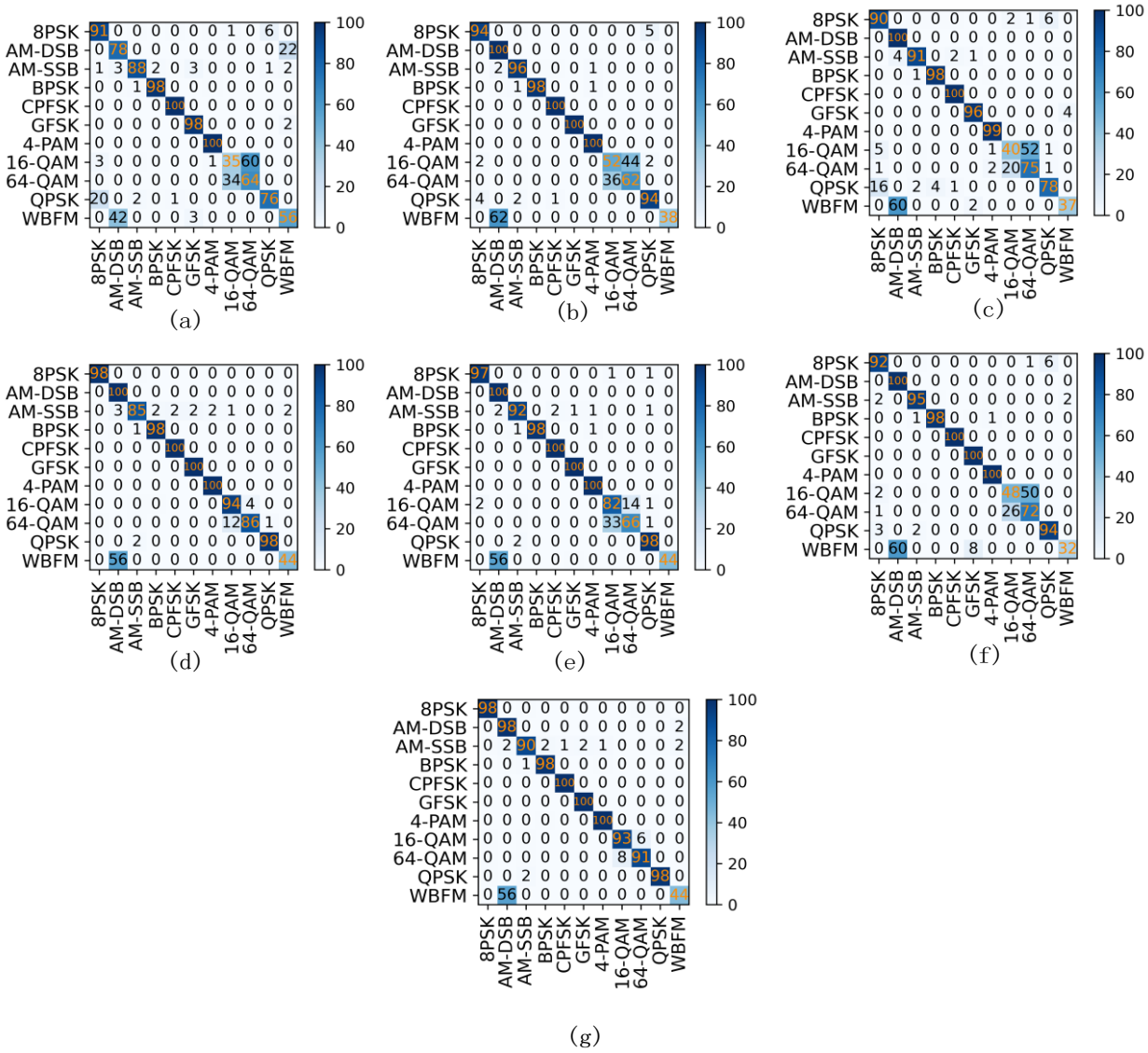


Figure 11. Confusion matrix at an SNR of 18 dB for (**a**) 1D-CNN, (**b**) 2D-CNN, (**c**) CLDNN, (**d**) DenseNet, (**e**) LSTM, (**f**) ResNet, and (**g**) proposed model.

Figure 12 provides a comprehensive analysis of the average recognition accuracies for the six baseline models and the proposed MPHNN at various signal–noise ratio (SNR) levels. The results highlight the significant performance improvements achieved by the MPHNN, especially in challenging low SNR environments. At lower SNR levels, where signal distortion and noise interference pose significant challenges, the MPHNN demonstrates outstanding capability in accurately recognizing modulation patterns compared to baseline models. The notable enhancement in accuracy at low SNR levels is particularly noteworthy as it addresses a common limitation in modulation recognition systems, where reliable classification becomes increasingly difficult in the presence of noise and interference. Furthermore, at an SNR of 18 dB, MPHNN achieves significantly higher classification accuracy compared to the baseline models. These findings emphasize the robustness and effectiveness of the proposed MPHNN architecture, which exploits spatio-temporal attention mechanisms and multimodal signal integration to enhance modulation recognition performance across different SNR levels. By outperforming the baseline models, especially at low SNR scenarios, the MPHNN demonstrates considerable potential for real-world applications, achieving higher recognition accuracy.

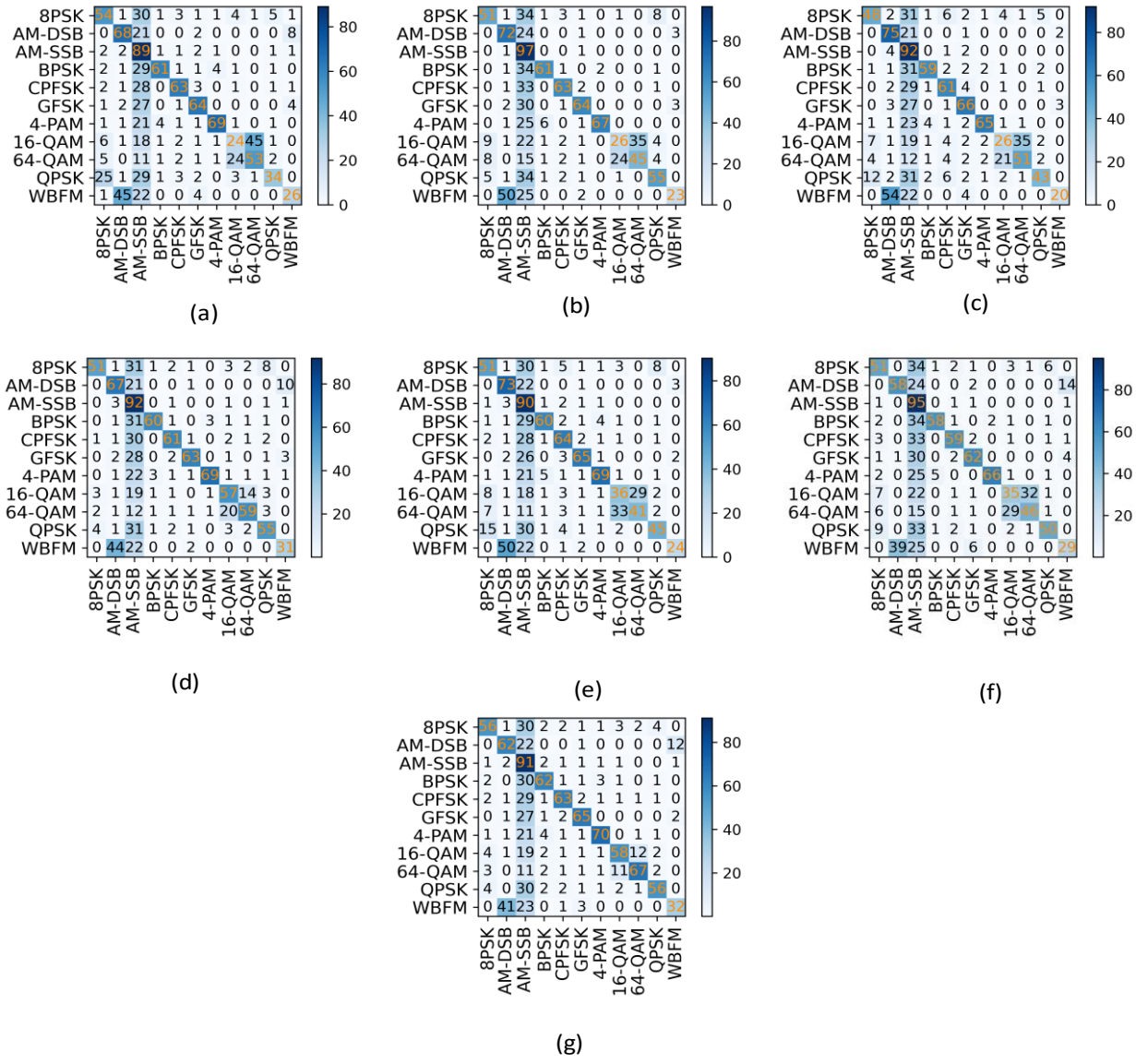
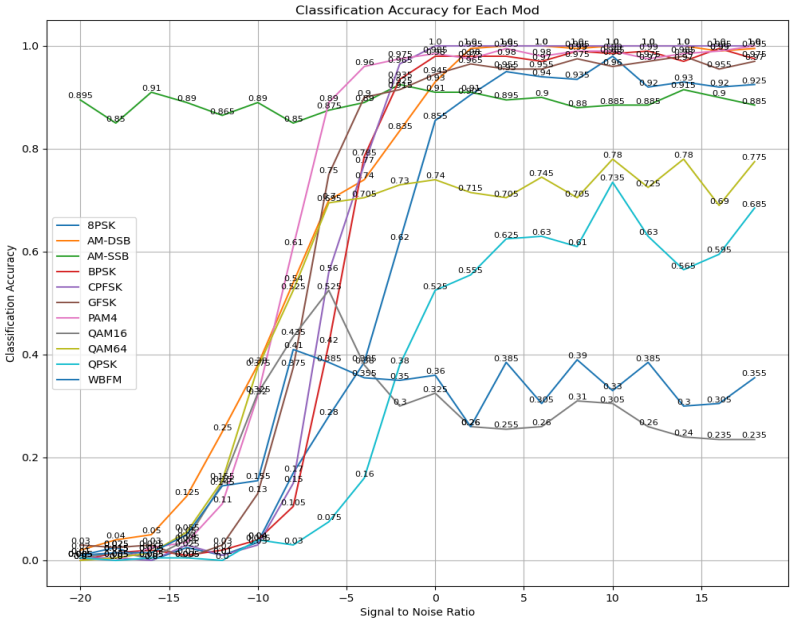
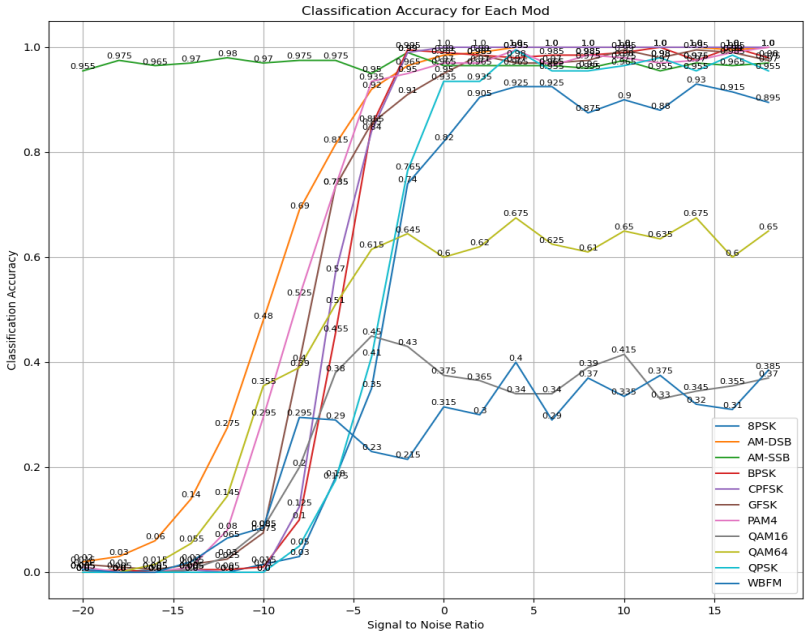


Figure 12. Confusion matrix at full SNR: (**a**) 1D-CNN, (**b**) 2D-CNN, (**c**) CLDNN, (**d**) DenseNet, (**e**) LSTM, (**f**) ResNet, and (**g**) proposed model.

Furthermore, these results are used to divide different SNR ranges, and then the recognition curves of various modulation signals under different recognition methods are drawn, as shown in Figure 13. Due to the lack of clear reflection in amplitude and phase for various analog modulation types, the recognition rate of WBFM is lower. This leads to the similarity in time-domain between AM-DSB and WBFM in Figure 1. However, as the SNR increases, the classification accuracy of various modulation methods also improves. Some WBFM samples may be misclassified as AM-DSB. This is because at a lower SNR, the noise occupied a large proportion of the signal, making the modulation signal more irregular. Therefore, lower SNR poses more challenges for signal recognition.

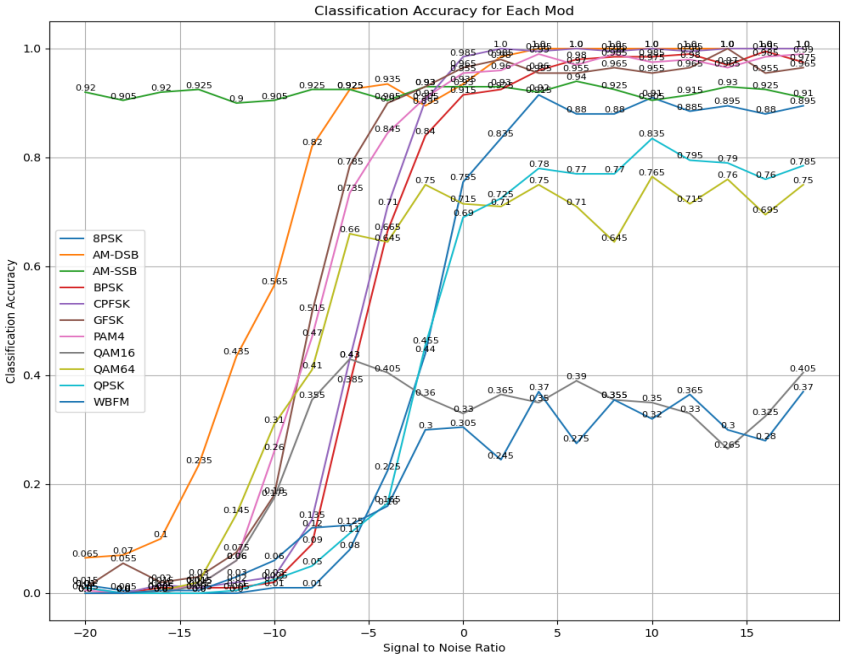


（a）

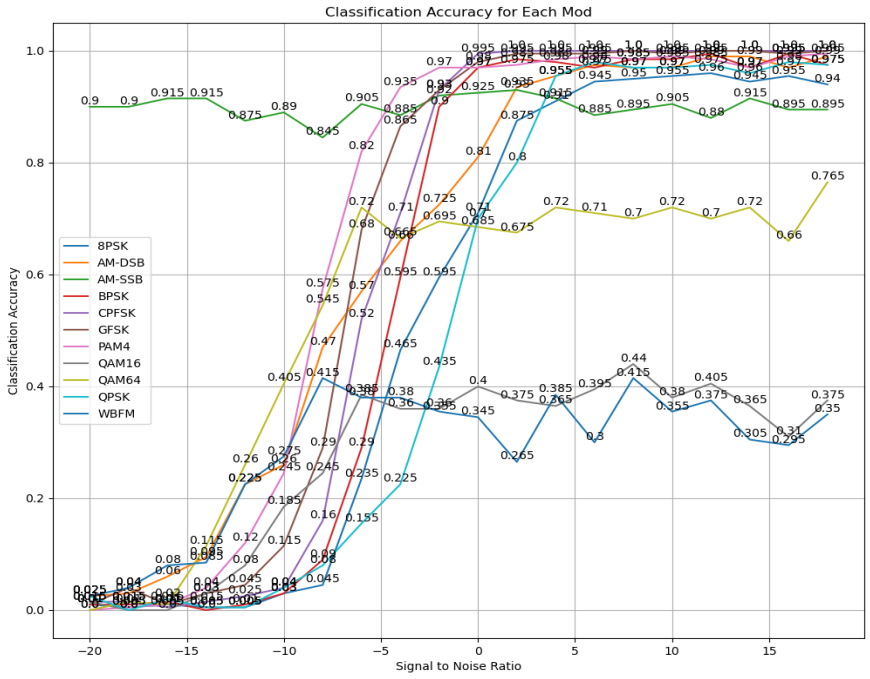


（b）

Figure 13. *Cont*.

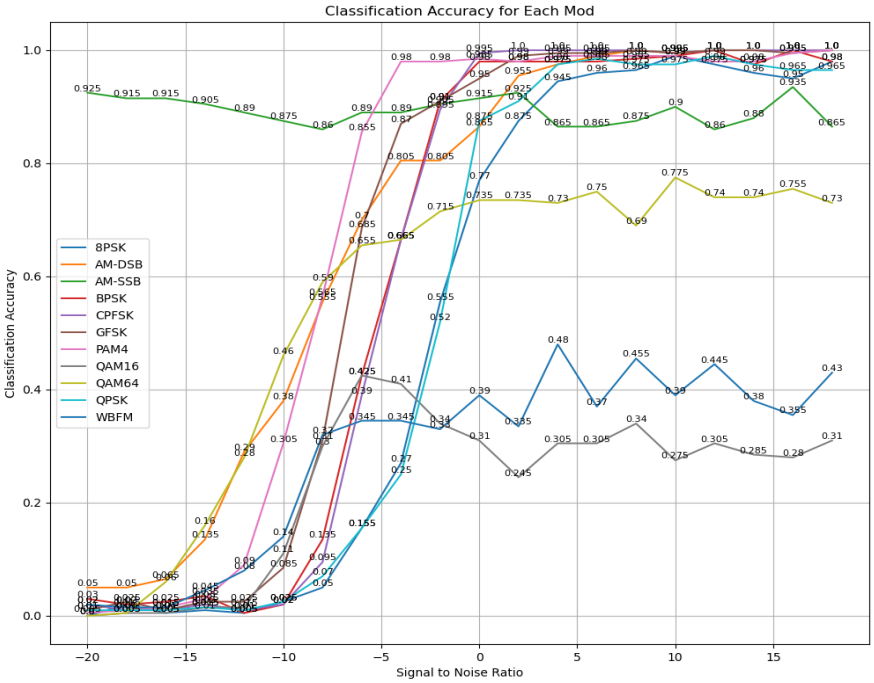


（c）

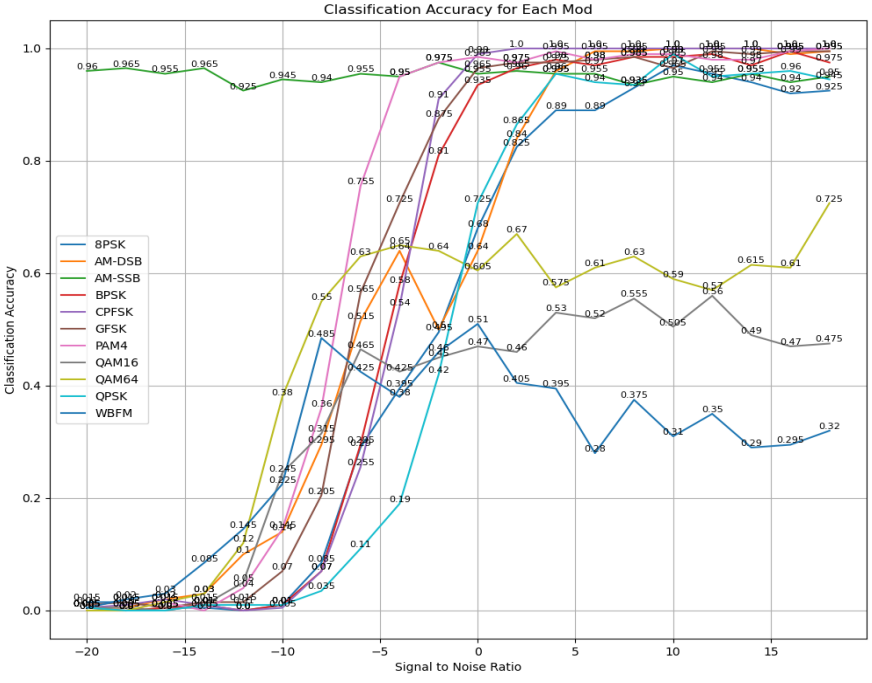


（d）

Figure 13. *Cont*.

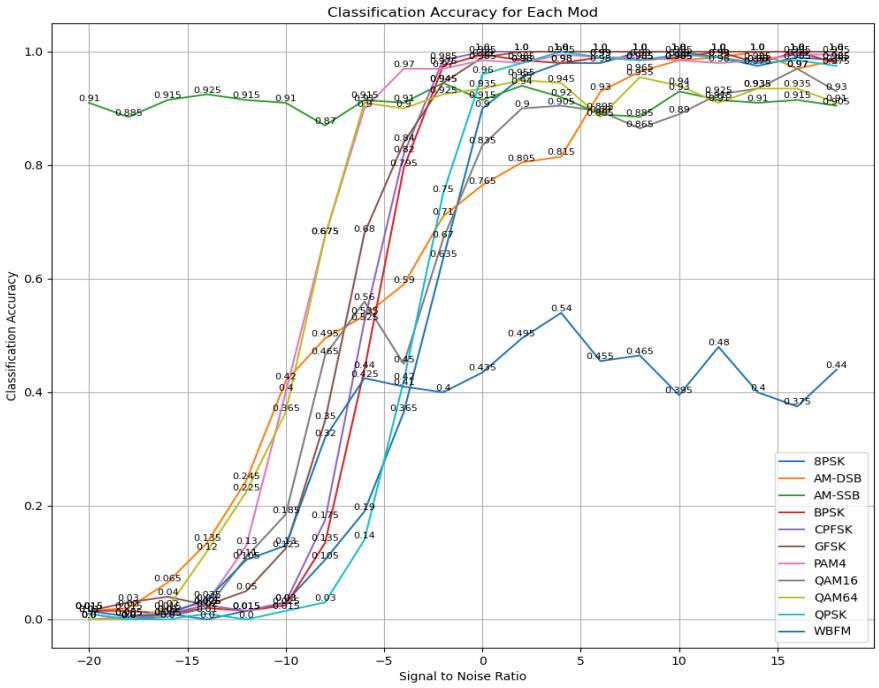


(e)



(f)

Figure 13. *Cont.*



（g）

Figure 13. Recognition accuracy for each modulated signal in the range of −20~18 db for all seven methods. (**a**) The recognition accuracy of each modulated signal using 1D-CNN. (**b**) The recognition accuracy of each modulated signal using 2D-CNN. (**c**) The recognition accuracy of CLDNN for each modulated signal. (**d**) The recognition accuracy of each modulated signal using DenseNet. (**e**) The recognition accuracy of each modulated signal using LSTM. (**f**) The recognition accuracy of each modulated signal using ResNET. (**g**) The recognition accuracy of the proposed model in this paper for each modulated signal.

In Figure 13a–f, it is observed that the recognition performance for modulation modes such as QAM16 and QAM64 is also unsatisfactory; this is attributed to the fact that the high-order modulation modes such as QAM16 and QAM64 possess more intricate signal structures, with denser and more complex distributions in the signal space. Moreover, these higher-order modulation modes are more susceptible to the effects of noise and interference during transmission, resulting in increased signal ambiguity and thereby, augmenting the difficulty of the recognition algorithm. In Figure 13a, the recognition rate curve for QPSK signals fluctuates around 60%, whereas in Figure 13b, there is a notable improvement, with the rate rising to approximately 80%. This indicates that the recognition achieved on the 2D-CNN model effectively addresses the shortcomings observed with the 1D-CNN model. The relatively stable performance of the other methods in recognizing QPSK signals can be attributed to their ability, including the proposed model in this paper, to effectively capture the temporal correlations within the sequences, thus facilitating better extraction of signal features.

Overall, as can be seen from Figures 12 and 13, the proposed method exhibits better classification performance relative to the other methods under most SNR conditions. Additionally, the proposed method demonstrates more stable and reliable classification results both in high and low SNR conditions. These results suggest that the proposed method holds great potential for recognizing QAM signals in multipath fading channel conditions and can serve as a promising research direction.

4.4. Computational Cost Analysis

The computational cost analysis is presented in Table 2. This includes the total parameters in the proposed model, the time required for each epoch during training, and the time required for classifying a single sample.

Table 2. Complexity comparison of RadioML 2016.10A dataset.

Models	(a) 1D-CNN	(b) 2D-CNN	(c) CLDNN	(d) DenseNet	(e) LSTM	(f) ResNet	(g) Proposed Model
Overall parameters	1,592,383	858,123	517,643	3,282,603	201,099	3,098,283	68,395
Training time s/epoch	7	18	17	36	8	27	3
Prediction time ms/epoch	53	140	135	277	62	210	26
Maximum GPU memory usage during training (MB)	2000	3500	4200	5000	3200	6500	3000
GPU memory usage during inference (MB)	1800	3200	3800	4500	2800	5800	2700
Memory changes during training (MB/s)	20	35	40	45	30	50	28
Memory changes during inference (MB/s)	15	25	30	35	22	40	20

Compared with the other evaluated models, the proposed model has the smallest number of parameters. This suggests that the proposed model may be more effective in terms of model complexity and may be less susceptible to overfitting since it learns fewer parameters from the training data. The proposed model has a fast learning speed: each round of training takes only 3 s, while 1D-CNN takes 7 s, 2D-CNN takes 18 s, and the training time of the other models is longer. Therefore, the efficiency performance of MPHNN in the training phase is excellent, which is especially suitable for application scenarios that require fast iteration. The prediction time of our model is 26 ms, which is significantly better than the other baseline models (e.g., ResNet 210 ms, DenseNet 277 ms, etc.), indicating fast inference for classifying a single sample. This fast prediction time is crucial for real-time applications that require fast decisions, such as in wireless communication systems or signal processing applications. During training, the maximum GPU memory usage of MPHNN is 3000 MB, which is more moderate compared to the other complex models such as ResNet's 6500 MB and DenseNet's 5000 MB. However, in the inference phase, the GPU memory usage of MPHNN is 2700 MB, which is lower than the other models, indicating the advantage of MPHNN in memory consumption. The memory change rate of MPHNN is 28 MB/s during training and 20 MB/s during inference, which is lower than the other models, which means that MPHNN performs well in the dynamic management of computing resources. With the supplement of these quantitative data, we further highlight the computational efficiency of MPHNN, especially when dealing with large-scale data or applying in resource-constrained environments, and its low computational complexity and memory footprint make it have better potential for practical applications.

Firstly, when comparing the performance of different methods/models, the statistical computation for the Friedman test is conducted. Ranks are calculated for each model on each metric, with the sum of ranks for the overall parameter ranking, training time ranking, and prediction time ranking being 32, 36, and 36, respectively. Subsequently, upon substituting these rank sums into the formula for the Friedman test statistic, a value of 0.857 is obtained.

$$T_F = \frac{(N-1)T\chi^2}{N(K-1) - T\chi^2}, \tag{17}$$

The statistical value obtained from the Friedman test is utilized to conduct the Nemenyi follow-up test. The Nemenyi post hoc test is employed to identify specific differences between models while the Friedman test indicates significant disparities. It relies on rank-

based comparisons and calculates a set of critical values to determine which differences between pairs of models are significant.

$$CD = q\alpha\sqrt{\frac{k(k+1)}{6N}}, \tag{18}$$

Given n = 3 and k = 7, with $\alpha = 0.1$ selected, referencing the table reveals $q\alpha = 2.693$. Thus, CD $\approx$ 4.750. Subsequently, this value can be utilized to compare the average rank differences between each pair of models to ascertain if significant differences exist. Refer to Table 3 for details.

Table 3. Significance assessment between models.

Number	Model	Model	Average Rank Difference	Compared to CD	Significant Difference
1	(a) 1D-CNN	(b) 2D-CNN	1	<	NO
2	(a) 1D-CNN	(c) CLDNN	2	<	NO
3	(a) 1D-CNN	(d) DenseNet	1	<	NO
4	(a) 1D-CNN	(e) LSTM	3	<	NO
5	(a) 1D-CNN	(f) ResNet	4	<	NO
6	(a) 1D-CNN	(g) Proposed Model	5	>	YES
7	(b) 2D-CNN	(c) CLDNN	1	<	NO
8	(b) 2D-CNN	(d) DenseNet	2	<	NO
9	(b) 2D-CNN	(e) LSTM	2	<	NO
10	(b) 2D-CNN	(f) ResNet	3	<	NO
11	(b) 2D-CNN	(g) Proposed Model	4	<	NO
12	(c) CLDNN	(d) DenseNet	3	<	NO
13	(c) CLDNN	(e) LSTM	1	<	NO
14	(c) CLDNN	(f) ResNet	2	<	NO
15	(c) CLDNN	(g) Proposed Model	3	<	NO
16	(d) DenseNet	(e) LSTM	4	<	NO
17	(d) DenseNet	(f) ResNet	5	>	YES
18	(d) DenseNet	(g) Proposed Model	6	>	YES
19	(e) LSTM	(f) ResNet	1	<	NO
20	(e) LSTM	(g) Proposed Model	2	<	NO
21	(f) ResNet	(g) Proposed Model	1	<	NO

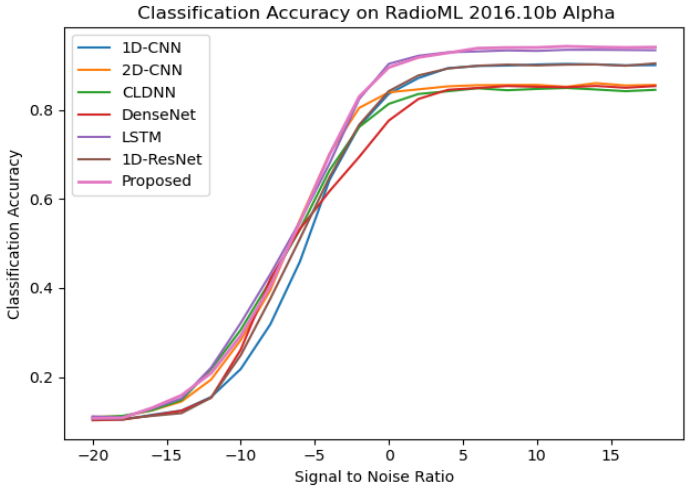
From the above table, it can be observed that the model combinations with significant differences are the 6th, 17th, and 18th pairs, while the model combinations with marginal differences are the 5th, 11th, and 16th pairs. Differences between the other models are not significant.

The RadioML2016.10B dataset includes 10 types of modulation: BPSK, QPSK, 8PSK, QAM16, QAM64, CPFSK, PAM4, GFSK, AM-DSB, and WBFM. The signal–noise ratio (SNR) range is from −20 to 18 dB in increments of 2 dB. The rest of the settings are consistent with RadioML2016.10A.

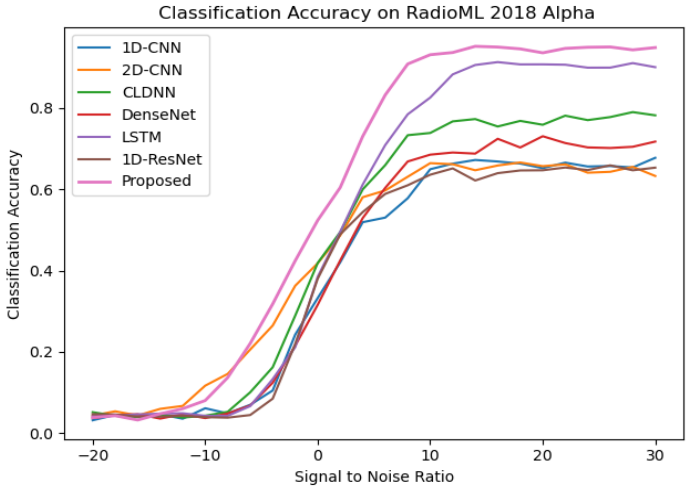
4.5. Validation on Other Datasets

The RadioML2018.01A dataset has three categories: X, Y, and Z. X is three-dimensional, with the first dimension containing 2,555,904 data entries, the second dimension containing 1024 data entries, and the third dimension containing 2 data entries. In total, there are 2,555,904 signals with 24 modulation types. There are 26 SNR levels, ranging from −20 to 30 dB in increments of 2 dB. Each SNR level has 4096 signal samples, with each sample consisting of 1024 data points (sampled at 1024 points), each composed of IQ (in-phase and quadrature) data. Y is two-dimensional and corresponds to the labels of each sample point in X. There are 24 modulation types, so a 24-bit one-hot encoding scheme is used. Z is also two-dimensional and corresponds to the SNR of each sample point in X, so it consists of a single data entry.

The MPHNN model not only performs well on the RadioML2016.10A dataset, but also on the RadioML2018.01A-sample and RadioML2016.10B datasets. Notably, it achieves a high accuracy of up to 94.8% on the RadioML2018.01A-sample dataset. Additionally, Figure 14 illustrates that the MPHNN model consistently outperforms several other models across various signal–noise ratios (SNRs), showcasing its robustness under different conditions. A key factor contributing to its superior performance is the incorporation of attention mechanisms within the model architecture. This mechanism effectively captures temporal, spatial, and salient features, providing a significant advantage over other models. The attention mechanism enables the model to focus on relevant information while filtering out noise, thereby enhancing recognition accuracy, particularly at low SNRs. Overall, the remarkable performance of the MPHNN model highlights its effectiveness in handling various datasets and challenging conditions, making it a promising candidate for practical applications in signal processing and recognition tasks.



（a）



（b）

Figure 14. Validation on other datasets: (**a**) RadioML2016.10B and (**b**) RadioML2018.01A-sample.

4.6. Description of Model Uncertainty

The sources of model uncertainty primarily encompass two aspects:

(a) Parameter Uncertainty in the Model: The MPHNN model comprises numerous parameters, the values of which may exhibit uncertainty during the training process or be influenced by data noise. This implies that the accuracy of the model parameters may be somewhat compromised, consequently affecting the model's classification and prediction capabilities.

(b) Uncertainty Caused by Input Data Noise: The presence of noise in the input data can lead to increased uncertainty in the output results, particularly when identifying modulation types under low signal–noise ratio (SNR) conditions. The existence of noise can make it challenging for the model to accurately differentiate between different signal types, thereby augmenting classification uncertainty.

To address model uncertainty, this study employs a representation and estimation method that utilizes the probability distribution of the model's output to indicate uncertainty in classifying each modulation category. The softmax output layer is utilized to obtain probability values or confidence levels for each category, reflecting the model's confidence in classifying each modulation type.

5. Conclusions

In this paper, an MPHNN based on a spatio-temporal attention mechanism is proposed, which can effectively utilize the information from multimodal signals for the classification prediction of modulation modes. The model extracts the local features of each input modality through multiple parallel convolutional layers, and fuses the features of different modalities through a feature fusion layer. Meanwhile, CNN and Bi-GRU are used in the model structure to extract spatial and temporal correlated features. To better utilize the spatio-temporal information, a spatio-temporal attention mechanism layer is introduced to weigh the features with different time steps and spatial correlations of the signals. Finally, the global features are extracted through a global feature extraction layer, and the classification prediction of modulation modes is carried out by a classifier layer. This method can effectively handle multimodal signals and extract useful features from them for classification tasks. To evaluate the performance of the MPHNN, several baseline networks such as 1D-CNN, 2D-CNN, CLDNN, DenseNet, LSTM, and ResNet are compared through experiments on the RadioML2016.10A, RadioML2016.10B, and RadioML2018.01A-sample datasets. The experimental results demonstrate that the proposed network achieves the best results at most SNRs, with fewer learning parameters, lower memory costs, higher efficiency, and greater robustness compared to other models. It also shows significant potential for AMR.

Although MPHNN demonstrates excellent performance in multimodal modulation recognition, it still has certain limitations, particularly a decline in recognition accuracy in high-noise and complex environments, as well as difficulties in distinguishing easily confusable modulation schemes. Additionally, despite the incorporation of attention mechanisms, the model's computational complexity and long training time limit its application in real-time systems. In future work, MPHNN will be applied to a variety of different datasets to verify its robustness in different signal environments. Future research could focus on improving the model's robustness in low signal–noise ratio (SNR) environments, optimizing feature fusion strategies, reducing modulation scheme confusion, and exploring lightweight network architectures and efficient training algorithms to shorten both training and inference time. At the same time, approaches such as cross-domain learning and meta-learning could enhance the model's adaptability, paving the way for broader applications of MPHNN in fields like wireless communications and radar systems.

Author Contributions: Conceptualization, W.Z. and Y.S.; methodology, W.Z.; software, W.Z.; validation, W.Z., K.X. and Y.S.; formal analysis, Y.S.; investigation, W.Z.; resources, Y.S.; data curation, W.Z.; writing—original draft preparation, W.Z.; writing—review and editing, Y.S. and A.Y.; visualization, W.Z.; supervision, Y.S.; project administration, Y.S.; funding acquisition, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: This dataset is downloaded from https://www.deepsig.ai/datasets (accessed on 20 October 2022). The name of the datasets are RadioML2016.10A, RadioML2016.10B, and RadioML2018.01A.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Huang, S.; Yao, Y.; Wei, Z.; Feng, Z.; Zhang, P. Automatic Modulation Classification of Overlapped Sources Using Multiple Cumulants. *IEEE Trans. Veh. Technol.* **2017**, *66*, 6089–6101. [CrossRef]
2. Li, L.; Zhu, Y.; Zhu, Z. Automatic modulation classification using ResNeXt-GRU with deep feature fusion. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2519710. [CrossRef]
3. Jagannath, A.; Jagannath, J. Multi-task learning approach for modulation and wireless signal classification for 5G and beyond: Edge deployment via model compression. *Phys. Commun.* **2022**, *54*, 101793. [CrossRef]
4. Saad, W.; Bennis, M.; Chen, M.Z. A vision of 6G wireless systems: Applications, trends, technologies, and open research problems. *IEEE Netw.* **2020**, *34*, 134–142. [CrossRef]
5. Liang, L.; Ye, H.; Yu, G.D.; Li, G.Y. Deep-learning-based wireless resource allocation with application to vehicular networks. *Proc. IEEE* **2020**, *108*, 341–356. [CrossRef]
6. Jdid, B.; Hassan, K.; Dayou, I.; Lim, W.H.; Mokayef, M. Machine learning based automatic modulation recognition for wireless communications: A comprehensive survey. *IEEE Access* **2021**, *9*, 57851–57873. [CrossRef]
7. Li, T.; Xiao, Y.Z. Domain Adaptation-Based Automatic Modulation Recognition. *Sci. Program.-Neth.* **2021**, *2021*, 4277061. [CrossRef]
8. Zheng, Q.; Tian, X.; Yu, Z.; Wang, H.; Elhanashi, A.; Saponara, S. Generalized automatic modulation classification method based on deep learning with priori regularization. *Eng. Appl. Artif. Intel.* **2023**, *122*, 106082. [CrossRef]
9. Che, J.; Wang, L.; Bai, X.; Liu, C.; Zhou, F. Spatial-temporal hybrid feature extraction network for few-shot automatic modulation classification. *IEEE Trans. Veh. Technol.* **2022**, *71*, 13387–13392. [CrossRef]
10. Salama, A.A.; Morsy, M.E.S.H.; Darwish; Mohamed, E.I. A novel SVM-based automatic modulation classifier. In Proceedings of the 2022 International Telecommunications Conference (ITC-Egypt), Alexandria, Egypt, 26–28 July 2022. [CrossRef]
11. Xie, W.W.; Hu, S.; Yu, C.; Zhu, P.; Peng, X.; Ouyang, J. Deep learning in digital modulation recognition using high order cumulants. *IEEE Access* **2019**, *7*, 63760–63766. [CrossRef]
12. Fang, T.; Wang, Q.; Zhang, L.; Liu, S. Modulation Mode Recognition Method of Non-Cooperative Underwater Acoustic Communication Signal Based on Spectral Peak Feature Extraction and Random Forest. *Remote Sens.* **2022**, *14*, 1603. [CrossRef]
13. Luan, S.Y.; Gao, Y.R.; Liu, T.; Li, J.Y.; Zhang, Z.J. Hierarchical Blind Modulation Classification for Underwater Acoustic Communication Signal via Cyclostationary and Maximal Likelihood Analysis. *Digit. Signal. Process.* **2022**, *126*, 103476. [CrossRef]
14. Peng, S.; Jiang, H.; Wang, H.; Alwageed, H.; Zhou, Y.; Sebdani, M.M.; Yao, Y.D. Modulation classification based on signal constellation diagrams and deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 718–727. [CrossRef] [PubMed]
15. Xie, B.; Ma, C.F.; Li, H.Q.; Zhang, G.Y.; Han, C.Z. Simple and Robust Log-Likelihood Ratio Calculation of Coded MPSK Signals in Wireless Sensor Networks for Healthcare. *Appl. Sci.* **2022**, *12*, 2330. [CrossRef]
16. Kim, T.; Lee, S.H.; Kim, J.O. A Novel Shape Based Plant Growth Prediction Algorithm Using Deep Learning and Spatial Transformation. *IEEE Access* **2022**, *10*, 37731–37742. [CrossRef]
17. Abbaschian, B.J.; Sierra-Sosa, D.; Elmaghraby, A. Deep learning techniques for speech emotion recognition, from databases to models. *Sensors* **2021**, *21*, 1249. [CrossRef]
18. Agnihotri, D.; Verma, K.; Tripathi, P. Variable global feature selection scheme for automatic classification of text documents. *Expert Syst. Appl.* **2017**, *81*, 268–281. [CrossRef]
19. Zhao, B.; Chen, X.B.; Le, X.Y.; Xi, J.T. A quantitative evaluation of comprehensive 3D local descriptors generated with spatial and geometrical feature. *Comput. Vis. Image Underst.* **2020**, *190*, 102842. [CrossRef]
20. Moshayedi, A.J.; Roy, A.S.; Kolahdooz, A.; Shuxin, Y. Deep Learning Application Pros And Cons Over Algorithm. *EAI Endorsed Trans. AI Robot.* **2022**, *1*, e7. [CrossRef]
21. Hu, G.; Meng, X.; Wang, X.; Liu, Z. A Novel Explainable Impedance Identification Method Based on Deep Learning for the Vehicle-grid System of High-speed Railways. *IEEE Trans. Transp. Electrif.* **2024**. [CrossRef]
22. Wang, H.; Liu, Z.; Wang, X. Model-Based Data-Efficient Reinforcement Learning for Active Pantograph Control in High-Speed Railways. *IEEE Trans. Transp. Electrif.* **2024**, *10*, 2701–2712. [CrossRef]
23. O'Shea, T.J.; Corgan, J.; Clancy, T.C. Convolutional radio modulation recognition networks. In *International Conference on Engineering Applications of Neural Networks (EANN)*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 213–226. Available online: https://www.arxiv.org/abs/1602.04105v3 (accessed on 5 October 2023).
24. West, N.E.; O'Shea, T.J. Deep architectures for modulation recognition. In Proceedings of the 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Baltimore, MD, USA, 6–9 March 2017; IEEE: New York, NY, USA, 2017; pp. 1–6.
25. Wang, H.; Guo, L.L.; Lin, Y. Modulation recognition of digital multimedia signal based on data feature selection. *Int. J. Mob. Comput. Multimed. Commun. IJMCMC* **2017**, *8*, 90–111. [CrossRef]
26. Ettefagh, Y.; Moghaddam, M.H.; Eghbalian, S. An adaptive neural network approach for automatic modulation recognition. In Proceedings of the 2017 51st Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 22–24 March 2017; IEEE: New York, NY, USA, 2017; pp. 1–5.
27. Zhang, D.N.; Ding, W.R.; Zhang, B.C.; Xie, C.Y.; Li, H.G.; Liu, C.H.; Han, J.G. Automatic Modulation Classification Based on Deep Learning for Unmanned Aerial Vehicles. *Sensors* **2018**, *18*, 924. [CrossRef] [PubMed]
28. Rajendran, S.; Meert, W.; Giustiniano, D.; Lenders, V.; Pollin, S. Deep learning models for wireless signal classification with distributed low-cost spectrum sensors. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 433–445. [CrossRef]

29. Tayakout, H.; Ghanem, K.; Bousbia-Salah, H. On classifiers for feature-based automatic modulation recognition over D-STBC cooperative networks. In Proceedings of the 2019 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting, Atlanta, GA, USA, 7–12 July 2019; IEEE: New York, NY, USA, 2019; pp. 1839–1840.
30. Wang, T.; Jin, Y. Modulation Recognition Based on Lightweight Neural Networks. In Proceedings of the 2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Chengdu, China, 17–19 October 2020; IEEE: New York, NY, USA, 2020; pp. 468–472.
31. Liu, Y.; Liu, Y.; Yang, C. Modulation recognition with graph convolutional network. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 624–627. [CrossRef]
32. Jafar, N.; Paeiz, A.; Farzaneh, A. Automatic modulation classification using modulation fingerprint extraction. *J. Syst. Eng. Electron.* **2021**, *32*, 799–810. [CrossRef]
33. Li, K.; Shi, J. Modulation Recognition Algorithm based on Digital Communication Signal Time-Frequency Image. In Proceedings of the 2021 8th International Conference on Dependable Systems and Their Applications (DSA), Yinchuan, China, 5–6 August 2021; IEEE: New York, NY, USA, 2021; pp. 747–748.
34. Wang, Y.; Gui, G.; Ohtsuki, T.; Adachi, F. Multi-task learning for generalized automatic modulation classification under non-Gaussian noise with varying SNR conditions. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 3587–3596. [CrossRef]
35. Shi, F.; Hu, Z.; Yue, C.; Shen, Z. Combining neural networks for modulation recognition. *Digit. Signal Process.* **2022**, *120*, 103264. [CrossRef]
36. Ansari, S.; Alnajjar, K.A.; Saad, M.; Abdallah, S.; El-Mours, A.A. Automatic Digital Modulation Recognition Based on Genetic-Algorithm-Optimized Machine Learning Models. *IEEE Access* **2022**, *10*, 50265–50277. [CrossRef]
37. Zhang, X.X.; Zhao, H.T.; Zhu, H.B.; Adebisi, B.; Gui, G.; Gacanin, H.; Adachi, F. NAS-AMR: Neural Architecture Search Based Automatic Modulation Recognition for Integrated Sensing and Communication Systems. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 1374–1386. [CrossRef]
38. Qi, P.; Zhou, X.; Zheng, S.; Li, Z. Automatic modulation classification based on deep residual networks with multimodal information. *IEEE Trans. Cogn. Commun. Netw.* **2021**, *7*, 21–33. [CrossRef]
39. Meng, X.; Hu, G.; Liu, Z.; Wang, H.; Zhang, G.; Lin, H.; Sadabadi, M.S. Neural Network-Based Impedance Identification and Stability Analysis for Double-Sided Feeding Railway Systems. *IEEE Trans. Transp. Electrif.* **2024**. [CrossRef]
40. Zhang, W.; Sun, Y.; Xue, K.; Yao, A. Research on Modulation Recognition Algorithm Based on Channel and Spatial Self-Attention Mechanism. *IEEE Access* **2023**, *11*, 68617–68631. [CrossRef]
41. Kohsasih, K.L.; Zarlis, M.; Hayadi, B.H. Comparison of CNN Architecture for White Blood Cells Image Classification. In Proceedings of the 2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM), Laguboti, Indonesia, 19–21 October 2022; pp. 1–7.
42. Lu, L.M.; Zhang, C.L.; Cao, K.; Deng, T.; Yang, Q.Q. A Multichannel CNN-GRU Model for Human Activity Recognition. *IEEE Access* **2022**, *10*, 66797–66810. [CrossRef]
43. Xiao, W.S.; Luo, Z.Q.; Hu, Q. A Review of Research on Signal Modulation Recognition Based on Deep Learning. *Electronics* **2022**, *11*, 2764. [CrossRef]
44. Huang, G.; Liu, Z.; Pleiss, G.; Maaten, L.V.D.; Weinberger, K.Q. Convolutional Networks with Dense Connectivity. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 8704–8716. [CrossRef]
45. Ke, Z.; Vikalo, H. Real-time radio technology and modulation classification via an LSTM auto-encoder. *IEEE Trans. Wireless Commun.* **2022**, *21*, 370–382. [CrossRef]
46. Huang, S.; Dai, R.; Huang, J.; Yao, Y.; Feng, Z. Automatic Modulation Classification Using Gated Recurrent Residual Network. *IEEE Internet Things J.* **2020**, *7*, 7795–7807. [CrossRef]