sensors

MDPI

Article

AMSCN: A Novel Dual-Task Model for Automatic Modulation Classification and Specific Emitter Identification

Shanchuan Ying [†], **Sai Huang** [*,†], **Shuo Chang**, **Jiashuo He** and **Zhiyong Feng**

Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China

* Correspondence: huangsai@bupt.edu.cn

† These authors contributed equally to this work.

Abstract: Specific emitter identification (SEI) and automatic modulation classification (AMC) are generally two separate tasks in the field of radio monitoring. Both tasks have similarities in terms of their application scenarios, signal modeling, feature engineering, and classifier design. It is feasible and promising to integrate these two tasks, with the benefit of reducing the overall computational complexity and improving the classification accuracy of each task. In this paper, we propose a dual-task neural network named AMSCN that simultaneously classifies the modulation and the transmitter of the received signal. In the AMSCN, we first use a combination of DenseNet and Transformer as the backbone network to extract the distinguishable features; then, we design a mask-based dual-head classifier (MDHC) to reinforce the joint learning of the two tasks. To train the AMSCN, a multitask cross-entropy loss is proposed, which is the sum of the cross-entropy loss of the AMC and the cross-entropy loss of the SEI. Experimental results show that our method achieves performance gains for the SEI task with the aid of additional information from the AMC task. Compared with the traditional single-task model, our classification accuracy of the AMC is generally consistent with the state-of-the-art performance, while the classification accuracy of the SEI is improved from 52.2% to 54.7%, which demonstrates the effectiveness of the AMSCN.

Keywords: specific emitter identification (SEI); automatic modulation classification (AMC); deep learning; multitask learning

check for updates

Citation: Ying, S.; Huang, S.; Chang, S.; He, J.; Feng, Z. AMSCN: A Novel Dual-Task Model for Automatic Modulation Classification and Specific Emitter Identification. *Sensors* **2023**, *23*, 2476. https://doi.org/10.3390/s23052476

Academic Editors: Ekin Ozer, Shizhi Chen, Jingfeng Zhang, Zilong Ti and Xiaoming Lei

Received: 7 January 2023
Revised: 18 February 2023
Accepted: 20 February 2023
Published: 23 February 2023

1. Introduction

Radio monitoring helps spectrum management agencies to plan and use frequencies [1], avoid incompatible uses [2], and identify sources of harmful interference [3]. With the development of wireless communication, ensuring the security of the communication process has become a topic of great concern [4,5]. A typical scenario is that in satellite communications, the ground station needs to receive signals relayed by satellites or other ground stations. In some civil–military satellite systems, both military satellites and civilian relay satellites may access the core network through the ground station, so the ground station needs to have the ability to discriminate radio signals to prevent interfering satellites or hostile satellites from accessing the network. If the authorization is based on the secret key, the ground station will face huge pressure when signal resolving because of the large number of signals attempting to gain access, and in many cases, the ground station is only responsible for relaying, and does not have the ability to resolve the signal content. Therefore a physical-layer-security-based authentication scheme is feasible. In addition, as satellite Internet technology is put into practice, more and more countries need to regulate the satellite networks of other countries, including service hours, service bands, radio signal range, and communication traffic. Specific emitter identification (SEI) and automatic modulation classification (AMC) are two common identification tasks for signal characteristics. SEI is a process of extracting individual characteristics from the signals and identifying the communication transmitters. These individual characteristics of a transmitter are often referred to as radio frequency (RF) fingerprints [6–9]. AMC is a process of blindly identifying the modulation format of an unknown received signal [10–12]. In the aforementioned communication scenario, the transmitter may change the current modulation at any time, and the receiver will be disturbed by the change in modulation when performing RF fingerprint recognition on the current signal. How to improve the efficiency of RF fingerprint recognition in the case of variable modulation becomes particularly important.

Generally speaking, the generalized modulation of signals is a process that makes certain characteristics of one waveform change according to another waveform or signal. The instantaneous amplitude, instantaneous phase, and other information of the baseband signal are not only artificially changed by the message signal, but they are also unintentionally modulated by the defects of the RF hardware. Therefore, both the AMC task and the SEI task can be viewed as the process of identifying the generalized modulation information from the received signal. However, in the existing research, these two tasks have been conducted separately and have not been studied together. In some specific monitoring scenarios, the need to identify both the modulation format and the RF fingerprints of the signal at the receiver side may exist simultaneously, and if we can use one model for both tasks, then the classification efficiency of the system can be improved.

In our work, we believe that it is feasible to implement both the AMC task and the SEI task in a single model for two reasons. Firstly, both are classification tasks, and they both extract the distinguishable features from the received signal. Secondly, in terms of the model design, the network structure of the two tasks is quite similar, which indicates that we can use the same network to perform both tasks. Moreover, the modulation and transmitter characteristics are contained within the same segment of the signal, and they both have similar significant impacts on the waveform of the signal. In traditional single-task detection, if a model is only used to identify modulation, then variations in the transmitter characteristics in the signal are viewed as interference to the AMC, and vice versa. If we learn these two characteristics simultaneously in one model, i.e., using two labels to guide the learning process of the network, then both kinds of information are valid for the model, which can facilitate the model to better distinguish between the two tasks.

In this paper, we design a framework for AMC and SEI signal characteristics classification using a multitask learning approach to mine the correlations between them and improve the recognition efficiency of both tasks. The contributions of the paper are as follows:

- We propose an AMC-mask-based SEI Classification Network (AMSCN) for the AMC and SEI. To our knowledge, this is the first approach to consider these two classification tasks together;
- In the AMSCN, we design a multitask classification model based on deep learning, which consists of a backbone network and a mask-based dual-head classifier (MDHC). The backbone network has a DenseNet–Transformer structure, which is responsible for extracting discriminative features that can be adapted to different signal feature scales in both tasks;
- The MDHC consists of an AMC head and an SEI head. It can enhance the correlation between the two tasks through a mask mechanism and finally output the classification results of the two tasks. With the help of the MDHC, we are able to balance the learning process using only the sum of the cross-entropy losses of the two tasks;
- We generate a simulated dataset for the AMC and SEI tasks. Extensive experiments are carried out on this simulated dataset to demonstrate that the fusion of AMC and SEI can achieve better predictions than single-task learning. Furthermore, some contrast experiments have also been conducted to verify the effectiveness of each module in the AMSCN.

This paper is organized as follows. In Section 2, we discuss the common feature design and model design approaches in the AMC and SEI domains, as well as the design of multitask models in deep learning. In Section 3, we introduce the entire classification framework, including the system model and the signal model. Section 4 details the design of the core module and the training method of the AMSCN. The results in terms of the accuracy and the additional ablation experiments are shown in Section 5. Finally, the conclusion and the prospective research activities are presented in Section 6.

2. Related Work

Through the development of artificial intelligence (AI), AI-based applications have entered every aspect of human society, including industry [13], agriculture [14], healthcare [15], and education [16]. In wireless communications, deep-learning-based methods also introduced a new way to consider the problem of signal characteristics recognition [17–25]. For the AMC and SEI tasks, the deep-learning-based approach is essentially a statistical pattern recognition problem, which can be divided into two steps: feature extraction and pattern recognition [26]. That is, first, we extract the reference features from the received signal; then, we judge the modulation type or RF fingerprints based on these features. In addition to extracting the features based on time–domain waveforms [8,27–29], there are also preprocessing methods that convert waveforms into explicit features for deeper feature extraction, such as time–frequency diagrams [30], spectrograms [31], higher-order cumulants [32], wavelet transform features [33], cyclostationary features [34], constellation diagrams [23,35], etc [36,37]. These preprocessing methods are applicable to both AMC and SEI tasks almost simultaneously, indicating the similarity of the essential characteristics of the AMC and SEI tasks. In terms of the model design, the two tasks also share similarities; for example, structures, such as regular convolution neural networks (CNNs) [22,38], ResNet [37,39], Inception [12], DenseNet [40,41], long short-term memory (LSTM) [42], and Transformer [7,43], are present in both tasks. The model of one task can achieve good results on the other task after some simple modifications.

In deep learning, multitask learning has been widely researched. Multitask learning is a method that enables a model to have better generalization performance on the original task by sharing feature representations between related tasks [44]. The benefits of multitask learning are manifold; it not only saves network parameters through hard parameter sharing [45,46] but also reduces the overfitting of the model on a single task [47]. In the field of radio signal classification, there is no multitask learning model combining AMC and SEI, but some multitask learning applications exist in other scenarios. In [48], an algorithm to simultaneously learn the modulation method and signal-to-noise ratio was proposed, and the results showed that the addition of a new task improved the classification efficiency of the AMC. In [49], a multitask deep convolutional neural network was proposed to perform a modulation classification and direction-of-arrival (DOA) estimation simultaneously.

Apart from multitask learning, attention mechanisms are increasingly applied to signal detection. Ref. [43] used an R-transformer-based model to achieve state-of-the-art performance on the AMC task. Ref. [50] was inspired by the high-quality representation capability of Transformer and proposed a novel openset SEI algorithm to find accurate and stable boundary samples with more robust representations. In our practice, we found that the self-attention mechanism or Transformer structure could effectively capture the changes in global information in the case of longer sequences, and thus, it is more suitable than LSTM for the tasks of AMC and SEI, which have large differences in feature scales.

3. System Model and Problem Statement

In this section, we provide an assumption of the application scenario, the workflow of the whole framework, and the signal model of the simulated dataset.

3.1. System Model

We assumed that there was an application scenario where the transmitter had multiple similar devices that could establish communication with the receiver, and that these devices constantly changed the modulation when sending signals; meanwhile, the receiver only received signals from one device at the same time. In order to achieve the effect of safe signal accessing, the receiver needed to determine the modulation of the current signal for subsequent algorithm demodulation, as well as the individual transmitter to which the current signal belongs. Figure 1 shows the framework of our proposed multitask classification method. The receiver acquires the signal and transforms it into a complex baseband signal sequence $x(n)$ with a length of N. The real part of each complex sampling point is called the in-phase (I) component, and its imaginary part is called the orthogonal (Q) component. Then, the proposed AMSCN is applied to predict both the modulation and the transmitter of the received signal, where $H_{i,j}$ indicates that the current signal uses the ith modulation format in the modulation set, together with the jth device in the transmitter set. Finally, these predictions can be used to support subsequent communication applications, such as demodulation and access authorization.

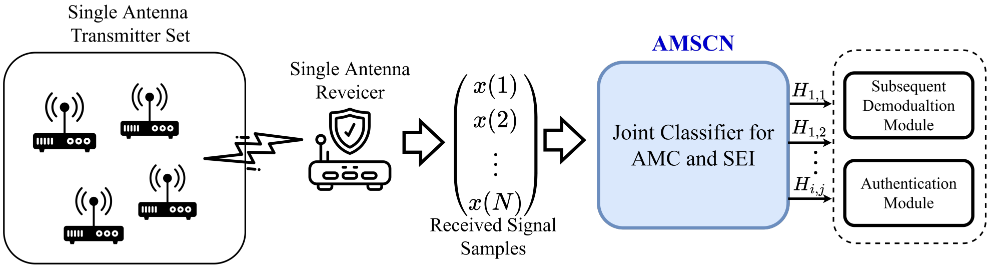

Figure 1. The modulation and transmitter joint-identification framework.

3.2. Signal Model

In a communication system, the transmitter sends information by changing the amplitude, frequency, or phase of the carrier signal. Figure 2 illustrates the signal generation process in a simplified zero intermediate frequency (IF) transmitter. Baseband modulation means the process of mapping from the bit stream to the symbol stream. Different modulation formats correspond to different symbol patterns; thus, modulation classification identifies the symbol patterns according to the received signals. Since the baseband signal can usually be expressed in plural form, the output of the modulation module is also called the baseband IQ signal. Up-conversion block refers to the process of moving the baseband IQ signal from the lower frequency to the carrier frequency. A digital-to-analog converter (DAC) transforms the digital signal into an analog signal, which is then mixed with two separate quadrature carriers and summed together. Before the synthesized RF signal is sent through the antenna, it has to be amplified to ensure that the signal can travel a sufficient distance.

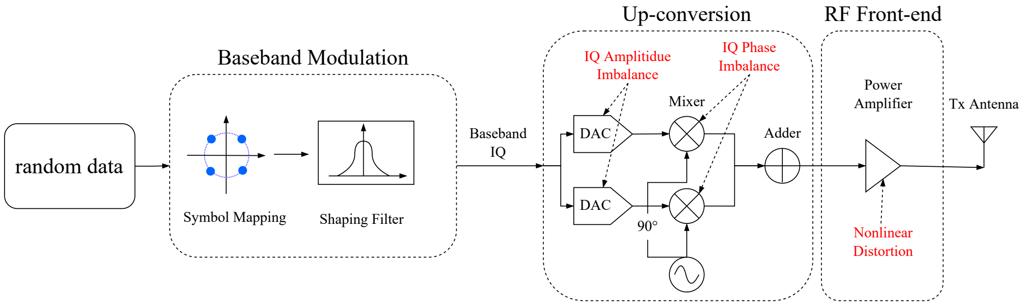

Figure 2. The process of digital modulation and up-conversion.

The signal output from the baseband modulation module can be expressed as

$$s^i(t) = \sum_{k=-\infty}^{+\infty} g(t - kT_s)S_k^i, \tag{1}$$

where $s^i(t)$ is the time-domain signal expression of the ith modulation, $g(t)$ is the time-domain response of the shaping filter, T_s is the symbol period, and S_k^i is the symbol sequence obtained using the ith modulation. In the up-conversion block, there is a certain degree of amplitude imbalance and carrier-phase non-orthogonality between the I and Q signals due to manufacturing defects of the DAC and the mixer. The signal output from up-conversion module can be expressed as

$$\hat{s}(t) = \big(\mu s(t) + v s^*(t)\big)e^{j2\pi f_c t} \tag{2}$$

with

$$\mu = \frac{1}{2}(\alpha + 1)cos(\frac{\pi\beta}{360}) + \frac{j}{2}(\alpha - 1)sin(\frac{\pi\beta}{360}) \tag{3}$$

$$v = \frac{1}{2}(\alpha - 1)cos(\frac{\pi\beta}{360}) + \frac{j}{2}(\alpha + 1)sin(\frac{\pi\beta}{360}) \tag{4}$$

$$\alpha = 20\log\frac{Gain_I}{Gain_Q}, \tag{5}$$

where $\hat{s}(t)$ is the signal under the complex conjugate model and represents the RF signal generated by a quadrature modulator with IQ imbalance. $s(t)$ is the IQ signal in Equation (1) without superscript, and $s^*(t)$ is the conjugate of $s(t)$. f_c is the carrier frequency. $Gain_I$ is the gain in the in-phase branch, $Gain_Q$ is the gain in the quadrature branch, and α denotes the amplitude imbalance between the two DACs. β is the extent to which the two orthogonal carriers deviate from orthogonality, expressed in radians. After being amplified by a non-ideal amplifier function $PA[\cdot]$, the signal sent by the antenna can be expressed as

$$RF(t) = PA[\hat{s}(t)] \tag{6}$$

In wireless communication systems, since the carrier frequency is usually much higher than the modulated signal bandwidth, the nonlinearity of the amplifier can be approximated to be frequency independent. $RF(t)$ can be written as

$$\begin{aligned} RF(t) &= PA\Big[\big(\mu s(t) + v s^*(t)\big)e^{j2\pi f_c t}\Big] && (7)\\ &= PA\big[\big(\mu s(t) + v s^*(t)\big)\big]e^{j2\pi f_c t} && (8)\end{aligned}$$

At the receiver side, the baseband signal after down-conversion can be expressed as

$$\begin{aligned} y(t) &= [h(t) * RF(t)]e^{-j2\pi f_{LO} t + \theta_0} + n(t) && (9)\\ &= PA\big[\big(\mu s(t) + v s^*(t)\big)\big]e^{j2\pi f_c t} \cdot e^{-j2\pi f_{LO} t + \theta} + n(t) && (10)\\ &= PA\big[\big(\mu s(t) + v s^*(t)\big)\big]e^{j2\pi (f_c - f_{LO}) t + \theta} + n(t) && (11)\end{aligned}$$

where $h(t)$ denotes the impulse response of the channel. In this work, for the purpose of simplifying the problem, we assume that the signal is only affected by Gaussian noise and the interference of channel variations on the detection effect is ignored, so we set the function $h(t)$ to a constant. $n(t)$ is the additive white Gaussian noise (AWGN) with zero mean and variance σ^2. f_{LO} is the demodulation carrier frequency at the receiver, and θ is the demodulation carrier-phase offset. We define the signal-to-noise ratio (SNR) of the received signal as $\gamma = 1/\sigma^2$. From Equation (11), it can be seen that the nonlinear mapping of the amplifier in the received signal acts equivalently on the baseband signal at the transmitter; therefore, the mapping process of the amplifier can be described using an equivalent baseband model.

To determine the form of the $PA[\cdot]$ function, we chose several memoryless amplifier models to simulate the nonlinear amplification behavior of different transmitter individuals. Meanwhile, we referred to the model parameter settings in the published literature [51–54] to approximate the real-device characteristics. To simplify the analysis, we assumed that the equivalent baseband signal of the frequency band signal input to the amplifier was

$$\hat{s}(n) = r(n) \cdot exp[j\phi(n)], \tag{12}$$

where $r(n)$ is the instantaneous amplitude, and $\phi(n)$ is the instantaneous phase; then, the output of the equivalent baseband amplifier model can be expressed as

$$y(n) = A[r(n)] \cdot exp[j\phi(n) + \Phi(r(n))], \tag{13}$$

where $A[\cdot]$ and $\Phi[\cdot]$ denote the AM/AM distortion and AM/PM distortion effects of the amplifier, respectively.

The Saleh model [51,52] can be used to describe the nonlinear characteristics of the traveling wave tube amplifier (TWTA), and it is widely used in the simulation of satellite communication systems. This model can be determined by the following four parameters α_α, β_α, α_ϕ, and β_ϕ, and its AM/AM and AM/PM response functions can be expressed, respectively, as

$$A[r(n)] = \frac{\alpha_\alpha r(n)}{1 + \beta_\alpha r^2(n)} \tag{14}$$

$$\Phi[r(n)] = \frac{\alpha_\phi r^2(n)}{1 + \beta_\phi r^2(n)}. \tag{15}$$

The Rapp model [53] is applied to solid-state power amplifiers (SSPA), and this model considers that the phase distortion of the signal is relatively small and therefore negligible. Its AM/AM and AM/PM response functions can be expressed, respectively, as

$$A[r(n)] = \frac{r(n)}{[1 + (\frac{r(n)}{V_{sat}})^{2p}]^{\frac{1}{2p}}} \tag{16}$$

$$\Phi[r(n)] \approx 0, \tag{17}$$

where V_{sat} is the saturation output voltage of the amplifier, p is the smoothness factor, and the larger the p value, the more linearized the amplifier will be.

The CMOS model [54] is based on the Rapp model, which requires that its AM/AM characteristics obey the Rapp model criteria, while its AM/PM characteristics are also nonlinear rather than constant, which can be expressed as

$$A[r(n)] = \frac{r(n)}{[1 + (\frac{r(n)}{V_{sat}})^{2p}]^{\frac{1}{2p}}} \tag{18}$$

$$\Phi[r(n)] = \frac{d \cdot r^f(n)}{1 + [\frac{r(n)}{e}]^g}, \tag{19}$$

where d, e, f, and g are the model parameters that control the degree of the phase nonlinearity.

4. AMSCN: AMC Mask-Based SEI Classification Network

This section is divided into two subsections. In the first, we discuss the overall training process of the AMSCN, as well as the objective function of the model. In the second section, we give the implementation details of the DenseNet part, the Transformer part, and the mask-based dual-head classifier (MDHC).

4.1. Offline Training Process

A supervised learning algorithm in deep learning aims to learn a mapping from the input to the output given a training set of inputs x and outputs y. Figure 3 provides the offline training process of our proposed AMSCN model.

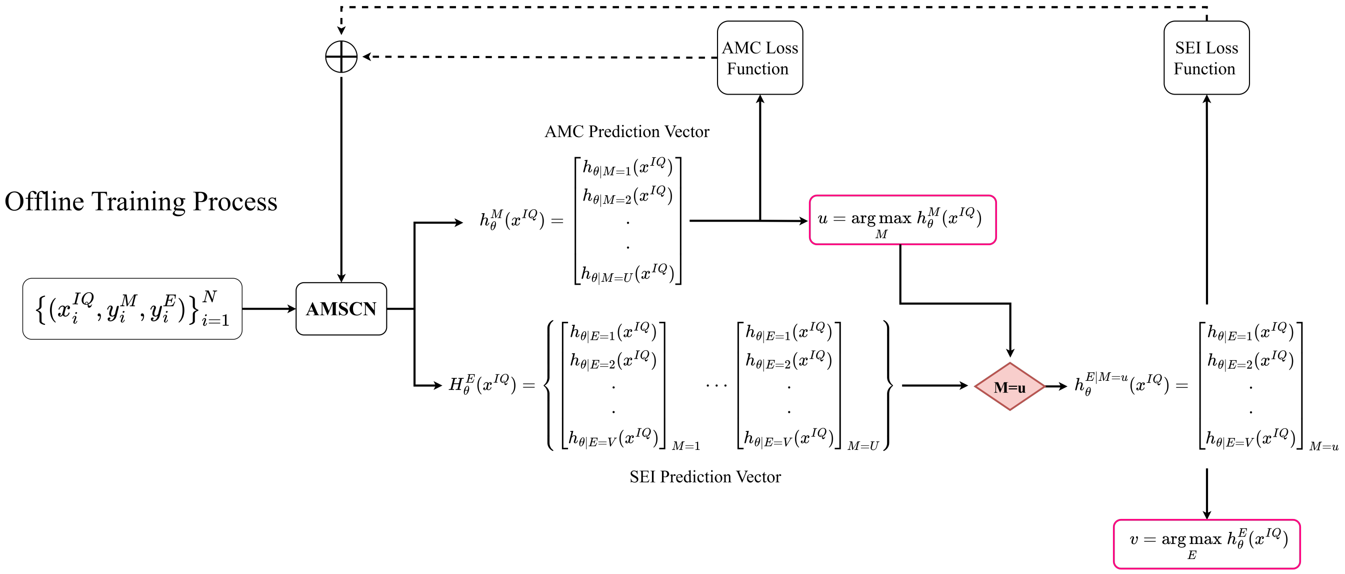
AMC Loss Function
SEI Loss Function
AMC Prediction Vector
Offline Training Process
$\{(x_i^{IQ}, y_i^M, y_i^E)\}_{i=1}^N$
AMSCN
$h_\theta^M(x^{IQ}) = \begin{bmatrix} h_{\theta|M=1}(x^{IQ}) \\ h_{\theta|M=2}(x^{IQ}) \\ \cdot \\ \cdot \\ h_{\theta|M=U}(x^{IQ}) \end{bmatrix}$
$u = \arg\max_M h_\theta^M(x^{IQ})$
$H_\theta^E(x^{IQ}) = \left\{ \begin{bmatrix} h_{\theta|E=1}(x^{IQ}) \\ h_{\theta|E=2}(x^{IQ}) \\ \cdot \\ \cdot \\ h_{\theta|E=V}(x^{IQ}) \end{bmatrix}_{M=1} \cdots \begin{bmatrix} h_{\theta|E=1}(x^{IQ}) \\ h_{\theta|E=2}(x^{IQ}) \\ \cdot \\ \cdot \\ h_{\theta|E=V}(x^{IQ}) \end{bmatrix}_{M=U} \right\}$
M=u
$h_\theta^{E|M=u}(x^{IQ}) = \begin{bmatrix} h_{\theta|E=1}(x^{IQ}) \\ h_{\theta|E=2}(x^{IQ}) \\ \cdot \\ \cdot \\ h_{\theta|E=V}(x^{IQ}) \end{bmatrix}_{M=u}$
SEI Prediction Vector
$v = \arg\max_E h_\theta^E(x^{IQ})$

Figure 3. Offline training process of the proposed AMSCN.

As shown in Figure 1, in the data collection phase, we needed to choose each emitter in turn to send signals under each modulation type at the transmitter side. We denoted the received signal by x^{IQ} and assumed that there were U modulation formats and V different transmitters; then, there would be $U \times V$ kinds of received-signal sets. Each set of the same kind was further split into multiple signal segments of length L, and the entire dataset, which contained N samples, can be represented as

$$(\mathbf{\Omega}_{x^{IQ}}, \mathbf{y^M}, \mathbf{y^E}) = \left\{ \left(\mathbf{x}_1^{IQ}, \mathbf{y}_1^M, \mathbf{y}_1^E\right), \cdots, \left(\mathbf{x}_i^{IQ}, \mathbf{y}_i^M, \mathbf{y}_i^E\right), \cdots, \left(\mathbf{x}_N^{IQ}, \mathbf{y}_N^M, \mathbf{y}_N^E\right) \right\}, \tag{20}$$

where $\mathbf{\Omega}_{x^{IQ}}$ is the sample set, $\mathbf{y}$ is the label set, $\left(\mathbf{x}_i^{IQ}, \mathbf{y}_i^M, \mathbf{y}_i^E\right)$ is the ith (i = 1, 2, ..., N)-labeled sample in the entire dataset, and $\mathbf{y}_i^M$ and $\mathbf{y}_i^E$ are vectors encoded with one-hot to indicate the modulation and the transmitter to which this sample belongs.

We used the maximum likelihood estimation to find the best parameters $\mathbf{\Theta}$ in the AMSCN. Since our framework contained two classification tasks, we let the model output the predicted probabilities under each classification task. For the AMC task, the probability $\mathbf{h}_\theta^M(x^{IQ})$ vector can be expressed as

$$\mathbf{h}_\theta^M(x^{IQ}) = \left[h_{\theta|M=i}(x^{IQ})\right], \ i = 1, 2, ..., U, \tag{21}$$

with

$$\sum_{i=1}^{U} h_{\theta|M=i}(x^{IQ}) = 1, \tag{22}$$

where M denotes the AMC task, and $h_{\theta|M=i}(x^{IQ})$ is the probability that the current signal belongs to the ith modulation type. In this vector, the subscript of the largest probability is the subscript of the modulation method to which the current sample belongs, which is given by

$$u = \arg\max_{i} h_{\theta|M=i}(x^{IQ}), \tag{23}$$

where u indicates the modulation format of the current signal. For the SEI task, the probability vector was generated according to the type of modulation predicted by the current model. In fact, the classifier of the SEI task initially output a set of U vectors.

$$\mathbf{H}_{\theta}^{E}(x^{IQ}) = \left\{\mathbf{h}_{\theta}^{E|M=u}\right\}, \quad u = 1, 2, ..., U, \tag{24}$$

where $\mathbf{H}$ is the ensemble of U sets of vectors, and $\mathbf{h}_{\theta}^{E|M=u}$ is the representation of the probability that the signal belongs to each transmitter under a modulation assumed to be u. We selected one of these U-group vectors based on the results of the AMC classifier in the previous step, and used this vector as a distribution probability to characterize the source of the device to which the current signal belongs.

The goal of the AMSCN model is to optimize the two tasks simultaneously. In this process, the classification process of the AMC was relatively independent, while the classification process of the SEI needed the prediction results of the AMC to improve this task. The overall loss of the AMSCN was derived by summing the cross-entropy losses of the two tasks, namely

$$L(\theta) = L^{M}(\theta) + L^{E}(\theta), \tag{25}$$

with

$$L^{M}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{u=1}^{U} y_i^{M} \log\Big(h_{\theta|M=u}(x^{IQ})\Big) \tag{26}$$

$$L^{E}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{v=1}^{V} y_i^{E} \log\Big(h_{\theta|E=v}(x^{IQ})\Big), \tag{27}$$

where $L^{M}(\theta)$ and $L^{E}(\theta)$ denoted the cross-entropy loss of the AMC and SEI, respectively. In the case of using the MDHC, we could achieve a good training effect without balancing the weight between $L^{M}(\theta)$ and $L^{E}(\theta)$.

4.2. Details of the AMSCN

As is shown in Figure 4, the AMSCN consisted of two modules, i.e., a shared-backbone module and an MDHC. The shared-backbone module was responsible for providing the distinguishable features for the following AMC and SEI tasks. To achieve better multitask classification, we subsequently used two different heads in the MDHC to map the common features into two task-specific features. Finally, the softmax function was used to convert these features into their respective class probability distributions, and the subscript of the maximum probability was used as the output of this classification task.

The shared-backbone module was responsible for extracting the common features needed for both tasks, and was subdivided into a DenseNet part and a Transformer part. The DenseNet part consisted of a cascade of several convolutional units of the same configuration, each of which was computed in the order of BatchNorm, ReLU activation, and one-dimensional convolutional operations. The more convolution units, the richer the nonlinear transformation of the signal; however, this increased the computational complexity. Unlike the application of DenseNet in image recognition, the data length of the signal was shorter than that of the image. In the SEI task, the max-pool operation corrupts minor characteristics in the signal; therefore, we did not use the transition layer as in [55]. In the DenseNet part of the AMSCN, the length of the input signal and the length of the output features were the same, which had the benefit of allowing shallow features to be directly cascaded with deep features by channel, thus increasing the scale diversity of the CNN output. The detailed parameter settings are shown in Table 1; the feature dimensions are represented using $B \times C \times L$, where B denotes the batch size, C denotes the channel size, and L denotes the feature length.

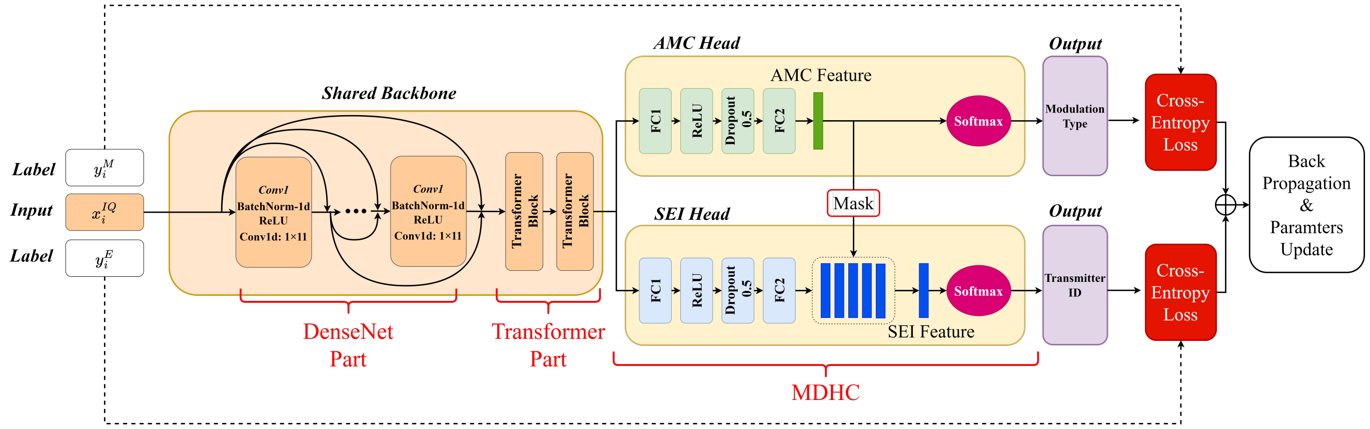

Figure 4. Schematic diagram of forward and backward propagation of AMSCN.

Since the modulation type and transmitter hardware impairments affect the waveform of the signal at different scales, a large convolution kernel 1 × 11 was used in our base convolution unit instead of the traditional small convolution kernel to increase the network's ability to capture features at different scales. In addition, since we did not use pooling operations to reduce the feature dimension, the computational effort grows exponentially with more convolutional layers, so we fixed the number of channels output from each convolutional layer to 16 and used cross-channel cascading to increase the channel richness of the final output features. The number of feature channels output from the DenseNet part is equal to the sum of the output channels of each convolution unit and the original 2 channels of the signal, i.e., 162.

Table 1. The hyperparameters of the DenseNet part of the AMSCN.

Input: x^{IQ} (Dimension: $B \times 2 \times L$)				
Layers	**Kernel Size**	**Padding**	**Stride Step**	**Output Feature Dimension**
Conv1	16@(1×11)	5	1	$B \times (16 + 2) \times L$
Conv2	16@(1×11)	5	1	$B \times (16 \times 2 + 2) \times L$
Conv3	16@(1×11)	5	1	$B \times (16 \times 3 + 2) \times L$
...	...	...	...	...
Conv10	16@(1×11)	5	1	$B \times (16 \times 10 + 2) \times L$
Output: Feature Matrix (Dimension: $B \times 162 \times L$)				

After the local features were extracted by the convolution module, we used the Transformer structure to improve the relevance of the information in the features with different time spans. Since the features were not shortened in the DenseNet part, the Transformer structure was well suited for processing such long span features. In the AMSCN, the Transformer module mainly consisted of two transformer blocks, and the complete structure is shown in Figure 5.

In the figure above, we used $\mathbf{X}$ to denote the multichannel features output by the convolution module. Since the features on each channel had the same length, we grouped the features with the same sequence position on each channel and named them as a node. $\mathbf{X}$ is a sequence of nodes of length L, and the vector length of each node is C. Since the Transformer does not consider the relationship order between nodes during the computation, we needed to encode the position relationship of the nodes into the input sequence $\mathbf{X}$ before the first transformer layer. In the AMSCN, we set a learnable parameter sequence with the same dimension as $\mathbf{X}$, i.e., position embedding, and added it directly to

X to obtain the encoded feature sequence. Before the position embedding, we needed to add an additional node to **X**. This node was separated for the classification vector after the two-layer transformer computation was completed.

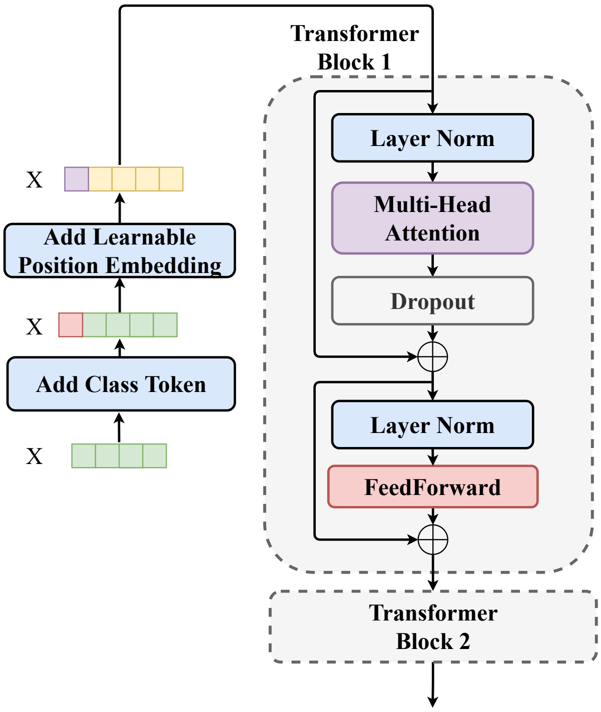

Figure 5. Schematic diagram of the Transformer part of the AMSCN. Before the features are fed into the first transformer block, preprocessing is required, i.e., adding the classification token and position embedding.

A transformer block consists of two main components: self-attention and feed-forward. The outputs of both components were connected using residuals to enhance the backpropagation of the gradient. The self-attention part was the core of the transformer block, whose structure is shown in Figure 6.

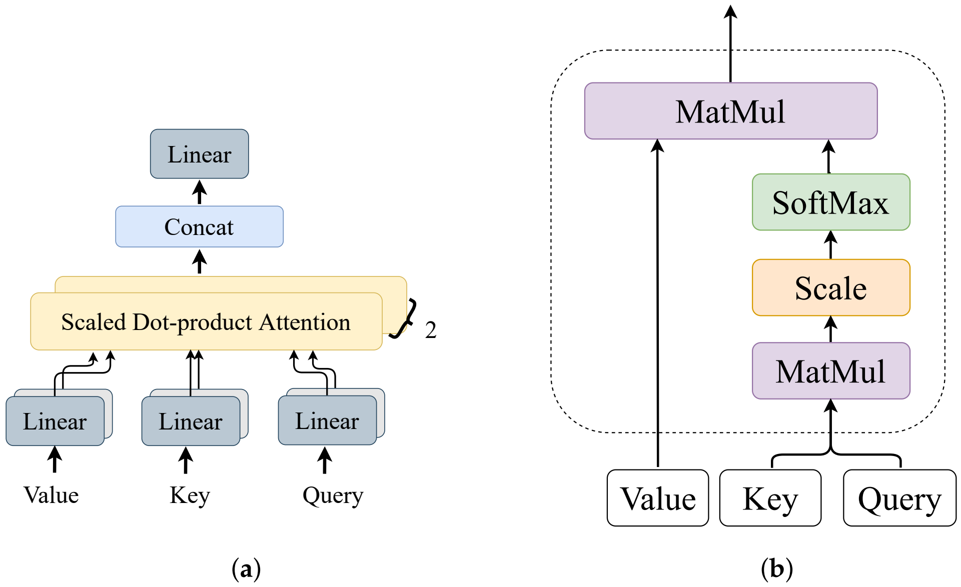

Figure 6. The structure of the self-attention part a transformer block. (**a**) Details of the multi-head attention structure. (**b**) Scaled dot-product attention structure.

A multi-head attention module can be seen as a composition of multiple single-head scaled dot-product attention modules. The calculation of scaled dot-product attention can be expressed as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{QK}^T}{\sqrt{d_k}})\mathbf{V}, \tag{28}$$

where $\mathbf{Q}$ is a matrix representation of the query vector, $\mathbf{K}$ is a matrix representation of the key vector, $\mathbf{V}$ is a matrix representation of the value vector, and d_k is the dimension of the input feature, which can be used to scale the $\mathbf{QK}^T$; $softmax(\cdot)$ is the activation function to ensure that the output vector value is between 0 and 1. The calculation of the multi-head attention can be expressed as

$$\text{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{W}^O Concat(\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_m) \tag{29}$$

$$\mathbf{h}_i = \text{Attention}(\mathbf{W}_i^Q \mathbf{Q}, \mathbf{W}_i^K \mathbf{K}, \mathbf{W}_i^V \mathbf{V}), \tag{30}$$

where $\mathbf{W}^O$ is the linear transformation matrix of the multi-head attention result, $\mathbf{h}_i$ is the vector of a dot-product attention calculation, and m is the number of multi-heads. $\mathbf{W}_i^Q$ is the linear transformation matrix of the input query matrix $\mathbf{Q}$, and $\mathbf{W}_i^K$ and $\mathbf{W}_i^V$ have a similar meaning. In the AMSCN, there were two heads in each multi-head attention module. The feed-forward was a two-layer fully connected layer; the first layer contained the ReLU activation function and had 1024 neuron nodes, and the second layer did not use the activation function, which can be expressed by the following equation

$$\text{FF} = \mathbf{W_2}\text{Max}(0, \mathbf{W_1 X} + \mathbf{b_1}) + \mathbf{b_2}, \tag{31}$$

where $\text{Max}(\cdot)$ denotes the ReLU activation, $\mathbf{X}$ is the input to the feed-forward module, $\mathbf{W}_1$ and $\mathbf{W}_2$ are the linear transformation matrices of the first and second layers, respectively, and $b1$ and $b2$ are the biases of each layer.

The second transformer block, shown in Figure 5, had exactly the same parameters as the first transformer block; the stacked transformer blocks can achieve a more complex synthesis of temporal information. We took out the first node of the sequence output by the second transformer block, which was the classification token, as the semantic feature of the shared-backbone network output. A more specific hyperparameter setting is shown in Table 2.

To implement a network model that performs two classification tasks, we connected the MDHC to the shared-backbone network. The structure of the two classifier heads in the MDHC was basically the same, except for the setting of $FC2$. For each classifier head, the number of the $FC1$ layer's hidden neurons was 256. The length of the feature vector output from the AMC header was equal to the number of modulation types U, while the length of the feature vector output from the SEI header was equal to $U \times V$. We reshaped this feature vector into a U-row and V-column feature matrix $\mathbf{P}$ and generated the mask matrix $\mathbf{Q}$ based on the AMC feature vector. The dimension of the mask matrix $\mathbf{Q}$ was $U \times V$. We note that the subscript of the maximum value in the AMC eigenvector was k. Subsequently, the kth row of the mask matrix was set to 1, and the remaining rows were all set to 0. Finally, we summed the matrix $\mathbf{P} \cdot \mathbf{Q}$ by columns to obtain the final SEI feature vector. Figure 7 gives a schematic representation of the calculation process when $U = 3$ and $V = 3$.

Generally speaking, each classifier of multi-task learning is independent of each other; however, in this paper, we need the decision results of the AMC task to have an impact on the decision process of SEI. In MDHC, we can consider that there are multiple SEI classifiers existing simultaneously, each of which predicts a classification result. We picked one of the results based on the prediction of AMC. If the prediction of the AMC task fails, then to a large extent, the SEI results will also be unreliable. At this point, the loss value of the network rises, thus forcing the model to first learn the modulation features and then use them to assist in adjudicating the SEI task.

Table 2. The hyperparameters of the two Transformer blocks.

Input: x^{IQ} (Dimension: $B \times 162 \times L$)		
Layers	**Description**	**Output Feature Dimension**
Reshape	Change the order of data dimensions	$B \times L \times 162$
Add Class Token	Increase sequence length	$B \times (L+1) \times 162$
Add Position Embedding	No change in data dimension	$B \times (L+1) \times 162$
Layer Norm1	Channel-by-channel normalization	$B \times (L+1) \times 162$
Multi-Head	Fully connected structure, 2 MHA groups, output merging	$B \times (L+1) \times 162$
Dropout	Drop rate 0.4	$B \times (L+1) \times 162$
Residual connection	Add operation	$B \times (L+1) \times 162$
Layer Norm2	Channel-by-channel normalization	$B \times (L+1) \times 162$
FeedForward	Two layers fully connected, hidden cell 128, drop rate 0.4	$B \times (L+1) \times 162$
Residual connection	Add operation	$B \times (L+1) \times 162$
The second transformer block	Same configuration as the first transformer block	$B \times (L+1) \times 162$
Extracting class token	The first vector in the sequence	$B \times 1 \times 162$
Output: Feature Matrix (Dimension: $B \times 1 \times 162$)		

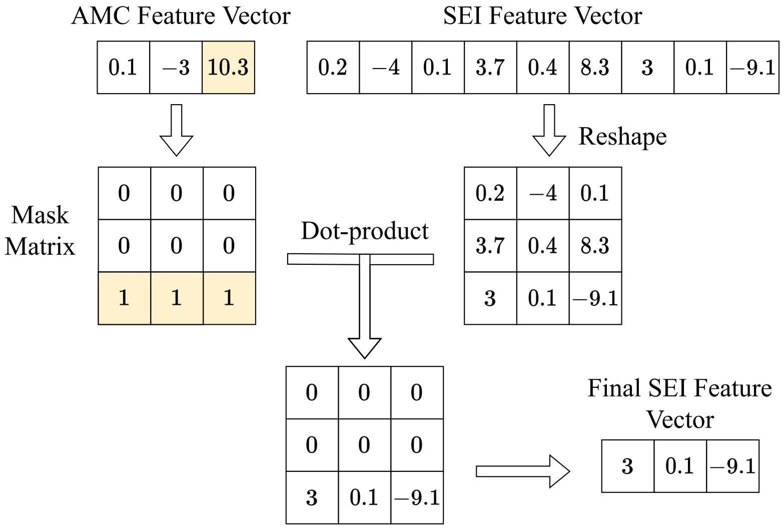

Figure 7. The calculation process of the SEI feature vectors. The mask matrix is generated from the AMC feature vector. The subscript of the largest element of the AMC feature vector determines which row of the mask matrix is 1. The colored part of this figure indicates the maximum value in the vector and the row of the matrix that takes 1.

5. Results and Analysis

5.1. Dataset Generation and Training

Since there was no publicly available dataset suitable for the problem at hand, we generated a simulation dataset and verified the effectiveness of the AMSCN on this dataset. We selected five types of digital modulations, which were BPSK, QPSK, 8PSK, 16QAM, and 32QAM. At the same time, we also set the parameters of the five transmitters with different amplitude imbalance, phase imbalance, and nonlinear power amplifier behaviors. The parameter settings are shown in Table 3.

Table 3. Parameter settings for the five different transmitters.

Device	Amplitude Imbalance α(dB)	Phase Imbalance β(°)	PA Model	Parameters	Limits
Device1	-0.5	-10	Saleh [51,52]	$\alpha_\alpha = 1.2$, $\beta_\alpha = 0.36$, $\alpha_\phi = 0.374$, $\beta_\phi = 0.36$	$\mathbb{E}\ \{\lvert r(n)^2\rvert\} = 0.5$, $\max(\lvert r(n)\rvert) \le 1$
Device2	-0.3	-6	Rapp [53]	$V_{sat} = 2$, $p = 1$	$\mathbb{E}\{\lvert r(n)^2\rvert\} = 0.6$, $\max(\lvert r(n)\rvert) \le 1$
Device3	-0.1	-2	Saleh [51,52]	$\alpha_\alpha = 1.9638$, $\beta_\alpha = 0.9945$, $\alpha_\phi = 2.5293$, $\beta_\phi = 2.8168$	$\mathbb{E}\{\lvert r(n)^2\rvert\} = 0.5$, $\max(\lvert r(n)\rvert) \le 1$
Device4	0.1	2	CMOS [54]	$V_{sat} = 0.81$, $p = 0.58$, $d = 44.68$, $e = 0.114$, $f = 2.4$, $g = 2.3$	$\mathbb{E}\{\lvert r(n)^2\rvert\} = 1.162$, $\max(\lvert r(n)\rvert) \le 1$
Device5	0.3	6	Saleh [51,52]	$\alpha_\alpha = 2.1587$, $\beta_\alpha = 1.1517$, $\alpha_\phi = 4.0033$, $\beta_\phi = 9.1040$	$\mathbb{E}\{\lvert r(n)^2\rvert\} = 0.5$, $\max(\lvert r(n)\rvert) \le 1$

The symbols in this table are consistent with Equations (3)–(18). The term *Limits* means there were some restrictions on the signal before it entered the PA model. $\mathbb{E}\{|r(n)^2|\}$ indicates the average power of the input signal to be satisfied, and $max(|r(n)|)$ means that the amplitude of each IQ point of the input signal was limited to 1. We found the values of these amplifier parameters from the literature mentioned in Section 3.2, which were derived from measurements and fits to real amplifier devices.

We generated the baseband signal according to the process shown in Figure 2, with a roll-off factor of 0.25 for the raised cosine filter at the transmitter and a signal-to-noise ratio (SNR) from -20 to 20 dB in steps of 4 dB. At the receiver side, the number of points sampled per symbol was 8, and 625 signal segments were acquired for each combination of modulation, device, and signal-to-noise ratio, with each segment having a length of 256 samples. Thus, the total number of signal segments received in the dataset was $5 \times 5 \times 11 \times 625$ (5 transmitters, 5 modulations, 11 SNRs, and 625 segments for each condition). We randomly shuffled the above dataset and sliced it into training and testing sets in the ratio of 6:4. In the offline training phase, the Adam optimizer [56] was used to search for the model parameters, and the training process was terminated when the average classification accuracy on the testing set no longer increased.

5.2. The Effect of the Mask between the Two Heads

The initial idea of this paper was to fuse the AMC task and the SEI task within one single model to save network parameters and improve the computational efficiency. Therefore, we let the two tasks share the same backbone network, and let the two classifier heads be responsible for their respective prediction tasks. In general, different modulation methods have a greater impact on the signal waveform compared with different device hardware impairments. The design methodology of the MDHC was based on this assumption. When the two tasks occur at the same time, the change in modulation type may interfere with the extraction of the transmitter features from the signal. Therefore, we let the SEI head predict five sets of results simultaneously, and each set could be viewed as the conditional probability of the five transmitters with a known modulation method. We selected the appropriate one from the five sets of results output from the SEI head based on the predicted results from the AMC head. This process was similar to the AMC head, generating a mask that covered the output results of the SEI head. Figure 8 compares the effect of using and not using the mask operation.

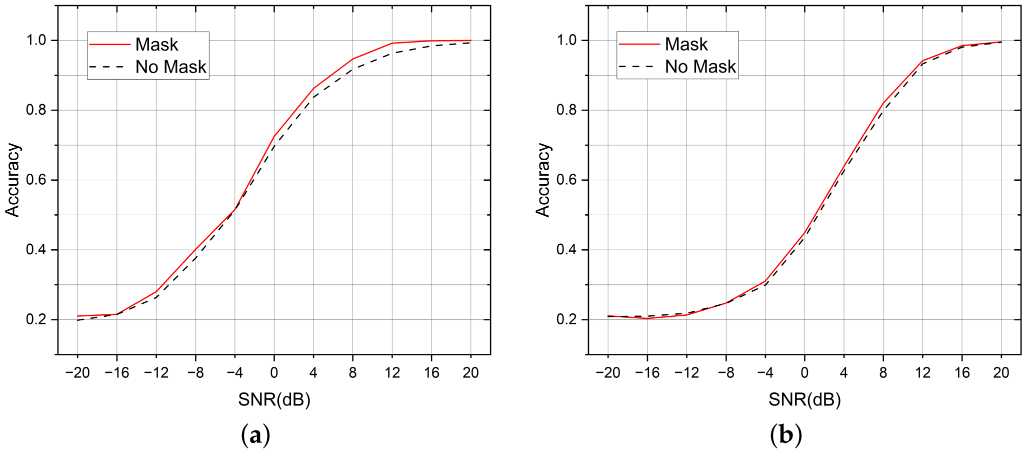

Figure 8. Adding a mask between the two task headers improves the performance on both tasks. (**a**) Comparison of the mask effects in the AMC tasks. (**b**) Comparison of the mask effects in the SEI tasks.

We used the average accuracy to measure the difference in performance between the different methods. For a given task, the average accuracy is the average probability of being able to classify all the samples of the test set correctly, and the average accuracy can be seen as the average of the accuracies under each SNR in the graph. In the AMC task, the average accuracy of using the mask method was 0.65, and it was 0.6327 without the mask; in the SEI task, the average accuracy of using the mask was 0.5472, and it was 0.5413 without the mask.

5.3. The Effect of the Fusion of the Two Tasks

In order to demonstrate that the fusion of two tasks in the AMSCN could ensure the optimal result of both tasks, the following comparative experiments were conducted. We trained a single-head model for the AMC task and a single-head model for the SEI task. The structure of the single-head model was similar to that of the AMSCN, except that each single-head model used one classifier head to predict the current task without considering the possible information gain from the other task. In Figure 9, we use the term *Fusion* to refer to our AMSCN model,and *Single* to refer to the model with one task head. In the AMC task, the average accuracy of the fusion method was 0.65, and it was 0.651 for the single method; in the SEI task, the average accuracy of fusion method was 0.5472, and it was 0.5142 for the single method.

In Figure 9a, the accuracy curve of the single-task model was slightly higher at high SNRs than that of the dual-task model, which we believe is mainly caused by setting the average accuracy as the optimization target. During model training, the optimization weights were the same for each SNR, but the accuracy curve may fluctuate in the range of high and low SNRs when the average accuracy is similar. Specifically, in the early stage of model optimization, the accuracy under low SNR and high SNR would increase simultaneously, while at the end of model convergence, the accuracy under high SNR would decrease if the model gradually advances toward improving the accuracy under low SNR. This observation may be caused by the conflicting judgment methods between the signal feature vectors of various modulation types at low SNRs and those at high SNRs, which makes it difficult for the model to improve the accuracy at each SNR simultaneously. In addition, the structure of the AMSCN showed that the model was not affected by the predicted results of SEI under the AMC task, and thus the dual task maintained an essentially similar performance to the single task. The reason for our design is that we argue the judgment difficulty of the AMC task is lower than that of the SEI, i.e., the hardware impairments of the transmitter have less impact on the signal characteristics than the modulation type, and thus the detection results of the AMC task do not suffer from the SEI task. Therefore, the advantage of our model is not to improve the performance of AMC task, instead, we exploit the strength of multi-task learning to improve the performance of SEI tasks while maintaining the performance of AMC.

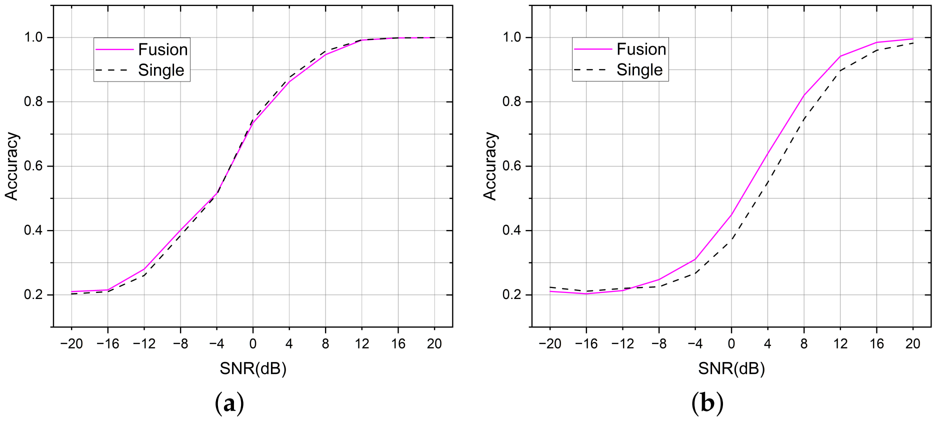

Figure 9. For the AMSCN models, multitask training is more effective than single-task training. (**a**) Comparison of the fusion method and the single-head method in the AMC tasks. (**b**) Comparison of the fusion method and the single-head method in the SEI tasks.

5.4. The Performance Comparison among Different Methods

We selected eight popular signal classification models to compare with our AMSCN model. Some of these models have achieved state-of-the-art results with publicly available AMC or SEI datasets [57,58]. The ResNet12 is a simplified version of ResNet18 [59], and the DenseNet-backbone is the same as the DenseNet part of the AMSCN. All models were trained and tested in a single-task classification manner only. However, since the length of our sample data is 256, which is different from the length of the data on which these models were designed, the training results tend to be poor when the hyperparameters are set according to the original version; therefore, we fine-tuned the hyperparameters of these models. These adjustments included changing the original classifier hidden layer unit of 128 to 512, and the original LSTM structure's hidden unit of 128 to 256 or 512. After the adjustments, the prediction accuracy results are generally better than the original parameters.

Figure 10 gives a comparison of the SNR performance curves between the models, and Table 4 lists the detailed average accuracy of each model. On the AMC task, the AMSCN remained at the same level as other models, and we had the best results in terms of average accuracy; however, in terms of each SNR, they were not optimal. On the SEI task, we were ahead of other models and achieved the best results at almost all SNRs. As can be seen from Table 4, a model that performs better on one task may perform worse or not converge on the other task, which indicates that there is variability, such as differences in feature scales, between the two tasks. For the AMSCN, there are two design considerations that enable our model to be compatible with both tasks: the design of a more compatible backbone network and the design of a mechanism (MDHC) for transferring relevant information between tasks.

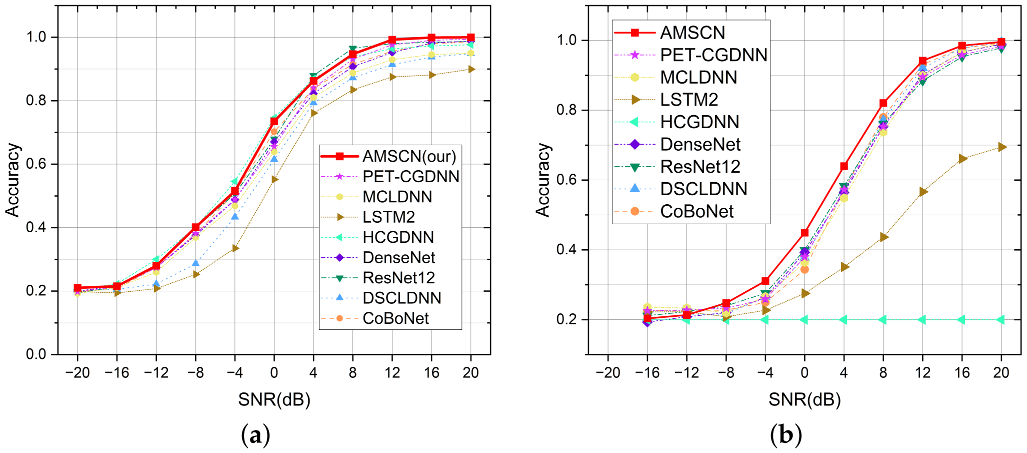

Figure 10. Performance comparison between the AMSCN and other popular models. (**a**) Model performance curves in the AMC tasks. (**b**) Model performance curves in the SEI tasks.

Table 4. The average accuracy comparison between the AMSCN and other popular models.

Model	AMC	SEI
AMSCN (ours)	0.650	0.547
HCGDNN [60]	0.649	0.2
ResNet12	0.642	0.519
PET-CGDNN [61]	0.631	0.520
CoBoNet [58]	0.631	0.522
DenseNet (backbone)	0.625	0.513
MCLDNN [62]	0.606	0.518
DSCLDNN [63]	0.584	0.521
LSTM2 [64]	0.545	0.372

By taking advantage of multi-tasking, we can reduce the computation of both models into one model, thus improving the detection efficiency of the tasks. Table 5 gives the number of parameters, the amount of floating point calculations, and the single-inference time for each model. These tests were performed on the same hardware and software environment, including Pytorch 1.12.1, Intet(R) Xeon CPU E5-2620, and one GeForce RTX 2080Ti. As can be seen from Table 5, although AMSCN is not the smallest in terms of number of parameters and computational effort, it benefits from the better parallel performance of the convolution and Transformer structures in the network structure, resulting in a shorter single-inference time compared with the model using the LSTM structure.

It should be emphasised that the computational complexity of the other models is measured under a single task, whereas the AMSCN is under a dual task; therefore, the computational complexity of all the other models should be doubled for a system containing two tasks. If we consider the single processing time as an important indicator of complexity, we can say that the dual-task-based AMSCN approach achieves the best results in terms of the combination of accuracy and efficiency metrics.

Our model has the advantage of good parallelism, thus maintaining a high computational speed in high-capacity backbone networks; however, the storage overhead is high and is not suitable for cost-sensitive computing platforms. Therefore, compressing the size of the model to further increase the computational speed while keeping the performance constant will be our next research goal.

Table 5. Comparison of computational efficiency between models. The metric "Latency" stands for the inference time for a single signal sequence. If the two tasks are considered together, the weight and the calculated amount of all other models should be doubled.

Model	Weight	Flops	Latency
AMSCN (ours)	1.57 M	259.34 M	0.007 s
HCGDNN	0.457 M	92.53 M	0.012 s
ResNet12	13.85 M	1549.2 M	0.006 s
PET-CGDNN	0.841 M	206.3 M	0.009 s
CoBoNet	0.67 M	135.3 M	0.035 s
DenseNet(backbone)	0.174 M	33.9 M	0.004 s
MCLDNN	3.53 M	872.6 M	0.007 s
DSCLDNN	1.13 M	293.2 M	0.014 s
LSTM2	3.16 M	811.08 M	0.006 s

6. Conclusions

In this paper, we presented the problem of possible efficiency loss in conventional RF signal-monitoring systems, where the same signal is passed through multiple independent detection systems for achieving multiple related detection targets. Specifically, in the scenario where both AMC and SEI tasks are required, we proposed a novel multitask classification method, named an AMSCN, to obtain gains in detection accuracy or computational efficiency. The AMSCN learned the common features of two tasks simultaneously through a high-capacity backbone network, and then learned the unique features of each task separately through the mask-based dual-head classifier. Simulation experimental results showed that AMC and SEI are task-dependent, and the dual-task model can take full advantage of this correlation and improve the detection accuracy of SEI. On the AMC task, the detection accuracy of the AMSCN remained at the same level as the state-of-the-art model, while on the SEI task, there was a 2.5% improvement compared with the state-of-the-art model. In addition, the computational structure of the AMSCN was the same under both tasks, which can simplify the training and deployment of monitoring systems under multiple tasks. In conclusion, we believe that multi-task learning has great potential in the field of signal monitoring, and the more comprehensive the information obtained by the model during training, the more accurate the recognition of the signal will be. In the future, we will further explore the applicability of the AMSCN method, improving its robustness under different channels, detecting more modulations, and fusing more detection targets in the same model to improve efficiency.

Author Contributions: Writing—original draft preparation, S.Y.; Writing—review and editing, S.Y. and S.H.; Hardware modeling, S.Y. and J.H.; Model design and performance validation, S.Y. and S.C.; Funding acquisition, Z.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant (62171045, 62201090), and in part by the National Key Research and Development Program of China under Grants (2020YFB1807602, 2019YFB1804404).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset in the paper can be accessed from the following link https://github.com/WTI-Cyber-Team/Public_Wireless_Signal_Datasets (accessed on 5 January 2023).

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

References

1. Rembovsky, A.M.; Ashikhmin, A.V.; Kozmin, V.A.; Smolskiy, S.M. *Radio Monitoring: Automated Systems and Their Components*; Springer: Berlin/Heidelberg, Germany, 2018.
2. Weber, C.; Peter, M.; Felhauer, T. Automatic modulation classification technique for radio monitoring. *Electron. Lett.* **2015**, *51*, 794–796. [CrossRef]
3. Lu, Q.; Yang, J.; Jin, Z.; Chen, D.; Huang, M. State of the art and challenges of radio spectrum monitoring in China. *Radio Sci.* **2017**, *52*, 1261–1267. [CrossRef]
4. Podstrigaev, A.S.; Smolyakov, A.V.; Davydov, V.V.; Myazin, N.S.; Grebenikova, N.M.; Davydov, R.V. New method for determining the probability of signals overlapping for the estimation of the stability of the radio monitoring systems in a complex signal environment. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 525–533.
5. Ylianttila, M.; Kantola, R.; Gurtov, A.; Mucchi, L.; Oppermann, I.; Yan, Z.; Nguyen, T.H.; Liu, F.; Hewa, T.; Liyanage, M.; et al. 6g white paper: Research challenges for trust, security and privacy. *arXiv* **2020**, arXiv:2004.11665.
6. Xing, C.; Zhou, Y.; Peng, Y.; Hao, J.; Li, S. Specific Emitter Identification Based on Ensemble Neural Network and Signal Graph. *Appl. Sci.* **2022**, *12*, 5496. [CrossRef]
7. Huang, K.; Yang, J.; Liu, H.; Hu, P. Channel-Robust Specific Emitter Identification Based on Transformer. *Highlights Sci. Eng. Technol.* **2022**, *7*, 71–76. [CrossRef]
8. Tang, P.; Xu, Y.; Wei, G.; Yang, Y.; Yue, C. Specific emitter identification for IoT devices based on deep residual shrinkage networks. *China Commun.* **2021**, *18*, 81–93. [CrossRef]
9. Rajendran, S.; Sun, Z. RF Impairment Model-Based IoT Physical-Layer Identification for Enhanced Domain Generalization. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 1285–1299. [CrossRef]
10. O'Shea, T.J.; Pemula, L.; Batra, D.; Clancy, T.C. Radio transformer networks: Attention models for learning to synchronize in wireless systems. In Proceedings of the 2016 50th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 6–9 November 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 662–666.
11. Huynh-The, T.; Hua, C.H.; Pham, Q.V.; Kim, D.S. MCNet: An efficient CNN architecture for robust automatic modulation classification. *IEEE Commun. Lett.* **2020**, *24*, 811–815. [CrossRef]
12. West, N.E.; O'shea, T. 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: Piscataway, NJ, USA, 2017; pp. 1–6.
13. Shariati, M.; Davoodnabi, S.M.; Toghroli, A.; Kong, Z.; Shariati, A. Hybridization of metaheuristic algorithms with adaptive neuro-fuzzy inference system to predict load-slip behavior of angle shear connectors at elevated temperatures. *Compos. Struct.* **2021**, *278*, 114524. [CrossRef]
14. Sood, A.; Sharma, R.K.; Bhardwaj, A.K. Artificial intelligence research in agriculture: A review. *Online Inf. Rev.* **2022**, *46*, 1054–1075. [CrossRef]
15. Apell, P.; Eriksson, H. Artificial intelligence (AI) healthcare technology innovations: the current state and challenges from a life science industry perspective. *Technol. Anal. Strateg. Manag.* **2023**, *35*, 179–193. [CrossRef]
16. Chen, X.; Zou, D.; Xie, H.; Cheng, G.; Liu, C. Two decades of artificial intelligence in education. *Educ. Technol. Soc.* **2022**, *25*, 28–47.
17. Meng, F.; Chen, P.; Wu, L.; Wang, X. Automatic modulation classification: A deep learning enabled approach. *IEEE Trans. Veh. Technol.* **2018**, *67*, 10760–10772. [CrossRef]
18. Zhou, Y.; Lin, T.; Zhu, Y. Automatic modulation classification in time-varying channels based on deep learning. *IEEE Access* **2020**, *8*, 197508–197522. [CrossRef]
19. Wang, Y.; Wang, J.; Zhang, W.; Yang, J.; Gui, G. Deep learning-based cooperative automatic modulation classification method for MIMO systems. *IEEE Trans. Veh. Technol.* **2020**, *69*, 4575–4579. [CrossRef]
20. Wang, Y.; Yang, J.; Liu, M.; Gui, G. LightAMC: Lightweight automatic modulation classification via deep learning and compressive sensing. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3491–3495. [CrossRef]
21. Mendis, G.J.; Wei, J.; Madanayake, A. Deep learning-based automated modulation classification for cognitive radio. In Proceedings of the 2016 IEEE International Conference on Communication Systems (ICCS), Shenzhen, China, 14–16 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6.
22. Peng, L.; Zhang, J.; Liu, M.; Hu, A. Deep learning based RF fingerprint identification using differential constellation trace figure. *IEEE Trans. Veh. Technol.* **2019**, *69*, 1091–1095. [CrossRef]
23. Yang, T.; Zhao, J.; Wang, X.; Xu, F. Deep learning based RFF recognition with differential constellation trace figure towards closed and open set. In Proceedings of the 2022 IEEE/CIC International Conference on Communications in China (ICCC), Foshan, China, 11–13 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 908–913.
24. Huang, K.; Liu, H.; Hu, P. Deep Learning of Radio Frequency Fingerprints from Limited Samples by Masked Autoencoding. In *IEEE Wireless Communications Letters*; IEEE: Piscataway, NJ, USA, 2022.
25. Yu, J.; Hu, A.; Zhou, F.; Xing, Y.; Yu, Y.; Li, G.; Peng, L. Radio frequency fingerprint identification based on denoising autoencoders. In Proceedings of the 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Barcelona, Spain, 21–23 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.

26. Li, P. Research on radar signal recognition based on automatic machine learning. *Neural Comput. Appl.* **2020**, *32*, 1959–1969. [CrossRef]
27. Wen, X.; Cao, C.; Sun, Y.; Li, Y.; Peng, H.; Wang, M. RF Transmitter Identification and Classification Based on Deep Residual Shrinkage Network. In Proceedings of the 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Haikou, China, 20–22 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 327–335.
28. Liao, K.; Zhao, Y.; Gu, J.; Zhang, Y.; Zhong, Y. Sequential convolutional recurrent neural networks for fast automatic modulation classification. *IEEE Access* **2021**, *9*, 27182–27188. [CrossRef]
29. O'Shea, T.J.; Roy, T.; Clancy, T.C. Over-the-air deep learning based radio signal classification. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 168–179. [CrossRef]
30. Zhu, M.; Feng, Z.; Zhou, X. A novel data-driven specific emitter identification feature based on machine cognition. *Electronics* **2020**, *9*, 1308. [CrossRef]
31. Madhu, A.; Prajeesha, P.; Kulkarni, A.S. Radar Emitter Identification using Signal Noise and Power Spectrum Analysis in Deep Learning. In Proceedings of the 2022 Fifth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU), Riyadh, Saudi Arabia, 28–29 March 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 52–57.
32. Guo, S.; Xu, Y.; Huang, W.; Liu, B. An SEI-Based Identification Scheme for Illegal FM Broadcast. In Proceedings of the 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Haikou, China, 20–22 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 517–524.
33. Wang, P.; Wang, J.; Wang, G. Specific Emitter Identification Method Based on I/Q Imbalance with SNR Estimation U sing Wavelet Denoising. In Proceedings of the 2021 IEEE 21st International Conference on Communication Technology (ICCT), Tianjin, China, 13–16 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1057–1062.
34. Ca^mara, T.V.; Lima, A.D.; Lima, B.M.; Fontes, A.I.; Martins, A.D.M.; Silveira, L.F. Automatic modulation classification architectures based on cyclostationary features in impulsive environments. *IEEE Access* **2019**, *7*, 138512–138527. [CrossRef]
35. Mao, Y.; Dong, Y.Y.; Sun, T.; Rao, X.; Dong, C.X. Attentive Siamese Networks for Automatic Modulation Classification Based on Multitiming Constellation Diagrams. In *IEEE Transactions on Neural Networks and Learning Systems*; IEEE: Piscataway, NJ, USA, 2021.
36. Tong, L.; Fang, M.; Xu, Y.; Peng, Z.; Zhu, W.; Li, K. Specific Emitter Identification Based on Multichannel Depth Feature Fusion. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9342085. [CrossRef]
37. Wan, T.; Ji, H.; Xiong, W.; Tang, B.; Fang, X.; Zhang, L. Deep learning-based specific emitter identification using integral bispectrum and the slice of ambiguity function. *Signal Image Video Process.* **2022**, *16*, 2009–2017. [CrossRef]
38. Ramjee, S.; Ju, S.; Yang, D.; Liu, X.; Gamal, A.E.; Eldar, Y.C. Fast deep learning for automatic modulation classification. *arXiv* **2019**, arXiv:1901.05850.
39. Lu, X.; Tao, M.; Fu, X.; Gui, G.; Ohtsuki, T.; Sari, H. Lightweight Network Design Based on ResNet Structure for Modulation Recognition. In Proceedings of the 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), Virtual, 27 September–28 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–5.
40. Shaik, S.; Kirthiga, S. Automatic Modulation Classification using DenseNet. In Proceedings of the 2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP), Chennai, India, 24–25 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 301–305.
41. Li, R.; Hu, J.; Li, S.; Ai, W. Specific Emitter Identification based on Multi-Domain Features Learning. In Proceedings of the 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID), Guangzhou, China, 28–30 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 178–183.
42. Jiang, K.; Qin, X.; Zhang, J.; Wang, A. Modulation Recognition of Communication Signal Based on Convolutional Neural Network. *Symmetry* **2021**, *13*, 2302. [CrossRef]
43. Liu, X. Automatic Modulation Classification Based on Improved R-Transformer. In Proceedings of the 2021 International Wireless Communications and Mobile Computing (IWCMC), Harbin City, China, 28 June–2 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8.
44. Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv* **2017**, arXiv:1706.05098.
45. Vafaeikia, P.; Namdar, K.; Khalvati, F. A brief review of deep multi-task learning and auxiliary task learning. *arXiv* **2020**, arXiv:2007.01126.
46. Sun, X.; Panda, R.; Feris, R.; Saenko, K. Adashare: Learning what to share for efficient deep multi-task learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 8728–8740.
47. Phillips, J.; Martinez, J.; Bârsan, I.A.; Casas, S.; Sadat, A.; Urtasun, R. Deep multi-task learning for joint localization, perception, and prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 4679–4689.
48. 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]
49. Doan, V.S.; Huynh-The, T.; Hoang, V.P.; Nguyen, D.T. MoDANet: Multi-Task Deep Network for Joint Automatic Modulation Classification and Direction of Arrival Estimation. *IEEE Commun. Lett.* **2021**, *26*, 335–339. [CrossRef]

50. Xu, H.; Xu, X. A transformer based approach for open set specific emitter identification. In Proceedings of the 2021 7th International Conference on Computer and Communications (ICCC), Chengdu, China, 10–13 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1420–1425.
51. Rapp, C. Effects of HPA-nonlinearity on a 4-DPSK/OFDM-signal for a digital sound broadcasting signal. *ESA Spec. Publl.* **1991**, *332*, 179–184.
52. Saleh, A.A. Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers. *IEEE Trans. Commun.* **1981**, *29*, 1715–1720. [CrossRef]
53. Jayati, A.E.; Sipan, M. Impact of nonlinear distortion with the rapp model on the gfdm system. In Proceedings of the 2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE), Surabaya, Indonesia, 3–4 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
54. Zhang, C.; Xiao, Z.; Gao, B.; Su, L.; Jin, D. Power amplifier non-linearity treatment with distorted constellation estimation and demodulation for 60 GHz single-carrier frequency-domain equalisation transmission. *IET Commun.* **2014**, *8*, 278–286. [CrossRef]
55. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
56. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
57. Zhang, F.; Luo, C.; Xu, J.; Luo, Y.; Zheng, F. Deep learning based automatic modulation recognition: Models, datasets, and challenges. *Digit. Signal Process.* **2022**, *129*, 103650. [CrossRef]
58. Liu, Y.; Xu, H.; Qi, Z.; Shi, Y. Specific emitter identification against unreliable features interference based on time-series classification network structure. *IEEE Access* **2020**, *8*, 200194–200208. [CrossRef]
59. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
60. Chang, S.; Zhang, R.; Ji, K.; Huang, S.; Feng, Z. A Hierarchical Classification Head based Convolutional Gated Deep Neural Network for Automatic Modulation Classification. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 8713–8728. [CrossRef]
61. Zhang, F.; Luo, C.; Xu, J.; Luo, Y. An efficient deep learning model for automatic modulation recognition based on parameter estimation and transformation. *IEEE Commun. Lett.* **2021**, *25*, 3287–3290. [CrossRef]
62. Xu, J.; Luo, C.; Parr, G.; Luo, Y. A spatiotemporal multi-channel learning framework for automatic modulation recognition. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 1629–1632. [CrossRef]
63. Zhang, Z.; Luo, H.; Wang, C.; Gan, C.; Xiang, Y. Automatic modulation classification using CNN-LSTM based dual-stream structure. *IEEE Trans. Veh. Technol.* **2020**, *69*, 13521–13531. [CrossRef]
64. 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]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.