Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 1991471, 12 pages
https://doi.org/10.1155/2021/1991471

Research Article

Automatic Modulation Recognition Based on Hybrid Neural Network

Qiang Duan,[1] **Jianhua Fan**,[1] **Xianglin Wei**,[1] **Chao Wang**,[2] **Xiang Jiao**,[2] **and Nan Wei**[2]

[1] *63rd Research Institute, National University of Defense Technology, Nanjing 210007, China*
[2] *School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China*

Correspondence should be addressed to Jianhua Fan; duanqiang1997@163.com and Xianglin Wei; wei_xianglin@163.com

Received 10 August 2021; Revised 10 September 2021; Accepted 9 November 2021; Published 27 December 2021

Academic Editor: Narasimhan Venkateswaran

Copyright © 2021 Qiang Duan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recognizing signals is critical for understanding the increasingly crowded wireless spectrum space in noncooperative communications. Traditional threshold or pattern recognition-based solutions are labor-intensive and error-prone. Therefore, practitioners start to apply deep learning to automatic modulation classification (AMC). However, the recognition accuracy and robustness of recently presented neural network-based proposals are still unsatisfactory, especially when the signal-to-noise ratio (SNR) is low. In this backdrop, this paper presents a hybrid neural network model, called MCBL, which combines convolutional neural network, bidirectional long-short time memory, and attention mechanism to exploit their respective capability to extract the spatial, temporal, and salient features embedded in the signal samples. After formulating the AMC problem, the three modules of our hybrid dynamic neural network are detailed. To evaluate the performance of our proposal, 10 state-of-the-art neural networks (including two latest models) are chosen as benchmarks for the comparison experiments conducted on an open radio frequency (RF) dataset. Results have shown that the recognition accuracy of MCBL can reach 93% which is the highest among the tested DNN models. At the same time, the computation efficiency and robustness of MCBL are better than existing proposals.

1. Introduction

Wireless networks are currently undergoing dramatic development. With the increase of both the number and diversity of wireless devices, their spectrum demand is increasing too [1]. At the same time, the spectrum is not fully utilized due to the shortage of the knowledge about the spectrum usage. Therefore, monitoring and understanding the use of spectrum resources play an important role in improving and standardizing the use of precious radio frequency spectrum. To achieve this goal, realizing efficient modulation recognition is critical for detecting and utilizing wireless signals. However, modulation recognition usage in such a complex wireless systems need distributed sensing in a wide frequency range, leading to the flooding of a large volume of spectrum data. Extracting meaningful modulation information from a large amount of data requires more advanced algorithms. This paves the way for new innovative spectrum access schemes and the development of novel identification mechanisms about the radio environment [2]. Therefore, Automatic Modulation Classification (AMC) based on machine learning, and in particular deep learning, has been one of the practitioners' focus in wireless communications.

AMC plays a critical role in understanding the signals transmitted in an interested area in non-cooperative communications [3]. Traditional modulation recognition algorithms, including maximum likelihood hypothesis [4] and statistical pattern recognition, are labor-intensive in their feature extraction process, and their recognition accuracy severely relies on the prior knowledge about the signals [5]. Moreover, the accuracy and robustness of these two types of methods can be extremely low when limited or nonrepresentative features are adopted for recognition.

To enable a fully automatic feature extraction, several deep neural network- (DNN-) based proposals have been put forward recently, including back propagation (BP) neural network [6], convolutional neural network (CNN) [7], long short-term Memory (LSTM) [8], CGDNet [9], and

ECNN [10]. However, extensive experiments on the open dataset [11] have shown that the recognition accuracy of existing proposals is still unsatisfactory (in Section III) since they cannot fully capture the temporal-spatial characteristics of the signals. Moreover, under low signal-to-noise ratio (SNR), the recognition accuracy of these proposals could be extremely low.

In this backdrop, we introduce a robust and cost-efficient hybrid dynamic neural network structure which is motivated by the architecture proposed in [12]. The model is called multilevel attention CNN Bi-LSTM (MCBL), which combines CNN and Bi-LSTM to exploit their respective capability in automatic spatial and temporal feature extraction. Moreover, in order to improve the efficiency of the model, a multilevel attention mechanism is integrated into the neural network to dynamically extract and pay attention to the salient features included in both the input signal samples and the features extracted by the neural network. Our main contributions are threefold:

(1) A hybrid dynamic neural network that combines CNN, Bi-LSTM, and attention mechanism is put forward to conduct AMC

(2) A global attention mechanism is integrated in our recognition model to improve the training efficiency and prevent model overfitting

(3) Extensive experiments are conducted on the open dataset to compare our proposal with 10 other neural networks include two latest deep learning models that can be utilized for the same purpose. Results have shown that our proposal outperforms other counterparts in recognition accuracy

The reminder of this paper is organized as follows. Section II summarizes the related work. Section III introduces the framework and details the algorithm design. Section IV introduces the experimental settings and analyzes the results. Section V briefly concludes this work.

2. Related Work

2.1. Traditional ML Methods for Modulation Recognition. Previous research work in wireless communication related to modulation recognition is mainly based on signal processing tools for communication [13], such as cyclostationary feature detection [14], sometimes combined with traditional machine learning techniques (e.g., decision tree [15], support vector machine (SVM) [16], and naive Bayes [17]). It turns out that the design of these professional solutions is very time-consuming, because they usually rely on manual extraction of expert features and require a lot of domain knowledge.

2.2. Deep Learning for Modulation Recognition. Motivated by the remarkable success of deep learning, especially convolutional neural networks (CNN), the image recognition, speech recognition, machine translation, and other aspects have made great progress. Wireless communication engineers have recently used similar methods to improve the state of the technology in the modulation recognition task. One of the pioneers of domain names is O'shea et al. [3], who proved that CNN is trained on inphase and quadrature-phase (IQ) data in the time domain better than the traditional AMC methods obviously. Besides, they have implemented a CNN-based modulation recognition framework, named VT-CNN2, which consists of two convolutional layers and two dense layers and is tested on an publicly available dataset [18]. Tara et al. have put forward a model called CLDNN which combined the advantages of CNN, LSTM, and DNN to improve recognition accuracy [19]. In addition to the CNN-based model, LSTM architecture with time-correlated amplitude and phase information can achieve superior classification accuracy [7]. Njoku et al. proposed a CGDNet composed of a shallow convolutional network, a gated recurrent unit and a deep neural network which can incur a low computational complexity and reach high accuracy on DeepSig dataset [9]. Kim et al. extended the input size to $4 \times N$ by copying and concatenating the data in reverse order to enhance the classification accuracy [10]. Wang et al. introduced a federated learning-(FL-) based AMC (FedeAMC) whose advantage is low risk of data leakage without sever performance loss. Results demonstrated that the gap of FedeAMC and CentAMC is less than 2% [20]. Besides, Fu et al. proposed a lightweight AMC module called DecentAMC using model aggregation and lightweight design. Simulation result shows that the DecentAMC substantially reduced the storage and computational capacity requirements of the model [21].

3. System Model

This section first presents the AMC problem; then, we introduce the overall structure of the MCBL model; afterwards, the three modules in the model are separately detailed.

3.1. Problem Statement and Basic Idea. The essential differences between different modulation modes lie in their action modes for base-band signal amplitude, phase, and frequency which will be reflected by the modulated signals and the derived features [22, 23]. Therefore, feature extraction has always been the core of AMC. Traditional feature extraction based on pattern-recognition is labor-intensive, time-consuming, and domain-knowledge-dependent [24]. Recently, DNN-based AMC has been popular due to their unique capability for automatic feature extraction as mentioned above, but their accuracy and latency is still unsatisfactory when applying to complicated and diversified modulation modes [25, 26]. To promote the recognition accuracy and the computation efficiency, this paper develops a framework utilizing CNN and Bi-LSTM as the pattern-digger and multilevel attention to select the salient feature for classifying different modulation modes.

The inphase and quadrature-phase (IQ) data of modulated signal is intended as a two-dimensional image and is adopted as the input data as existing efforts do [27]. Existing efforts usually classify different modulation modes through extracting the spatial or temporal features from the input

IQ data. However, experimental results have shown that relying solely on the features in one domain could not achieve high accuracy (refer to Section III). This motivates us to develop our hybrid DNN framework, called MCBL.

3.2. Model Architecture. MCBL network contains three modules: ① CNN-based spatial feature extraction (CSFE) module, ② Bi-LSTM-based temporal feature extraction (BTFE) module, and ③ multilevel attention-based salient features extraction (MSFE) module. The overall structure of the MCBL network is shown in Figure 1.

Inspired by ECNN [10], before training the module, the whole dataset S is extended as $2 \times 2N$ by copying data and concatenating in reverse order to improve the recognition accuracy:

$$S_{\psi} = \begin{bmatrix} s_{I,0} \cdots s_{I,N-1} & s_{I,N-1} \cdots s_{I,0} \\ s_{Q,0} \cdots s_{Q,N-1} & s_{Q,N-1} \cdots s_{Q,0} \end{bmatrix}. \tag{1}$$

Then, the data is divided into two subsets:

$$S = \left\{ \{ (s_1, l_1), \cdots, (s_a, l_a) \}, \left\{ \left(\tilde{s}_1, \tilde{l}_1 \right), \cdots, \left(\tilde{s}_b, \tilde{l}_b \right) \right\} \right\}, \tag{2}$$

where l_i and $\tilde{l}_j$ are the labels of the i-th training sample s_i and the j-th testing sample $\tilde{s}_j$,respectively, and a and b are the number of training and testing samples, respectively.

Based on a set of training samples $\{s_1, s_2, \cdots, s_a\}$, CSFE module builds a few spatial feature maps. Then, these feature maps are treated as the inputs of the BTFE module for temporal feature extraction. Afterwards, MSFE weights the temporal features and the input samples to determine the salient features.

The MCBL algorithm is trained in a supervised manner. In the training phase, each training sample and its true label are put into the network during the forward propagation, and the parameters are updated through back propagation. In the testing phase, $\{\tilde{s}_1, \tilde{s}_2, \cdots, \tilde{s}_b\}$ are inputted into the network to obtain their predicted value $\{\tilde{l}_{p1}, \tilde{l}_{p2}, \cdots, \tilde{l}_{pb}\}$. Finally, the predicted labels are compared with the true labels to obtain the recognition accuracy. The target of each step in training process can be expressed as

$$\min \frac{1}{a} L(\phi(S), (l_1, l_2, \cdots, l_a)), \tag{3}$$

where L is the loss function, and $\phi(S)$ represents the function of the MCBL model.

3.3. CSFE Module. A CNN model is designed in the CSFE module to achieve automatic spatial feature extraction from the inputs [28]. Each input data sample is treated as a two-dimensional image. The CNN model contains three convolutional layers: Conv1, Conv2, and Conv3, and ReLU function is used as the unit activation function. The structure of the CSFE module is shown in Figure 2.

The number of convolutional kernels in Conv1, Conv2, and Conv3 is 16. The size of each convolutional kernel is (1, 3), (1, 5), and (1, 7), respectively. The size of feature map that CSFE module gets is (p, q, c), in which p and q are the dimension of the feature map, and c is the number of channels.

Zero padding method is adopted to fill zero on the data edge before each convolutional, in order to ensure the data dimension match. More importantly, the dropout method is utilized to prevent overfitting of the network. Before entering Bi-LSTM, we performed a dimensional transformation of the data to ensure that it meets the dimensional requirements of the BTFE model. The new feature dimension is $(p \times q, c)$.

3.4. BTFE Module. The DNN model for extracting temporal feature embedded in a signal sample is Bi-LSTM since it could better capture overall information of time series data than LSTM [29]. The structure of the adopted Bi-LSTM network is shown in Figure 3.

As shown in Figure 3, each feature map extracted by CSFE module is transformed into c time series, and the length of each series is $p \times q$. Bi-LSTM is a special category of LSTM for processing sequential data [30, 31]. Benefited from a specific LSTM memory cell mechanism, Bi-LSTM effectively solves the exploding and vanishing gradient problem of traditional RNN during training process. Specifically, Bi-LSTM combines a LSTM network that moves from the beginning of the sequence and a LSTM network that moves in the opposite direction. In this way, both previous and future information can be utilized in the output layer [32].

The output of BTFE module $\{y_1, y_2, \cdots, y_{2n}\}$ is the extracted temporal feature sequence. Because of the output vectors of the Bi-LSTM layers are processed by connecting the forward LSTM and backward LSTM, the output dimension of BTFE module is $2n$, in which n is the output dimension of forward LSTM.

3.5. MSFE Module. The attention mechanism [33] is selected to focus on part of the input related content and ignore other content. On the one hand, it can make the results more accurate, and on the other hand, it can solve the problem of high computational complexity. For now, attention is widely used in Natural Language Generation (NLG), dialogue systems, multimedia description (MD), text classification, recommendation systems, sentiment analysis, and other tasks [34, 35]. In MCBL, a multilevel attention mechanism is included to extract the salient feature, which contains two parts, i.e., attention block and global attention block.

The timing feature sequence $y(t)$ and input data $s(t)$ are adopted to calculate the element product with the attention factor. For an image, the attention mechanism is to make the network pay attention to the dominant characters, such as the contrast of pixels in color, intensity, and texture [36]. In the signal series considered here, the salient character refers to the contrast between continuous signal data, i.e., the change trend embedded in the data. For a specific data fragment, the higher the probability that it contains the feature for identifying the sample's modulation type (salient feature), the larger its attention factor (between 0 and 1) will

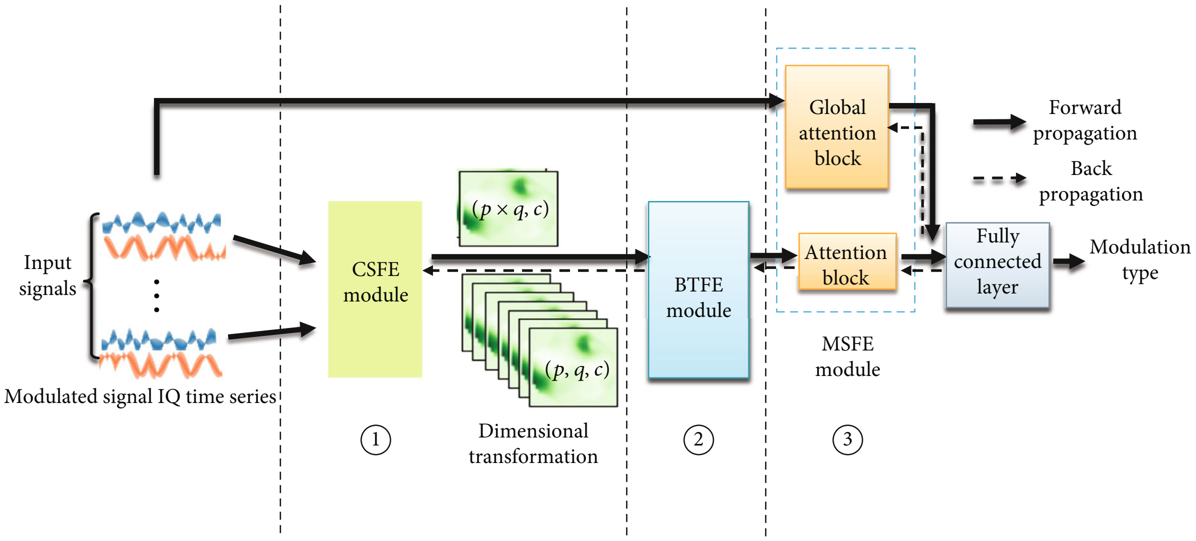


Figure 1: The architecture of MCBL network.

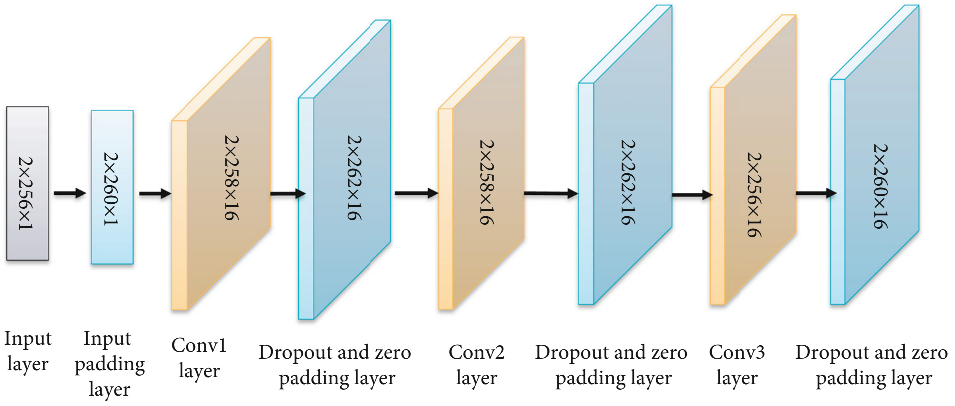


Figure 2: The structure of the CSFE module.

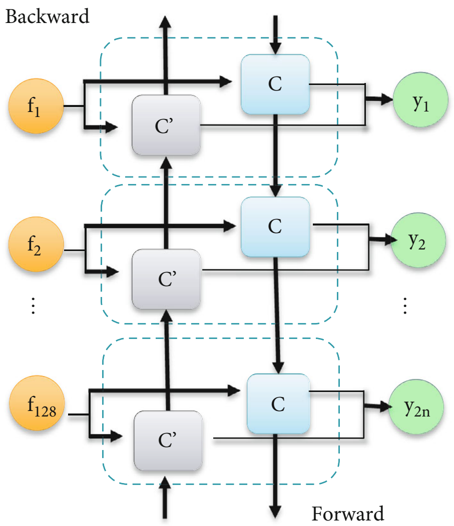


Figure 3: The structure of the adopted Bi-LSTM.

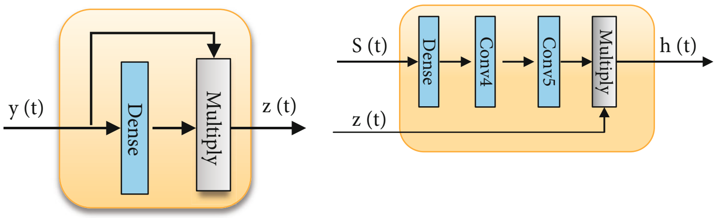
be. Thus, the attention block can dynamically pay attention to salient information and ignore irrelevant background information.

(1) *Attention block*: the timing feature sequence $y(t)$ is put into the dense layer and then the multiply layer, as shown in Figure 4(a). Let $A(t)$ be the attention factor, which indicates the importance of the current feature. Then, we have

$$\begin{gathered} A(t)=\sigma(W(t)\times y(t)+b_1), \\ z(t)=y(t)\times A(t), \end{gathered} \tag{4}$$

where σ represents the ReLU function, $W(t)$ is the weight vector of the Dense module, b_1 represents the offset of the attention module, and $y(t)$ is the timing feature sequence processed by BTFE module.

This method uses $A(t)$ and $y(t)$ to conduct element product. Therefore, it can weight all input timing features



(a) Attention block

(b) Global attention block

Figure 4: The structure of the attention blocks in MCBL.

sequence one by one and pay attention to salient temporal features extracted by BTFE module dynamically.

(2) *Global attention block*: the input data $\boldsymbol{s}(\boldsymbol{t})$ are put into a dense layer and two convolutional layers (Conv4, Conv5). The number of convolutional kernels in Conv4 and Conv5 is 16 and 1, respectively. The size of the convolutional kernels is (1, 3). Conv5 is adopted to compress the number of channels; thus, the output result can match the dimension of $\boldsymbol{z}(\boldsymbol{t})$. The formula to calculate the global attention factor $\mathbf{GA}(\boldsymbol{t})$ is

$$\begin{gathered} \mathrm{GA(t)} = \sigma\left(W(t)_{\mathrm{Conv4}} \times W(t)_{\mathrm{Conv5}} \times s(t) + b_2\right), \\ h(t) = z(t) \times \mathrm{GA}(t), \end{gathered} \tag{5}$$

where σ is the ReLU function, $W(t)_{\mathrm{Conv4}}$ and $W(t)_{\mathrm{Conv5}}$ represent the weight vectors of Conv4 and Conv5, b_2 is the offset of the global attention block, $\mathrm{GA}(t)$ refers to the global attention factor at moment t, $z(t)$ is the output of the attention block, and $h(t)$ is the salient features after feature selection at time t to the whole modulated signal sequence. By calculating element product with the global attention factor $\mathrm{GA}(t)$ and $z(t)$, the network can dynamically select salient features from a global perspective. Besides, since the weight of irrelevant feature maps is 0, a large number of feature maps are removed, and this method can effectively solve the overfitting problem caused by the background noise in data samples.

3.6. Algorithm Description. The pseudocode in Algorithm 1 describes the MCBL model in algorithm manner. It takes the modulation signal data as the input while outputs the trained MCBL model and the recognition results. Algorithm 1 contains 3 parts, i.e., data preprocessing shown from steps 1 to 5, model training shown from steps 6 to 21, and model testing from steps 22 to 25. In part 1, each inputted signal series is reordered in the reverse order in steps 2-4, and then the reversed series is concatenated at the tail of the original series in step 5. In part 2, the MCBL model is trained utilizing the training set until the loss function is lower than the thresholds. The weights update process in conducted from steps 18 to 20. In part 3, the testing data is feed into the trained MCBL model to evaluate its recognition performance.

4. Experiments and Results Analysis

After introducing the adopted dataset, we show the experimental settings for comparing MCBL model with 10 other DNN models. Then, the results are analyzed with different parameter settings.

4.1. Dataset. All experiments are conducted on the RML2016.10b, aka., "DeepSig" dataset, which was collected and opened to the public by DeepSig [11]. The dataset contains 10 modulation modes: 8 digital modulations (BPSK, QPSK, 8PSK, 16QAM, 64QAM, GFSK, CPFSK, and PAM4) and 2 analog modulations (WB-FM and AM-DSB). Note that during the generation of the data samples, the influence of the transmission environment and devices is taken into consider. Each data sample is attached with a SNR value that ranges from -20 dB to 18 dB, with an interval of 2 dB. The dataset simulates real-time radio communication signals using different modulations in various SNRs. The transmitted data includes voice and text formats. For digital modulation, a block randomizer is used in the device to calculate the bits. Therefore, the GNU channel model block is used to generate the dataset, and the 128 sample window technology is used to cut into the time series. During data acquisition, a number of error effects are added in channel environment, such as time varying multipath fading of the channel impulse response, random walk drifting of carrier frequency oscillator and sample time clocks, and additive Gaussian white noise.

The dataset contains 1,200,000 data samples, and the number of each modulation type under a single SNR is 6,000. Each data sample contains the inphase data and quadrature-phase data with a size of 2 × 128. The constellation diagrams of 10 modulated signals under different SNRs are shown in Figure 5. 70% of the data is adopted for training, and the remaining 30% is used for testing.

4.2. Experimental Settings. In order to improve the training efficiency, a dynamic learning rate is adopted. When val_loss does not decrease within 30 epochs, or the learning rate is reduced to the minimum learning rate 0.000001, the training process will be early terminated. The number of training epochs is set to 2000, and the batch size is 1024. To prevent overfitting, each layer adopts the Dropout technique, and the dropout rate is set to 0.6.

MCBL is trained from scratch with randomly initialized weights using adaptive moment (Adam) optimizer. The

```
Input: Modulation signal dataset
Output: MCBL model, Recognition rate, Confusion matrix
Part 1: Data preprocessing
for $i = 0 \longrightarrow N - 1$ do
    $\tilde{S}[i] \longleftarrow S[N - 1 - i]$
end for
$S_\psi \longleftarrow [S, \tilde{S}]$
Part 2: Model training
$W, b \longleftarrow 0$(Initialization);
while $loss = (1/a)L(\phi(S), (l_1, l_2, \cdots, l_a))$ decrease within 30 epochs do
    for $t = 0, 1, 2 \cdots$ do
        $f(t) \longleftarrow \sigma(W_{CSFE} \times s(t) + b_{CSFE})$
        //$W_{CSFE}$ and $b_{CSFE}$ are the weights and bias of CSFE module, and $\sigma$ is the ReLU activation function
        $y_{FW}(t) \longleftarrow \sigma(W_{BTFE} \times f(t) + b_{BTFE})$
        //$W_{BTFE}$ and $b_{BTFE}$ are the weights and bias of BTFE module
        $Y \longleftarrow [y_{FW}, y_{BW}]$
        $A(t) = \sigma(W(t) \times y(t) + b_1)$
        //$y(t)$ is an element in set $Y$, and is a vector
        $z(t) = y(t) \times A(t)$
        $GA(t) = \sigma(W(t)_{Conv4} \times W(t)_{Conv5} \times s(t) + b_2)$
        //$GA(t)$ is the global attention factors
        $h(t) = z(t) \times GA(t)$
    end for
    for $k = 0, 1, 2, \cdots$ do
        $W_k(t + 1) \longleftarrow W_k(t) + \nabla_k{}^{loss}(t)$
    end for
end while
Part 3: Model testing
Input testing data$\{(\tilde{s}_1, \tilde{l}_1), \cdots, (\tilde{s}_b, \tilde{l}_b)\}$ into the model
$result \longleftarrow Recognition\ rate, Confusion\ matrix$
return $result$
```

Algorithm 1: Algorithm description of the MCBL model.

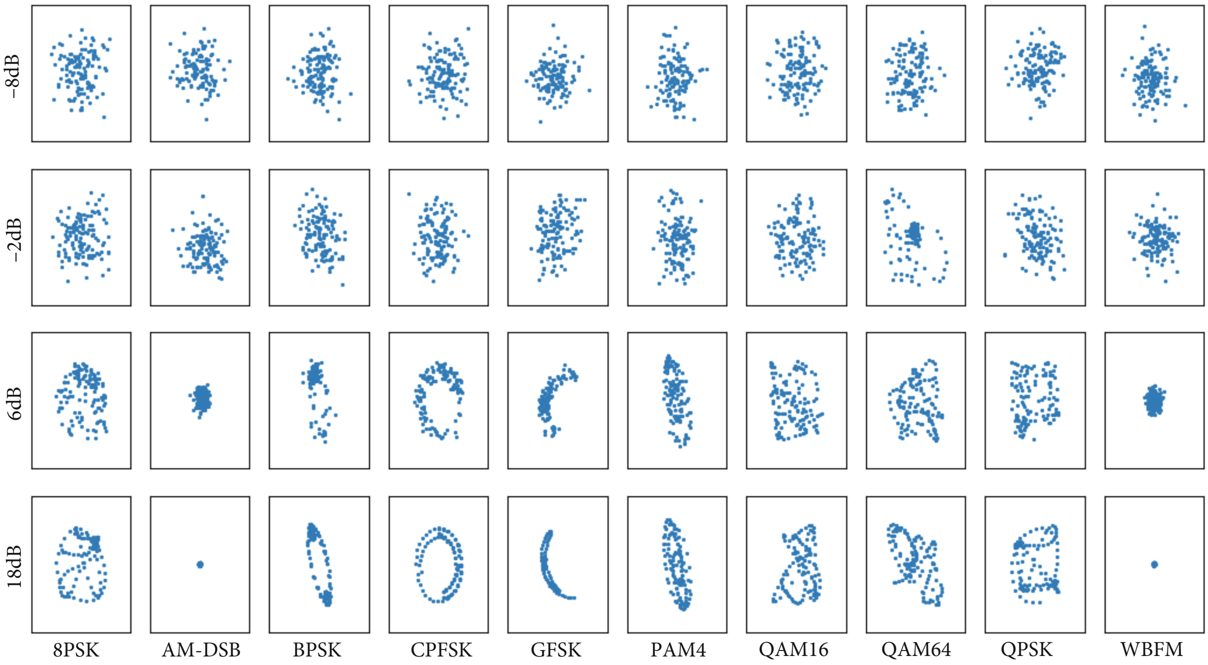


Figure 5: Constellation diagrams of 10 modulated signals under different SNRs.

model is trained with the categorical crossentropy as the loss function. To obtain the best parameters for training the model, a brute force technique was employed. Thus, we applied the model several times until the best parameters with the best performance were obtained. The specific parameters used in the model and the output volume of each layer are shown in Table 1.

4.3. Comparison Benchmark. 10 DNN models are chosen as comparison benchmarks, i.e., AlexNet [37], CLDNN [19], VT_CNN2 [18], LSTM [38], ResNet [39], VGG [40], CNN_LSTM [41], and two latest models ECNN [10] and CGDNet [9]. All of these algorithms use the same dataset without any preprocessing.

(1) Traditional CNN structures: AlexNet, ResNet, VGG, and VT_CNN2. These models contain 5, 18, 19, and 2 convolutional layers, respectively, to extract the features and two dense fully connected layers. Their kernel sizes are (1, 3), (1, 3), (1, 7), and (1, 5), respectively. In order to prevent model overfitting, the dropout technique is adopted in each layer. Besides, the activation function is ReLU

(2) Hybrid neural networks: CGDNet [9], ECNN [10], CNN_LSTM [41], and CLDNN [19]. CGDNet contains 2 convolutional layers, GRU, and DNN, to improve the recognition accuracy and reduce computation time, its kernel size is (1,6), and its activate function is ReLU. Besides, ECNN contains 3 types of network blocks to extract the features for improving the accuracy. CNN_LSTM contains 3 convolutional layers to extract spatial features and LSTM layers to extract temporal features. CLDNN consists of a three-layer CNN and a dropout layer to extract feature and prevent overfitting: a LSTM network of 250 layers and a DNN of 256 units. The convolutional layers of CLDNN make use of 50 filters of size 1 × 7 and the ReLU activation function

(3) DNN: This DNN contains 3 dense layers of size 256, 64, and 32, and each layer is followed by dropout layer. The dropout ratio is 0.5. The dense layers adopt ReLU as the activation function

MCBL and all the baseline models are trained on the same platform which equipped with Intel(R) Core (TM) i5-8300H CPU @ 2.30GHz, 8GB DDR4 RAM, and NVIDIA GeForce 1060 8GB, and the programming environment is Python 3.7, Tensorflow 2.7, and PyCharm, to make a fair comparison.

Table 1: MCBL network parameters.

Layer	Output volume	Description (or remarks)
Input	(2, 256, 1)	
Input padding	(2, 260, 1)	Zero padding (0, 2)
Conv1	(2, 258, 16)	(16, (1, 3))
Dropout1	(2, 258, 16)	Dropout 0.6
Zero padding1	(2, 262, 16)	Zero padding (0, 2)
Conv2	(2, 258, 16)	(16, (1, 5))
Dropout2	(2, 258, 16)	Dropout 0.6
Zero padding2	(2, 262, 16)	Zero padding (0, 2)
Conv3	(2, 256, 16)	(16, (1, 3))
Dropout3	(2, 256, 16)	Dropout 0.6
Zero padding3	(2, 260, 16)	Zero padding (0, 2)
Reshape	(520, 16)	Dimension transform
Bi_LSTM	(100)	merge_mode = concat
GA_Dense	(2, 256, 64)	64
Dense1	(64)	64
GA_Conv4	(2, 254, 16)	(16, (1, 3))
A_Dense	(64)	64
GA_Conv5	(2, 252, 1)	(1, (1, 3))
Multiply1	(64)	
Flatten	(504)	
Dropout4	(64)	Dropout 0.6
GA_Dense2	(64)	512
Multiply2	(64)	
Dense2	(10)	10
FC layer	(10)	Softmax

4.4. Comparison Metrics. There are 4 comparison benchmarks commonly used which are accuracy, precision, recall, and F1-score, respectively. For a multiclassification problem, the distinguish of a particular class and all other classes can be regarded as a two-class classification problem. All those samples that belong to this class are called positive, and all other samples are called negative. In this way, after deriving the classification results, we have 4 categories: true positives (TP) are examples correctly labeled as positives; false positives (FP) refer to negative examples incorrectly labeled as positive; true negatives (TN) correspond to negatives correctly labeled as negative; and false negatives (FN) refer to positive examples incorrectly labeled as negative [42]. The calculation formulas for accuracy, precision, recall, and F1-score are defined as follows.

$$
\begin{aligned}
\text{Accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \\
\text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\
\text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\
\text{F1} - \text{score} &= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.
\end{aligned}
\tag{6}
$$

After calculating these four metrics for all classes, their average values are adopted as the metric values of the adopted DNN model. For example, the accuracy of a model is $\sum_{i=1}^{g} a_i / g$, where g is the number of classes, and a_i is the accuracy of the i-th class.

4.5. Recognition Performance Analysis. The comparison results of the MCBL model and 10 benchmarks with

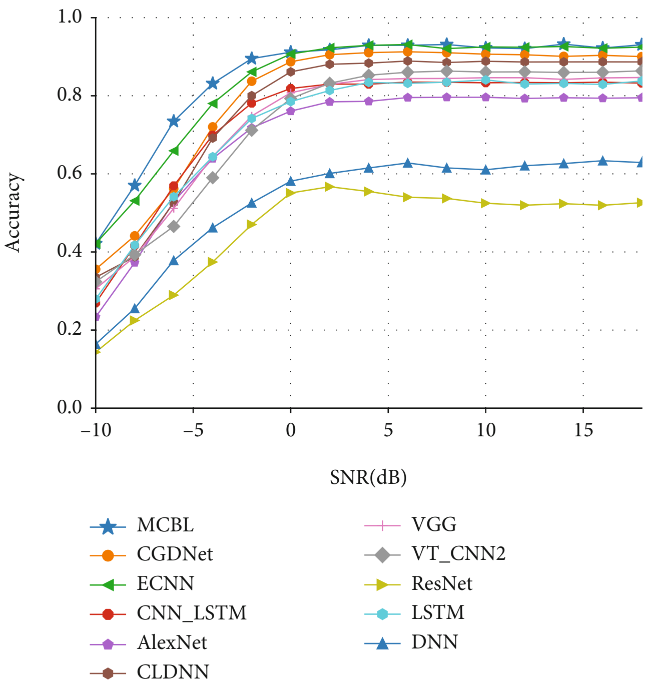


Figure 6: Model performance under different SNR. Note: the recognition results of ECNN model are from reference [10].

Table 2: Comparison results.

	MCBL	CGDNet	VT_ CNN2	CLDNN	ResNet	VGG
Precision	93%	90.03%	86.38%	88.67%	61.0%	86.73%
Recall	92.06%	88.59%	83.61%	86.81%	52.61%	84.67%
F1-score	92.52%	89.30%	84.97%	87.73%	54.49%	85.68%

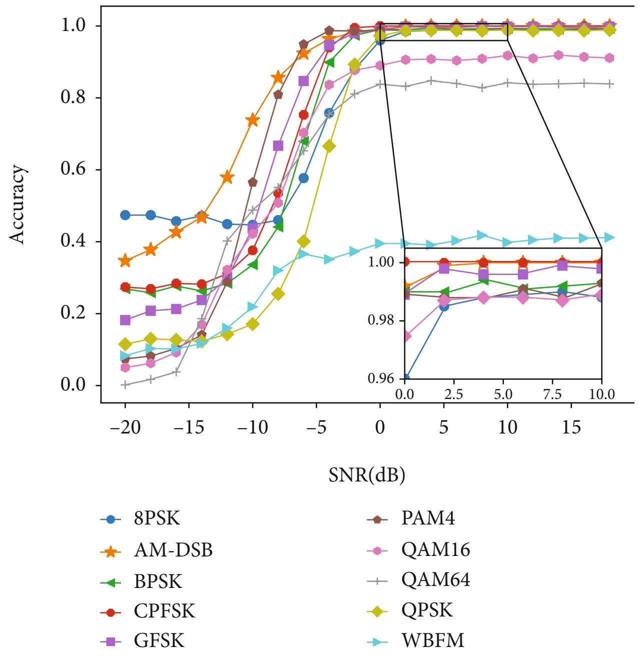


Figure 7: Classification accuracy for each modulation type.

different SNR values are shown in Figure 6. From Figure 6, we can see that MCBL achieves the highest overall recognition accuracy of 93% across the entire SNR range. At the same time, for each recognition method, the general trend is that with the increase of the SNR value, the recognition accuracy increases. Moreover, MCBL always performs the best among all recognition models. This is due to the fact that MCBL is more effective than other models in extracting and selecting temporal, spatial, and salient features, which greatly improves its recognition accuracy. Note that the accuracy of ECNN is close to MCBL when the SNR is higher than 0 dB. However, when the SNR is between -8 dB and -2 dB, MCBL model's accuracy is nearly 10% higher than ECNN model. When the SNR is -4 dB, MCBL can still reach a recognition rate of 83.32% that is 5% higher than that of ECNN, indicating that MCBL is more robust than other methods under low SNR values. When the SNR is lower than -6 dB, the differences between different modulation signals are vague and thus increasing the difficulty of distinguishing them. The features extracted by different DNN models tend to be inaccurate, leading to low recognition accuracy. It can also be seen from Figure 5 that the constellation diagrams of all modulated signals under low SNR are overlapped and cannot be distinguished.

Table 2 shows the comparison results of MCBL and 5 other models in precision, recall, and F1-score.

It can be seen from Table 2 that the precision, recall, and F1-score of MCBL is higher than other deep learning models including the latest model CGDNet. It shows that the classification and prediction capabilities of MCBL are better than other the common DNN model.

It can be seen from the Figure 7 that MCBL's recognition rate for digital modulation 8PSK, BPSK, CPFSK, GFSK, PAM4, and QPSK is close to 100% when the SNR is higher than 0. Moreover, the recognition rates of QAM16 and QAM64 which are 91.10% and 83.90%, respectively. Thus, one can say that MCBL can accurately extract and select temporal, spatial and salient features embedded in the input data. It can also be seen from Figure 7 that the recognition rate of WBFM is low. This is because the difference between the various types of analog modulation is not reflected in the amplitude and phase. Thus, the constellation of AM-DSB and WBFM is almost the same in Figure 5. Therefore, a few WBFM samples are wrongly classified as AM-DSB. The recognition accuracy of each modulation mode increases with the increase of the SNR. This is because the lower the SNR, the larger the proportion of noise in the signal, and the more irregular the modulation signal will be. Therefore, the lower the SNR value, the more difficult it is to identify the signal.

4.6. Computational Complexity. The complexity of a neural network is divided into time complexity and space complexity [9]. The time complexity is generally measured by floating-point-operations (FLOPs) which is an indicator that is often used to gauge the complexity of an algorithm or model. The space complexity refers to the number of parameters or capacity of the network. The number of parameters and time complexity (in FLOPs) are summarized in Table 3.

Table 3: Computational complexity.

	CBL	CGDNet	ECNN	VT_ CNN2	AlexNet	CLDNN	CNN_ LSTM	LSTM	ResNet	VGG	DNN
Parameters	278 K	124 K	43 K	314 k	27 K	167 K	238 K	59 K	146 K	224 K	543 K
FLOPs	0.00659 g	0.00827 g	0.001315 g	0.00504 g	0.000261 g	0.00768 g	0.00504 g	0.000347 g	0.00336 g	0.0086 g	0.00136 g

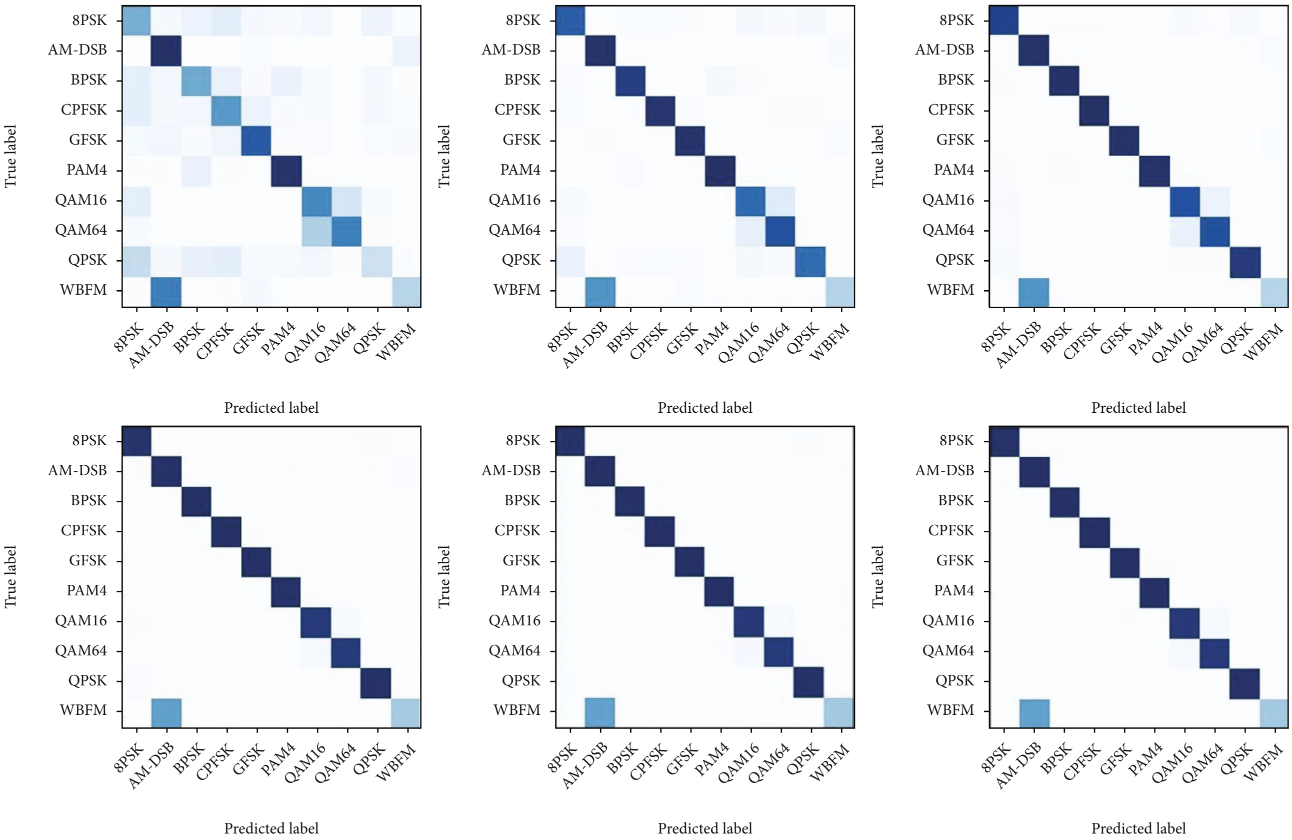


Figure 8: The confusion matrix of MCBL at -8, -4, -2, 2, 6, and 18 dB SNR.

We can see that the number of parameters in the MCBL model is slightly larger than those in CGDNet, VGG, and ResNet models. However, the time complexity of MCBL is lower than these three models, indicating that the spatial, temporal, and salient features in the data can be extracted more efficiently by MCBL. Although MCBL does not achieve the overall lowest space complexity, it achieves the best accuracy among all compared models. Its performance can further be attributed to the use of the BTFE model and the MSFE model, compared to VGG which only use convolutional layers. This not only leads to a high number of trainable parameters but also decreases its time complexity.

4.7. Confusion Matrix. The confusion matrices of the MCBL model at -8 dB, -4 dB, -2 dB, 2 dB, 6 dB, and 18 dB SNR are shown in Figure 8. In a confusion matrix, the deeper the color of the diagonal, the more accurate the classification. When the SNR is -8 dB, the colored grids are spread in the confusion matrix, which indicates a low recognition accuracy. In addition, most WBFM samples are mi-classified as AM-DSB. Under -2 dB SNR, the color of confusion matrix diagonal is deeper than that of the confusion matrix under -8 dB; however, QAM16 and QAM64 cannot be well distinguished. When the SNR value is higher than 0 dB, a higher accuracy can be observed in the confusion matrices in Figure 8.

4.8. The Impact of Modules and Dropout Rate. To evaluate the performance of each module, we experiment the MCBL model with and without each module, respectively. The results are shown in Figure 9. It can be seen in Figure 9 that the recognition accuracy of MCBL without CSFE, BTFE, and MSFE modules (Without_CSFE, Without_BTFE, and Without_MSFE, in Figure 9, respectively) is only 60%, 83%, and 85% when the SNR is 18 dB. Moverover, the recognition accuracy decreases more when the SNR is low. It can be analyzed from the Figure 9 that each module is helpful to the improvement of the overall accuracy, and the CSFE module has the greatest impact on the model.

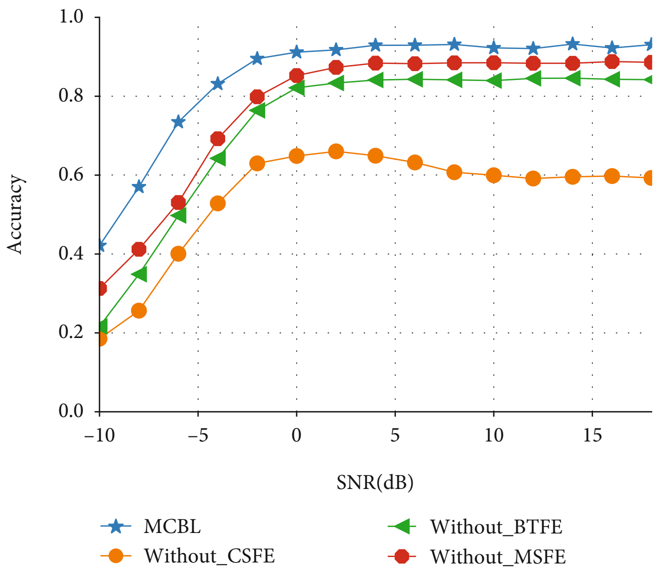


Figure 9: Training accuracy for MCBL and MCBL without each model.

Table 4: MCBL network parameters.

Dropout rate	0.3	0.4	0.5	0.6	0.7	0.8
Accuracy	92.06%	92.28%	92.25%	93%	91.76%	88.08%

The dropout technique is adopted in MCBL to prevent overfitting phenomenon in the training process. The recognition accuracy of the MCBL model with different dropout rates is shown in Table 4. It can be seen from Table 4 that with the increase of the dropout rate, the recognition accuracy increases first and then decreases. The increase is brought by the overfitting avoidance capability of the dropout mechanism. However, when the dropout is too high, too much neurons are neglected by the model, and an underfitting will occur. Generally speaking, the best dropout rate here is 0.6.

5. Conclusion

A hybrid dynamic neural network model, called multilevel attention CNN Bi-LSTM (MCBL), is presented in this paper to achieve automatic modulation recognition. MCBL contains three modules, i.e., CSFE module, BTFE module, and MSFE module to extract and select the spatial, temporal, and salient features of the modulated signals effectively. To evaluate the performance of MCBL network, 10 DNN networks are adopted for the comparison experiments on an open RF dataset. Experimental results have shown that MCBL's recognition accuracy is higher than state-of-the-art proposals. Moreover, the efficiency and robustness of MCBL are better than other models.

Data Availability

The dataset is downloaded from https://www.deepsig.ai/datasets. The name of dataset is RadioML.2016.10b.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] G. Ding, Q. Wu, J. Wang, and Y. D. Yao, "Big spectrum data: the new resource for cognitive wireless networking," http://arxiv.org/abs/1404.6508.

[2] M. Kulin, T. Kazaz, I. Moerman, and E. D. Poorter, "End-to-end learning from spectrum data: a deep learning approach for wireless signal identification in spectrum monitoring applications," *IEEE Access*, vol. 6, pp. 18484–18501, 2018.

[3] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[4] K. M. Chugg, C.-S. Long, and A. Polydoros, "Combined likelihood power estimation and multiple hypothesis modulation classification," in *Conference Record of The Twenty-Ninth Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1137–1141, Pacific Grove, CA, USA, 1995.

[5] S. Huang, C. Lin, W. Xu, Y. Gao, and F. Zhu, "Identification of active attacks in internet of things: joint model-and data-driven automatic modulation classification approach," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 2051–2065, 2021.

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[7] S. Jeong, U. Lee, and S. C. Kim, "Spectrogram-based automatic modulation recognition using convolutional neural network," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 843–845, Prague, Czech Republic, 2018.

[8] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[9] J. N. Njoku, M. E. Morocho-Cayamcela, and W. Lim, “Cgdnet: efficient hybrid deep learning model for robust automatic modulation recognition,” *IEEE Networking Letters*, vol. 3, no. 2, pp. 47–51, 2021.

[10] S.-H. Kim, J.-W. Kim, W.-P. Nwadiugwu, and D.-S. Kim, “Deep learning-based robust automatic modulation classification for cognitive radio networks,” *IEEE Access*, vol. 9, pp. 92386–92393, 2021.

[11] “RF datasets for machine learning,” https://www.deepsig.ai/datasets.

[12] S. Wei, Q. Qu, H. Su, M. Wang, J. Shi, and X. Hao, “Intra-pulse modulation radar signal recognition based on cldn network,” *IET Radar, Sonar & Navigation*, vol. 14, no. 6, pp. 803–810, 2020.

[13] E. Axell, G. Leus, E. Larsson, and H. Poor, “Spectrum sensing for cognitive radio : state-of-the-art and recent advances,” *IEEE Signal Processing Magazine*, vol. 29, no. 3, pp. 101–116, 2012.

[14] K. Kim, I. A. Akbar, K. K. Bae, J. S. Urn, and J. H. Reed, “Cyclostationary approaches to signal detection and classification in cognitive radio,” in *2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 212–215, Dublin, Ireland, 2007.

[15] A. K. Nandi and E. E. Azzouz, “Algorithms for automatic modulation recognition of communication signals,” *IEEE Transactions on Communications*, vol. 46, no. 4, pp. 431–436, 1998.

[16] C.-S. Park, J.-H. Choi, S.-P. Nah, W. Jang, and D. Y. Kim, “Automatic modulation recognition of digital signals using wavelet features and svm,” in *2008 10th International Conference on Advanced Communication Technology*, pp. 387–390, Gangwon, Korea (South), 2008.

[17] Wen Wei and J. M. Mendel, “Maximum-likelihood classification for digital amplitude-phase modulations,” *IEEE Transactions on Communications*, vol. 48, no. 2, pp. 189–193, 2000.

[18] T. J. Oshea, J. Corgan, and T. C. Clancy, “Convolutional radio modulation recognition networks,” in *International conference on engineering applications of neural networks*, pp. 213–226, Springer, 2016.

[19] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, long short-term memory, fully connected deep neural networks,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580–4584, South Brisbane, QLD, Australia, 2015.

[20] Y. Wang, G. Gui, H. Gacanin, B. Adebisi, H. Sari, and F. Adachi, “Federated learning for automatic modulation classification under class imbalance and varying noise condition,” *IEEE Transactions on Cognitive Communications and Networking*, 2021.

[21] X. Fu, G. Gui, Y. Wang et al., “Lightweight automatic modulation classification based on decentralized learning,” *IEEE Transactions on Cognitive Communications and Networking*, 2021.

[22] X. Hao, G. Zhang, and S. Ma, “Deep learning,” *International Journal of Semantic Computing*, vol. 10, no. 3, pp. 417–439, 2016.

[23] A. P. Hermawan, R. R. Ginanjar, D. S. Kim, and J. M. Lee, “Cnn-based automatic modulation classification for beyond 5g communications,” *IEEE Communications Letters*, vol. 24, no. 5, pp. 1038–1041, 2020.

[24] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, “Machine learning for 5g/b5g mobile and wireless communications: potential, limitations, and future directions,” *IEEE Access*, vol. 7, no. 99, pp. 137184–137206, 2019.

[25] Y. Nie, S. Xu, S. Huang, Y. Zhang, and Z. Feng, “Automatic modulation classification based multiple cumulants and quasi-newton method for mimo system,” in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–5, San Francisco, CA, USA, 2017.

[26] Y. Zhao, G. Ren, X. Wang, Z. Wu, and X. Gu, “Automatic digital modulation recognition using artificial neural networks,” in *International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003*, vol. 1, pp. 257–260, Nanjing, China, 2003.

[27] F. Restuccia and T. Melodia, “Big data goes small: real-time spectrumdriven embedded wireless networking through deep learning in the rf loop,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 2152–2160, Paris, France, 2019.

[28] T. Roska and L. O. Chua, “The cnn universal machine: an analogic array computer,” *IEEE Transactions on Circuits & Systems II Analog & Digital Signal Processing*, vol. 40, no. 3, pp. 163–173, 1993.

[29] M. Ma, “Multimedia emergency event extraction and modeling based on object detection and bi-lstm network,” in *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pp. 574–580, Guangzhou, China, 2021.

[30] S. Li, Z. Yan, X. Wu, A. Li, and B. Zhou, “A method of emotional analysis of movie based on convolution neural network and bi-directional lstm rnn,” in *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pp. 156–161, Shenzhen, China, 2017.

[31] Z. Xueqing, Z. Zhansong, and Z. Chaomo, “Bi-lstm deep neural network reservoir classification model based on the innovative input of logging curve response sequences,” *IEEE Access*, vol. 9, pp. 19902–19915, 2021.

[32] L. Liang, X. Ji, and F. Ren, “Attention-based bi-lstm-crf network for emotion cause extraction in texts,” in *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1670–1675, Beijing, China, 2020.

[33] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” http://arxiv.org/abs/1409.0473v7.

[34] Y. Sun and L. Gu, “Attention-based machine learning model for smart contract vulnerability detection,” *Journal of Physics Conference Series*, vol. 1820, no. 1, 2021.

[35] Y. Wang and Y. Wang, “Feature extraction by using attention mechanism in text classification,” in *Data Science*, pp. 77–89, Springer, 2020.

[36] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[37] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, and V. K. Asari, “The history began from alexnet: a comprehensive survey on deep learning approaches,” http://arxiv.org/abs/1803.01164.

[38] M. Sundermeyer, R. Schlter, and H. Ney, “Lstm neural networks for language modeling,” in *InThirteenth annual*

conference of the international speech communication association, pp. 194–197, Portland, Oregon, USA, 2012.

[39] Z. Wu, C. Shen, and A. Hengel, "Wider or deeper: revisiting the resnet model for visual recognition," *Pattern Recognition*, vol. 90, 2019.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, Nevada, USA, 2016.

[41] P. Sun, P. Liu, Q. Li, C. Liu, and J. Chen, "Dl-ids: extracting features using cnn-lstm hybrid network for intrusion detection system," *Security and Communication Networks*, vol. 2020, Article ID 8890306, 11 pages, 2020.

[42] J. Davis, "The relationship between precision-recall and roc curves," in *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, Pittsburgh, Pennsylvania, USA, 2006.