International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 02 | Feb 2026 www.irjet.net p-ISSN: 2395-0072

Hybrid CNN-Transformer Architecture for Enhanced Signal Classification in Wireless Communications

Nagla Elhaj Babiker[1], Khalid Hamid Bilal[2], Magdi B. M. Amien[3]

[1]Department. of Electronics Engineering (Communications and Control), University of Gezira, College of Engineering and Technology.
[2] Professor, Department of Electrical Engineering, Omdurman Islamic University, Omdurman, Sudan.
[3]Department of Electrical and Electronics Engineering, University of Khartoum, Khartoum, Sudan

---

Abstract - *Signal classification in noisy environments is a common issue for wireless communication systems, particularly while working with modulation strategies. The complexity of signal data makes it difficult to use traditional methods that rely on simple convolutional neural network (CNN) architectures and conventional machine-learning models, particularly when operating in environments with high signal-to-noise ratios (SNRs). In addition, for real-world heterogeneous datasets, these methods often lack required generalizability. This work proposes a hybrid CNN transformer model to overcome these constraints. The proposed model performs better at classification when the SNR changes because it combines the sequential modelling capabilities of the transformer architecture with the feature extraction capabilities of the convolutional layers. It is trained using a dataset of 11 modulation schemes under varying SNRs to ensure the resilient performance in various noise settings of the model. The model performance is assessed using its accuracy and bit error rate (BER). The model outperforms standard methods by both accuracy and generalization for classification. The model performance under various settings was examined employing a confusion matrix visualization.*

Key Words: Automatic modulation recognition (AMR), Deep-learning neural networks

1.I NTRODUCTION

Throughout the growth of new wireless communication technologies, the limited availability of radio spectra leads to a considerable drawback. This drawback is not a genuine deficiency of spectrum resources but rather ineffective regulatory frameworks that assign frequencies in a strict and unyielding fashion [1]. A wide range of organizations, such as the civilian, government, commercial, and military organizations, share the electromagnetic spectrum. Therefore, modern wireless communication environments require a dynamic spectrum allocation policy instead of the fixed policy that is commonly adopted today, leading to low spectrum utilization difficulties. Cognitive radio (CR) technology creation lets radios change and use unused frequency resources, it is called “spectrum holes” or “white spaces”. This work indicates the concept of dynamic access to the radio spectrum. By this, secondary users (SUs) can opportunistically access the frequency allocation to primary users (PUs) when they are not in use. This approach aims to enhance spectral efficiency by enabling transmissions on detected free bands, which addresses the issue of spectrum shortage [2].

Because spectrum sensing and detection are becoming more critical in spectrum monitoring, management, and secure communications such as 5G communications and beyond, IoT networks, and other services, CR has essential roles and capabilities in detecting active PU transmissions over the band. Deciding to transmit the sensing outcomes indicates that each PU transmitters is inactive at this band with a high probability, and CR becomes the promotion solution for scarcity and underutilization problems [3]. AMR is an essential item in digital communication systems and serves as a critical element for the effectiveness of CR. It enables dynamic radio resource management by using reconfigurable software-defined transceivers. These transceivers can reconfigure their transmission parameters by using the accessible communication resources in the electromagnetic environment to recognize modulation types of unknown signals without previous data [4]. By using AMR, CR is expected to accurately recognize or classify the modulation structure of the received signal rapidly, without any latency. This process helps to identify whether there is a PU in the channel, as all PUs use a single modulation technique for transmission over the frequency channel. This allows the corresponding demodulation process to be done on the receiving side [5], [6].

AMR is a transitional stage between signal modulation and demodulation. Because it is challenging to distinguish between numerous modulation schemes owing to several factors, including multipath fading, noise, center frequency offset, and signal

structure distortion caused by inadequate hardware design or crystal oscillator drifting, AMR technology is crucial before demodulating signals at the receiver and a vital link in wireless communication. The technique can automatically determine the modulation type of the signal and extract the information it contains without being acquainted with the system specifications. Its automation can advance the signal modulation detection accuracy while significantly reducing the need for human resources, thereby enhancing the accuracy of signal modulation recognition. Manual modulation recognition (MMR) has been replaced by a new automatic method (AMR) in which MMR requires manual observation with an oscilloscope. The MMR method will be limited by four types of information required for the search operator: the intermediate frequency (IF) time waveform, the average as well as instant spectrum of the signal, sound, immediate amplitude, and both long recognition and error recognition times.

AMR plays a major role in CR, which enables dynamic radio resource management by employing reconfigurable software-defined transceivers. These transceivers can reconfigure their transmission parameters by using obtainable communication resources in an electromagnetic environment to identify the unknown signals modulation types without prior knowledge [7].

As in Figure 1, the AMR block consists of two components, such as signal preprocessing and classifier modules. The preprocessing module estimates synchronization parameters, including the frequency offset of the timing recovery, received signal, and power. In the second part, there is elimination of signal disturbances such as interference identification, resulting in enhanced performance.

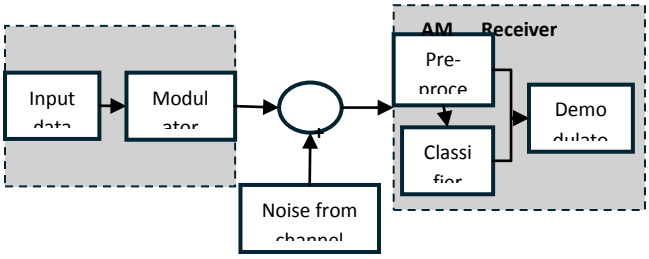


Chart – 1: AMR in the receiver part of communication systems

AMR is classified into two classes. The first comprises the existing AMR techniques, consisting of decision-theoretic approaches and feature-based (FB) methods. These methods depend on the form of the signal input, such as signal statistical approaches and image-based methods. The likelihood-based (LB) approach is the initial decision-theoretic approach. The probability density function (PDF) covers the AMC-LB of the observed waveform also learns from modulated signals and FB methods [8]. LB requires prior knowledge of the PDF and the calculation of the maximum likelihood value for all the proposed modulation schemes, which adds high computational complexity, lacks robustness against model mismatch, and poses a challenge for real-time systems. FB approaches concentrate low computational complexity on features extraction from the signal directly, removing the need for extra channel or signal data and, thus, lowering computational demands. Feature extraction and feature classification can be applied, in which in the feature extraction module, sudden features, high order cumulant (HOC) features, and wavelet features can be used [9]. Most existing well-developed AMC methods are FB-based, particularly with the rise of machine learning (ML) approaches using the deep neural network (DNN), where different ML approaches can be exploited. The support vector machine (SVM), k-nearest neighbour (KNN), etc. are regularly assumed. These methods significantly depend on the extraction and analysis of signal features [10], [11], [12], [13].

The second and most modern class is advanced ML and deep learning (DL) methods, which have drawn increasing care for improved spectrum-sensing detection, precision, and accuracy. DL with artificial neurons organized in a stacked multilayer architecture introduces innovative techniques that significantly improve the efficiency of modern wireless communication systems. Several DL algorithms, plus convolutional neural networks (CNNs), DNN, along with long short-term memory (LSTM) [14], [15], [16], [17], as well as their hybrids, demonstrate significant advantages in addressing challenges like spectrum sensing and AMR in CR. This improvement was done by accurately evaluating critical features of the received signals, such as modulation type, which can be enhanced to enable precise predictions of channel availability. Recently, DL transformer-based models have been developed, introducing a new recognition algorithm for real-time decision-making and leveraging self-attention to reduce complexity concerns in the AMR of signals, thereby achieving accurate modulation classification under practical channel conditions. Recently, transformer-based models explore for real-time decision-making, leveraging self-attention to handle complex spectrum-allocation tasks.

1.1 Problem Formulation

Spectrum sensing is a core function of CRNs; however, several limitations, such as noise and the requirement for preceding information of signal features, causes it to fail in environments with low signal-to-noise ratio (SNR). In addition, recognizing the type of modulation is critical to intelligent spectrum access and interference management. Some existing spectrum-sensing methods, for instance energy detection and matched filtering, combine AMR with spectrum sensing. AMR involves classifying the modulation schemes (e.g., quadrature phase-shift keying, binary phase-shift keying (BPSK), 16-QAM, frequency modulation, and amplitude modulation (AM)) used by the transmitted signal. These schemes can significantly improve the performance and robustness of spectrum sensing, particularly under real-world conditions with varying noise levels, fading, and signal distortions. Because the necessity for autonomous modulation recognition rises in wireless systems, where modulation schemes are likely to alter regularly as the environment changes, DNN are considered as a new and powerful method for AMR, overcoming the reliance on expert analysis and the inability to scale with the increasing signal complexity exhibited by traditional methods. The core issue this work address is the design and development of an accurate, noise-resilient, and computationally efficient DL-based framework for AMR to enhance the spectrum sensing performance in CR networks.

1.2 Contribution

1. This paper presents a data-driven, adaptive, and scalable solution that enhances the CRNs intelligence and reliability. The work is done by combining signal classification and spectrum detection capabilities in a CNN-transformer model to improve modulation recognition, mainly while working in low-SNR environments.

2. The proposed framework trains and validates 11 modulation schemes under various SNR conditions to achieve robust training across noise levels, while comparing them with previous methods that significantly decline at low SNRs.

3. As per the experimental evaluations using accuracy, BER, and confusion matrix analysis, the proposed hybrid model performs better in both classification accuracy and generalization ability than traditional CNN-based and classical ML approaches to improved generalization and accuracy.

This study offers a useful path for creating noise-resilient modulation recognition systems applicable to real-world systems and contemporary wireless networks by fusing sequential modelling with feature extraction in a lightweight hybrid architecture-based CR. Though the existing works have some advantages, these DL-based AMR methods often fail under low SNR environments owing to limited global feature modeling and over-dependence on local patterns. This study purposes to work with these existing gaps by proposing a hybrid CNN-Transformer model, which enhances robustness.

2. Literature Review

AMR is an essential technique in CR networks and possesses the or skill to classify the received signal modulation without past information of the modulation scheme. It is a blind or semi-blind method that enables SUs or CR receivers to sense, classify, and adapt without prior knowledge or coordination to identify the modulation types of unknown signals. Such an approach helps with intelligent spectrum sensing, robustness, and adaptivity, and reduces the signalling overhead required in CR networks [18].

DL network architectures target modulation recognition algorithms to enhance the reliability, simplicity, and effectiveness of AI-based AMC models in wireless communication applications. The results reveal that DL architectures can significantly outperform traditional methods, providing a strong foundation for future research in this area [19], [20].

Several DL techniques based on CNN models are presented for recognition of modulation techniques. In [21], Mohsen 2024 designed two CNN models for the automatic recognition of modulation methods: one based on the RadioML2016.10a dataset and the other using an image dataset. The hyperparameter modification enabled both models to achieve high validation and testing accuracies. DL models, particularly CNNs and RNNs, can learn discriminative representations directly from raw I/Q samples established to avoid the need for manual or handcrafted signal features, are suitable for extracting local waveform patterns, and are extensively used to improve the accuracy of AMR, especially in low SNR environments. However, capturing global features and mitigating irrelevant factors remain challenging for improving the AMR performance.

Hybrid networks have been introduced to improve AMRs by effectively capturing global features and enhancing the model generalization. Several studies [22] have suggested hybrid designs that take advantage of the complementary capabilities of transformers (global context modeling) and CNNs (local feature extraction) and the importance of integrating transformers and convolutional networks to increase classification accuracy in wireless communications at different SNRs. By utilizing both designs advantages, this method provides reliable performance in settings with both low- and high-SNR signals.

Based on the combination between the transformer and LSTM (TLDNN) framework [23], Qu improved the main challenge of AMR, which is capturing global features and enhancing model generalization, by proposing a data augmentation strategy called segment substitution (SS). The SS enhances robustness by altering parts of signals to force the network to rely more on global constellation patterns and less on spurious features or channel artifacts and mitigate the impact of irrelevant inherent variables, such as RF fingerprint characteristics and channel characteristics, and solve the primary problem of AMR. A hybrid convolutional transformer classifier (HCTC) has been designed to classify unknown signals [24]. The HCTC model employs a three-stage framework for features extraction from in-phase/quadrature signals using a convolutional layer, a transformer layer, as well as feature mapping. When comparing, the HCTC model gives superior performance by maintaining high average accuracies across the SNR range. This demonstrates complete robustness and reliability across various practical noise environments. This model shows good classification accuracy and works at high SNR levels. However, it slows down at very low SNR levels from 0 dB and lower, and hence it is limited in noisy communication.

The lightweight radio transformer method for AMC was used in [25], leveraging both large-scale and smaller-scale RadioML 2018.01A datasets to comprehensively evaluate the performance of MobileRaT, considering various modulation schemes and realistic communication conditions. Although the model is useful, it does not directly address the advantages of the CNN front ends for maintaining the local waveform structure. Because it is mostly transformer-centric rather than a complete CNN–transformer hybrid, which limits the front-end advantages of the CNN architecture. The researchers in [26] generated a solution for AMC noise, NMformer, based on a vision transformer (ViT), which is suitable for complex feature relationships and image reconstruction. Constellation diagrams have been generated from modulated signals, converting the signal information into a 2-D representation, which achieves high accuracy across various SNRs and demonstrates strong resilience to out-of-distribution data, outperforming baseline classifiers, compared with traditional CNN-based approaches that focus on local features.

Traditional CNN-based approaches in [21] and [22] work well in capturing local features but often fail to generalize in low SNR environments owing to their limited capability to model long-range dependencies. Alternatively, transformer-based models [24][25][26], while are effective at global feature extraction, commonly overlook localized signal characteristics, which are essential for modulation recognition. Recent hybrid designs like HCTC [24] and NMformer [26] work with this gap but either lack effective local feature encoding or exhibit instability at very low SNRs. The model proposed in this work addresses these shortcomings by joining convolutional layers for robust feature extraction with Transformer layers for sequence modeling. This strategy helps in achieving high classification accuracy and noise resilience across a broad SNR spectrum. This dual-capability framework enhances modulation recognition performance where both local and global signal attributes are important.

3. Proposed Methodology

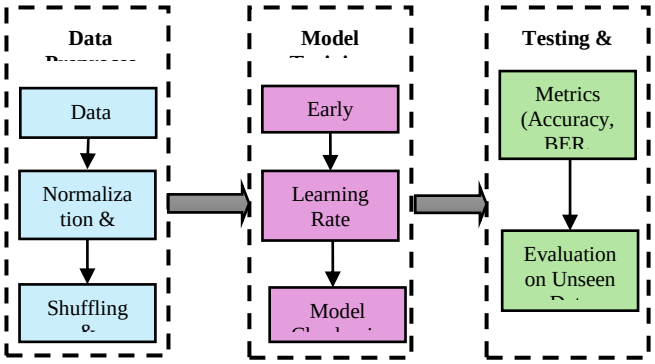


Chart -2: Graphical Abstract

3.1. Dataset

This work utilizes the RML2016.10a dataset, for verifying the results. The RML2016.10a dataset comprises 20 SNRs that ranges from -20 to 18 dB for 22,000 samples in total across 11 modulation schemes. These schemes include WBFM, 4PAM, AM-SSB, BPSK, AM-DSB, 16QAM, CPFSK, GFSK, QPSK, 64QAM, and 8PSK. The samples count of every modulation mode under all SNR condition is 1000. Also, the data format of all the samples is 128 × 2, where 128 signifies the length of the signal. The proposed architecture is shown in figure 2.

3.1.1. Data Splitting

Data splitting is the process of separating the input dataset into two distinct subsets, such as the training, and test sets. The goal is to let the model be trained on a subset of the data, checked on another subset to change hyperparameters, and estimated on a final subset (test set) to guess how well it will do on data that it hasn't seen before. It is common practice to use 80% of the dataset for training purposes. The model is fitted, and its weights are adjusted using this data. By adjusting hyperparameters including the batch size and learning rate, the validation set facilitates model selection. This work uses 20% of the data for testing. There is no bias in the test findings since the model does not view this data when training.

3.1.2. Normalization and One-Hot Encoding

Normalization and one-hot encoding are important processes to prepare the data for DL training and encoding. Scaling the features such that they have comparable ranges, usually between 0 and 1 or -1 and 1, is what normalization does. This is especially important for neural networks because it makes sure that the model treats all features the same. This way, problems don't happen where some features dominate the learning process because they are bigger in equation (1).

$$X_{norm} = \frac{X-\mu}{\sigma} \quad (1)$$

Where μ is the mean and σ is the standard deviation of the feature. A binary matrix may be generated from categorical labels using the one-hot encoding process. Each of the N classes in the dataset is represented by a vector of length N, except for the class's index, which is set to 1. All other values are 0.

3.1.3. Shuffling and Augmentation

If the data is sorted by class or time, for example, shuffling is conducted to prevent any possible biases in the training process. To avoid the model from learning from data order effects, the dataset may be rearranged at random. Data augmentation approaches artificially enlarge the dataset by creating variants of the already-existing data. This might include data modifications like flipping, rotating, or introducing noise, which enhances the model's generalization in equation (2)

$$Augmented\ Data = \{Original\ Data \cup Transformed\ Data\}$$

Tasks involving signal processing may include doing things like introducing random noise to the input signals or adjusting the signal intensity.

3.2. Hybrid CNN-Transformer Model

For some signal processing work, the hybrid CNN-transformer model works with DL architectures. This work makes use of the best features of two different network types, such as the transformer models and the CNNs.

3.2.*1. Convolutional Layers for Feature Extraction*

To extract local features from input data, CNNs makes use of the convolutional layers. CNNs are considered a viable option for signal detection since they detect edges and other high-frequency parts that show important local patterns in the way signal representation is done in terms of time and frequency. In a convolutional layer, the input data passes through a filters, or kernels

set. Each filter looks for various features, including textures, edges, or frequency patterns in signal processing. A 2D convolution operation's mathematical formulation is in equation (3):

$$Y(i,j) = \sum_{m=1}^{M} \sum_{n=1}^{N} X(i+m, j+n).K(m,n) \quad (3)$$

Where $Y(i,j)$ is the output, X is the input data, K is the kernel/filter, and M and N are the kernel dimensions. For classifying the modulations, it is necessary to remove the frequency components and phase shifts from the signals. Additionally, these layers also work well with other signal processing tasks related to this work.

3.2.2. Transformer Layers for Sequence Modelling

Transformers are well-suited for time-series data, such as signal data, since they are used to describe sequential relationships in data. Transformer layers in the hybrid CNN-transformer model can identify long-range relationships in signal data by concentrating on diverse input sequence parts. Mechanisms for self-attention are the basis of the transformer architecture. Prioritizing one part of a sequence above another is the main point. Here is the definition of the self-attention method as in equation (4)

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4)$$

Where Q is the query matrix, K is the key matrix, V is the value matrix, and d_k is the key vector dimension. Self-attention mechanisms may help models by focusing on signal features, such as certain frequency components, that may be important for classifying.

Figure 2 depicts the layer arrangement of the proposed hybrid CNN-transformer model. The architecture contains numerous vital and diverse layers, each performing a unique role. The input to the input layer is an issue-specific 1D or 2D signal, such as time-series or frequency-domain data. The form of a signal consists of its length and the number of channels (e.g., 1 for univariate data and more for multivariate data), which are represented as batch size, sequence length, and Num channels, respectively. On the other hand, convolutional layers extract localized features and patterns from the input data.

The convolution layer learns the important features, like frequency components, edges, or shifts from the input signal. A single convolutional block comprises various layers, which include the convolutional, max-pooling, and dropout layers.

The CNN network utilizes filters to generate feature maps. The maximum pooling method helps in the identification of the highest value in the area of interest. This method keeps the features that are considered most useful along with dimensionality reduction. By changing a unit's portion of the input to 0, the dropout layer prevents overfitting during the training process.

Following the CNN layer, this work uses a flatten layer, which converts the output to a 1D vector and passes it on to the transformer layers. This change turns the feature maps from the CNN layers into a sequence that the transformer accepts. The four layers in this network include a feedforward network, multi-head self-attention, normalization, and a dropout layer. To identify attention scores and highlight the important features, multi-head self-attention uses self-attention on the input sequence. Two fully connected layers in a feedforward network apply a ReLU activation function to the attention outputs. The dropout layer ensures constant training by standardizing the output of every transformer block. By randomly deactivating neurons, the dropout layer prevents over fitting.

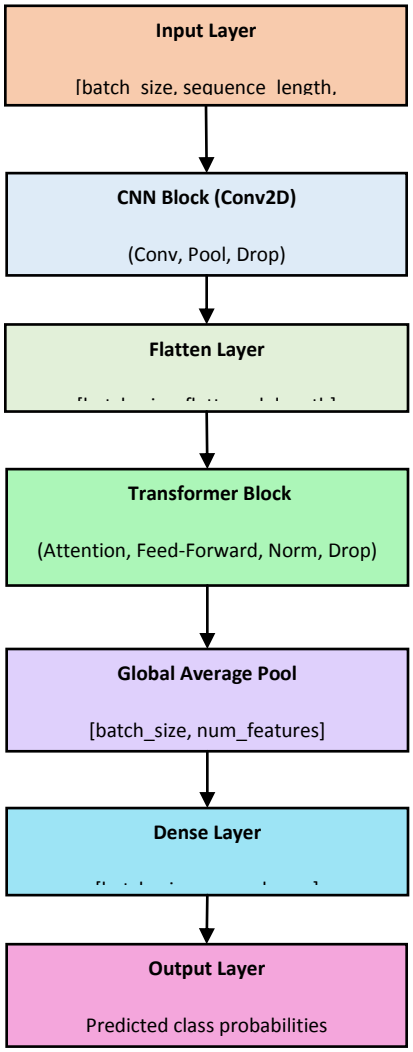


Chart -3: Hybrid CNN-Transformer model

A global average pooling layer helps summarize the learning features in the sequence. This procedure is done after passing through the transformer layers by reducing the sequence dimension and outputting a single value per feature. The output shape of this transformer layer is (batch_size, num_features). A dense layer is inserted after the transformer block to map the learned features to the output classes. This layer is responsible for carrying out the last change. In most cases, this work introduces non-linearity using ReLU or an equivalent activation function. As a last step, the output layer uses softmax as an activation function. The final result for classification is got from this layer.

3.3. Model Training

The ML model learns from the data through the model training phase. This work uses multiple methods to confirm that the model trains and generalizes well.

3.3.1. Early Stopping

Early stopping is meant to avoid the overfitting issue. By this method the training process ends once the validation loss improves for a prearranged epoch count. After the model has achieved excellent generalization, this prevents it from further learning noise from the training data.

3.3.2. Learning Rate Scheduling

This work uses a tool for learning rate scheduling called ReduceLROnPlateau, which changes the learning rate during training. This methodology ensures that the model is fine-tuned as it grows closer to an optimal state by decreasing the learning rate when the validation performance plateaus in equation (5).

$$New\ Learning\ Rate = Old\ Learning\ Rate \times Factor \quad (5)$$

The factor is usually less than 1, and in this work, it is 0.5.

3.3.3. Model Checkpoints

During training, the proposed model keeps a note of its checkpoints for some regular intervals, usually when the validation loss is low. The training process may resume from the optimal condition if overfitting occurs or interruption is necessary.

4. Results

This section discusses the advantages of the proposed approach and its implementation. This study implementation is done in Python by using the well-known libraries like scikit-learn, TensorFlow, and Keras. These libraries serve as building blocks for creation, training, and assessment of the DL models. The proposed hybrid CNN-transformer model combines the best features of convolutional layers for feature extraction also transformer layers for sequence modelling. This model is also suitable for real-world applications in communication systems, as it works better than existing works in generalization and accuracy. Furthermore, the proposed work manages noisy data and long-range relationships in signals.

Table – 1: Model Architecture Table

Layer	Output Shape	Parameters
InputLayer	(None, 128, 2, 1)	0
Conv2D	(None, 128, 2, 64)	256
MaxPooling2D	(None, 64, 2, 64)	0
Conv2D	(None, 64, 2, 128)	24,704
MaxPooling2D	(None, 32, 2, 128)	0
Flatten	(None, 8192)	0
Reshape	(None, 128, 64)	0
MultiHeadAttention	(None, 128, 64)	33,216
LayerNormalization	(None, 128, 64)	128
Flatten	(None, 8192)	0

Dense	(None, 128)	1,048,704
Dropout	(None, 128)	0
Dense	(None, 11)	1,419

Table 1 provides the output shape and the parameters used in this work. Each layer determines the final product's size. The first parameter, None, in the input layer represents the dynamic batch size. The output tensor's width, height, and channel depth are the remaining values in the input layer. The required parameters needed for layer operations, such as weights and biases, add up to the total parameters. This model handles feature extraction, sequence modelling, and classification tasks. The input layer analyzes the data first, then the CNN layers extract spatial features, and finally the transformer layers capture long-range relationships.

From the transformer process, after flattening the output, is a multi-head attention layer that reshapes the data and passes it on. Then the normalization layer stabilizes the learning process. Lastly, decision-making and classification are done by dense layers, which include dropout layers to avoid overfitting. The final output layer predicts the class labels using a softmax activation function.

4.1. Testing and Evaluation

Once trained, it is crucial to measure the model's performance on unseen test data to determine its generalization capabilities. The proportion of accurate predictions to all predictions is known as accuracy. The formula is in equation (6):

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \quad (6)$$

BER calculates the rate of misclassified bits as follows in equation (7):

$$BER = \frac{Number\ of\ Bit\ Errors}{Total\ Number\ of\ Bits} \quad (7)$$

By comparing the predicted labels with the true labels, a confusion matrix displays the classification model performance. It reveals the kinds of errors the model commits.

4.2. Comparison

The CNN model, DNN, and the existing CLDNN model are the three additional models. Tables 2, 3, and 4 describe the structure of layers in these existing models.

Table -2: Existing CLDNN Architecture

Layers	Input Parameters
Convolution (Conv2d)	Custom activation
MaxPooling2D	pool size=(1, 2)
Dropout (rate)	0.3
Conv2D	Custom activation
MaxPooling2D	pool size=(1, 2)

Dropout (rate)	0.3
Dense	128 hidden layer, Custom activation
Dropout (rate)	0.3
Reshape	(2, 4096)
LSTM	Custom activation
Dropout (rate)	0.3
Dense	128 hidden layer, Custom activation
Dropout (rate)	0.3
Dense	11 hidden layer, softmax

Table -3: Existing CNN Architecture

Layers	**Input Parameters**
Convolution (Conv2d)	Relu
BatchNormalization	-
MaxPooling2D	pool_size=(1, 2)
Dropout (rate)	0.3
Conv2D	Relu
Batch Normalization	-
MaxPooling2D	pool size= (1, 2)
Dropout (rate)	0.3
Convolution (Conv2d)	Relu
Batch Normalization	-
MaxPooling2D	pool_size=(1, 2)
Dropout (rate)	0.3
Convolution (Conv2d)	Relu
Batch Normalization	-
MaxPooling2D	pool size=(1, 2)
Dropout (rate)	0.3

Flatten	-
Dense	128 hidden layer, Relu
Batch Normalization	-
Dense	11 hidden layers, SoftMax

Table – 4: Existing DNN Architecture

Layers	Input Parameters
Dense	256 hidden layers, Relu
Dense	128 hidden layers, Relu
Dense	250 hidden layers, Relu
Dense	64 hidden layer, Relu
Flatten	-
Dense	128 hidden layer, Relu
Dense	11 hidden layer, softmax

Table - 5: Proposed accuracy, loss, validation loss and validation accuracy

SNR	accuracy	loss	Validation accuracy	Validation loss
18	0.7904	0.5046	0.7830	0.4861
16	0.7817	0.5139	0.7477	0.5929
14	0.7479	0.5763	0.7489	0.5711
12	0.8136	0.4134	0.8068	0.4545
10	0.8401	0.3632	0.8250	0.3794
8	0.8599	0.3313	0.8659	0.3181
6	0.8648	0.3347	0.8670	0.3167
4	0.8397	0.3553	0.8545	0.3300
2	0.8394	0.3685	0.8159	0.4162
0	0.7525	0.5406	0.7386	0.5129
-2	0.6950	0.6786	0.7239	0.6336

-4	0.5953	0.9196	0.6068	0.8932
-6	0.5152	1.1663	0.5432	1.0449
-8	0.3773	1.5435	0.4364	1.5010
-10	0.3133	1.8059	0.2966	1.8219
-12	0.0919	2.3979	0.0659	2.3992
-14	0.0995	2.3977	0.0795	2.3983
-16	0.0939	2.3979	0.0818	2.3984
-18	0.0961	2.3978	0.0784	2.3982
-20	0.0936	2.3978	0.0795	2.3993

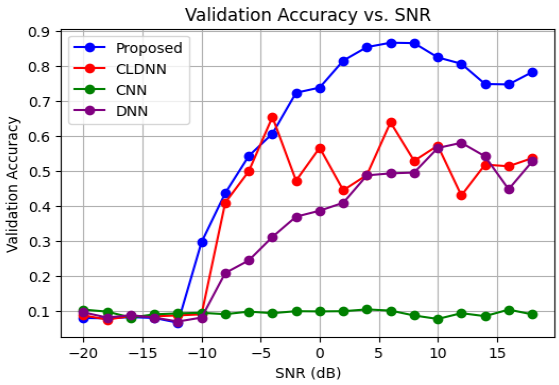


Chart -3: Validation Accuracy Comparison

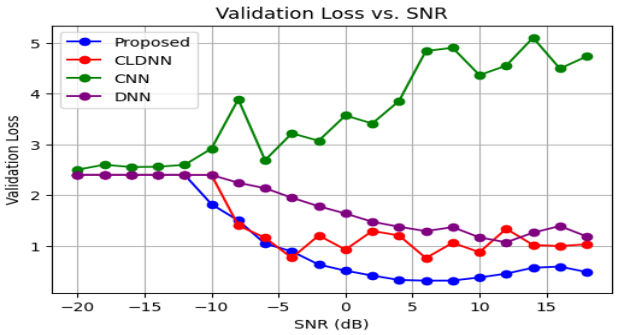


Chart – 4: Validation Loss Comparison

Table – 6: Proposed other performance measures

SNR	Signal Probability Detection	Bit Error Rate (BER)	Precision	Recall	F1score
18	0.7645454406738281	0.23545454545454544	0.77	0.76	0.76
16	0.7490909099578857	0.2509090909090909	0.75	0.75	0.74

14	0.7749999761581421	0.225	0.79	0.77	0.76
12	0.8122727274894714	0.18772727272727271	0.82	0.82	0.81
10	0.8322727084159851	0.16772727272727272	0.83	0.83	0.83
8	0.8450000286102295	0.155	0.85	0.85	0.85
6	0.8663636445999146	0.13363636363636364	0.87	0.87	0.86
4	0.8550000190734863	0.145	0.85	0.85	0.85
2	0.8345454335212708	0.16545454545454547	0.83	0.83	0.83
0	0.7549999952316284	0.245	0.76	0.76	0.75
-2	0.7077272534370422	0.2922727272727273	0.70	0.71	0.70
-4	0.6186363697052002	0.38136363636363635	0.59	0.61	0.59
-6	0.5199999809265137	0.48	0.54	0.53	0.51
-8	0.4140909016132355	0.5859090909090909	0.41	0.42	0.40
-10	0.34272727370262146	0.6572727272727272	0.29	0.34	0.30
-12	0.0886363610625267	0.9113636363636364	0.01	0.09	0.01
-14	0.08500000089406967	0.915	0.01	0.09	0.01
-16	0.08818181604146957	0.9118181818181819	0.01	0.09	0.01
-18	0.0822727307677269	0.9177272727272727	0.01	0.09	0.01
-20	0.08272727578878403	0.9172727272727272	0.01	0.09	0.01

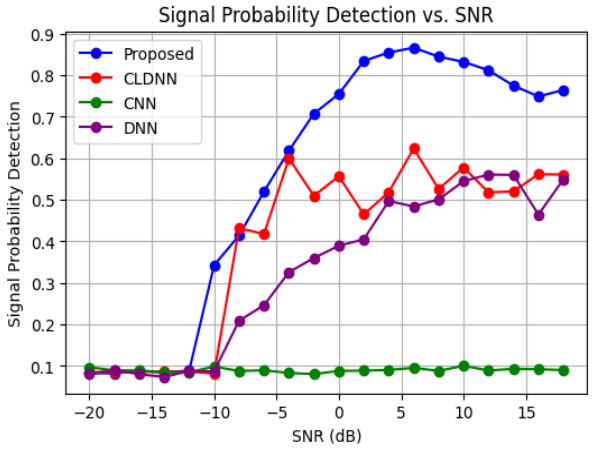


Chart - 5: Signal probability detection comparison

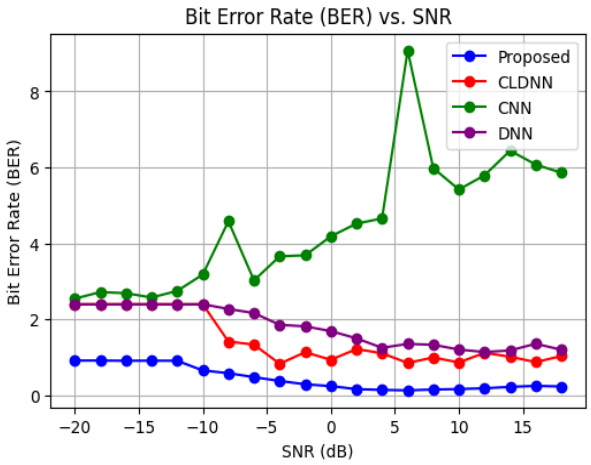


Chart – 6: BER comparison

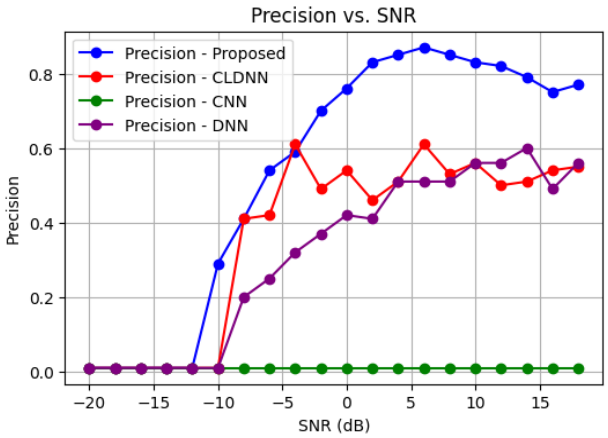


Chart – 7: Precision Comparison

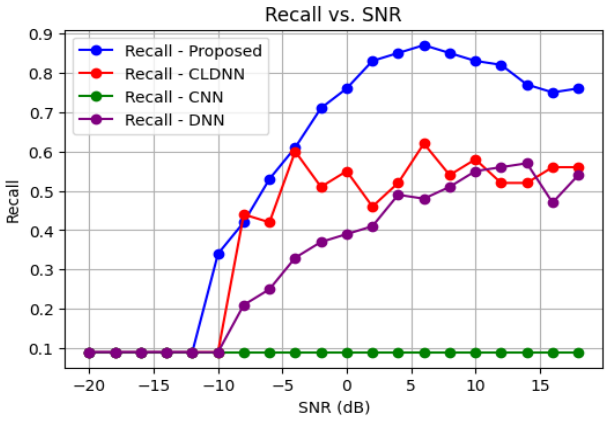


Chart – 8: Recall Comparison

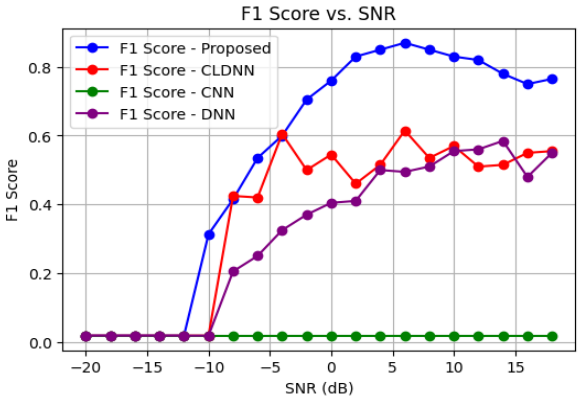


Chart – 9: F1score Comparison

The above figures from 3 to 9 show the evaluation results of several ML models using five important metrics, which include precision, recall, accuracy, F1 score, and BER. All the metric calculations are done for diverse SNR values in the dataset. The proposed model has the best accuracy of all the models at SNRs of 18, 16, and 14 dB. The proposed model gets 0.7830 accuracy at SNR = 18 dB and keeps that accuracy pretty high even as the noise level goes up, which is much better than CLDNN, CNN, and DNN. However, since it has such a challenging time dealing with noise, the CNN model consistently shows very poor accuracy, hovering around 0.1. The result shows that the proposed model is better at withstanding noise and also keeps its accuracy high even in difficult environments.

The generalizability of the proposed model is shown by using the validation loss metric. When comparing the existing CNN and DNN models, the proposed model shows a low and constant loss. While existing models show sharp lines in loss at higher SNR levels, the proposed model's loss is constant. At 18 dB SNR, the loss for the proposed model is 0.4861; however, CNN's loss is 4.7325. This suggests that CNN struggles with high SNR data, most likely because of its lack of generalizability. Thus, the proposed model outperforms other existing works, especially under settings with greater SNRs. The proposed model identifies signals with a probability of 0.7645 at 18 dB SNR. With lower SNR values, every model shows a reduction in detection likelihood as there is a rise in noise levels. But the proposed model maintains higher detection rates, which indicates high noise flexibility.

With a lower BER, the proposed model performs better since it assesses the rate of error bits in its predictions in comparison to the existing models. The proposed model has a BER of 0.235 for SNR = 18 dB, while CNN has a very high BER of 5.8593. This shows the proposed model's ability to convey more trustworthy data, without consideration of the noise levels. The performance metrics show the trade-off between correctly identifying positive samples (precision) and getting as many true positives. This is shown by the proposed model's high F1 scores in comparison to the existing CNN along with DNN. The proposed model is much better than CNN and DNN at dealing with unbalanced classes or noisy data. Its F1 score is around 0.77 at SNR = 18 dB, which is much higher than their scores of almost zero.

4.3. Discussion

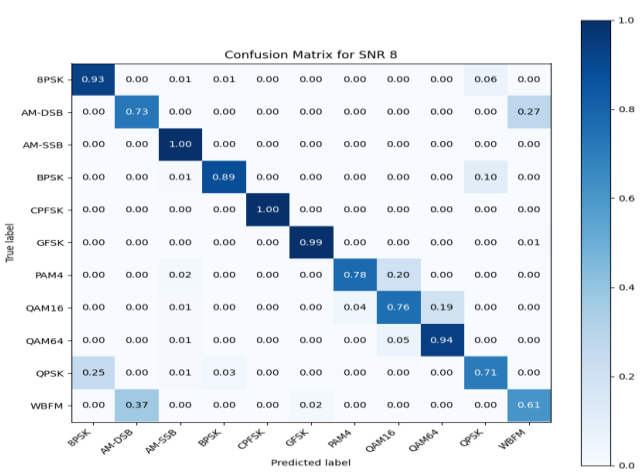
The proposed model shows better outcomes than the existing models (CLDNN, CNN, and DNN) on all important criteria. This advantage makes it the clear good in situations with different amounts of noise. Even at lower SNR levels, the proposed model keeps validation accuracy high, loss low, and signal detection efficient. The proposed model maintains a more constant performance at lower SNR levels, while CNN's performance declines significantly. This study shows that the proposed model has been modified to be more flexible to noise and is perfect for real-world applications where signal deterioration is prevalent. As the validation loss of the proposed model is low and constant over a range of SNR values, it appears to be able to generalize very well with new data. Alternatively, CNN and DNN models show a lot of loss when there's a lot of noise, which means they either overfit or don't generalize well. The proposed model is the preferred alternative when adapting to different noise situations.

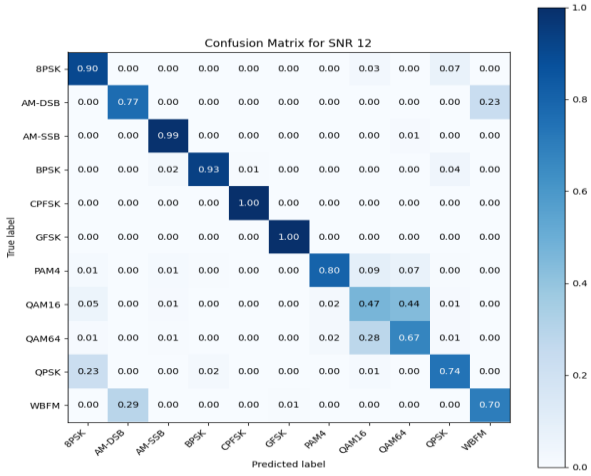
The proposed model does an improved job than the CNN and DNN models because it can be used repeatedly and is accurate at finding positive cases while reducing false positives and negatives. For activities where the proper identification of signals is

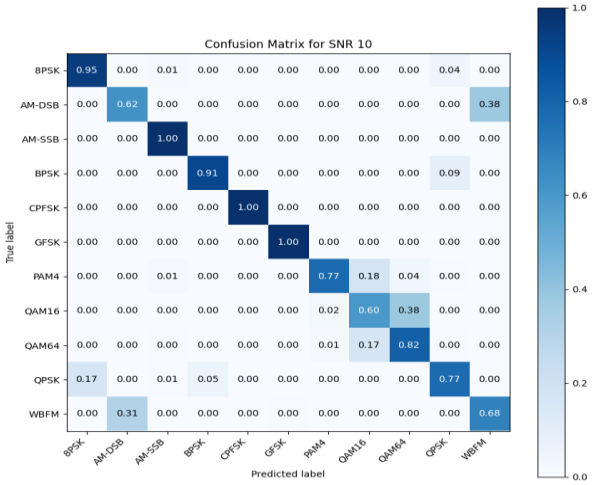
critical, such as in communication systems or detection tasks in noisy settings, this is essential. In communication systems where the integrity of the sent data is crucial, the proposed model consistently shows reduced bit error rates (BER). When compared to CNN and DNN, which have much higher BER values and are therefore less reliable, the proposed model has the unique ability to lower BER even when there is a lot of noise.

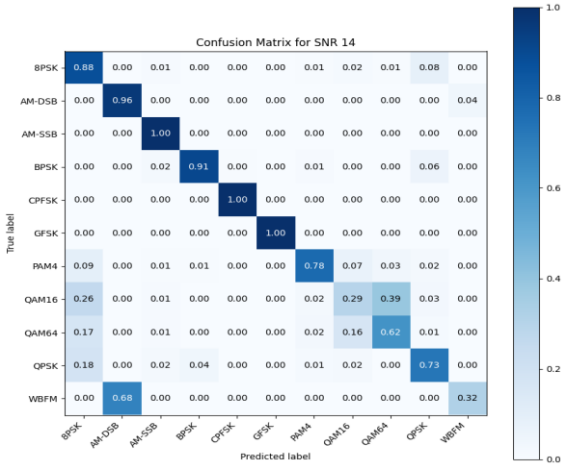
The proposed model does a good job with datasets that aren't balanced or noisy situations where both false positives as well as false negatives are important, as in the F1 score, which compares accuracy and recall. Because of its superior overall performance, the proposed model is more suited for real-world applications that need precision and dependability, as shown by its high F1 score. Due to its tolerance to noise, superior generalization, increased accuracy and recall, and decreased error rates, the proposed model surpasses the existing models. Its ability to handle different amounts of noise without sacrificing performance makes it a more dependable option for real-world applications.

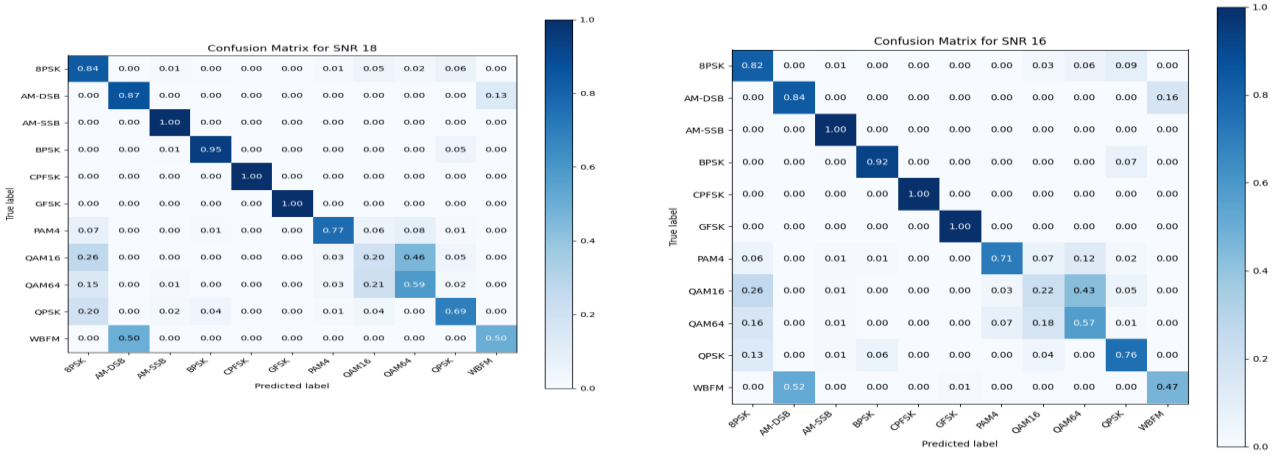
The hybrid model's superior results qualified to its ability to jointly exploit local patterns via CNNs and contextual dependencies via Transformer layers. This allows more robust modulation classification even under severe noise, unlike classical models which struggle with long-range dependencies.











5. Conclusion

This study introduced a hybrid CNN-Transformer model for AMR in wireless communication systems operating under varying noise conditions. The proposed model successfully overcomes the drawbacks of conventional CNNs and classical ML techniques, particularly in high SNR amplitude settings, by combining the long-range dependency modelling abilities of transformer architectures with the local feature extraction capabilities of convolutional layers. Its robustness is further demonstrated by the confusion matrix analysis, particularly in accurately classifying complex modulation types even in noisy environments. In summary, this proposed hybrid model, evaluated on a dataset comprising eleven modulation schemes across a wide SNR range, demonstrates improved classification accuracy, generalization, and noise robustness. Performance metrics, including confusion matrices and BER, confirm its superiority over baseline methods, particularly in differentiating closely spaced modulation types. Making it suitable for real-world applications such as CR, software-defined radios, and adaptive communication systems.

REFERENCES

[1] A. S. Jamwal and G. Kaur, “Cognitive Radio: An Emerging Trend for Better Spectrum Utilization,” IJCATR, vol. 2, no. 3, pp. 229–231, May 2013.

[2] R. Zhang, Y.-C. Liang, and S. Cui, “Dynamic Resource Allocation in Cognitive Radio Networks: A Convex Optimization Perspective,” IEEE Signal Processing Magazine, vol. 27, no. 3, pp. 102–114, May 2010.

[3] M. U. Muzaffar and R. Sharqi, “A Review of Spectrum Sensing in Modern Cognitive Radio Networks,” Telecommunication Systems, vol. 85, no. 2, pp. 347–363, Feb. 2024.

[4] E. M. Ali, G. M. Salama, K. A. A., and M. Ezz-Eldin, “Automatic Modulation Classification for Enhanced Cognitive Radio for IoT Systems Based on Deep Learning,” Journal of Advanced Engineering Trends, vol. 44, no. 1, pp. 0–0, Jan. 2025.

[5] B. Tang, Y. Tu, Z. Zhang, and Y. Lin, “Digital Signal Modulation Classification with Data Augmentation Using Generative Adversarial Nets in Cognitive Radio Networks,” IEEE Access, vol. 6, pp. 15713–15722, 2018.

[6] Q. Zheng, X. Tian, L. Yu, A. Elhanashi, and S. Saponara, “Recent Advances in Automatic Modulation Classification Technology: Methods, Results, and Prospects,” International Journal of Intelligent Systems, vol. 2025, no. 1, p. 4067323, Jan. 2025.

[7] B. Jdid, K. Hassan, I. Dayoub, W. H. Lim, and M. Mokayef, “Machine Learning-Based Automatic Modulation Recognition for Wireless Communications: A Comprehensive Survey,” IEEE Access, vol. 9, pp. 57851–57873, 2021.

[8] X. Liu, C. J. Li, C. T. Jin, and P. H. W. Leong, “Wireless Signal Representation Techniques for Automatic Modulation Classification,” IEEE Access, vol. 10, pp. 84166–84187, 2022.

[9] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, “A Survey of Automatic Modulation Classification Techniques: Classical Approaches and New Trends.”

[10] A. R., L. C., C. C., J. C. W. A., and A. B. R., "Modulation Classification in Cognitive Radio," in Foundations of Cognitive Radio Systems, S. Cheng, Ed. InTech, 2012.

[11] K. Tekbıyık, A. R. Ekti, A. Görçin, G. K. Kurt, and C. Keçeci, "Robust and Fast Automatic Modulation Classification with CNN under Multipath Fading Channels," in Proc. IEEE 91st Vehicular Technology Conference (VTC-Spring), May 2020, pp. 1–6.

[12] S. Huang et al., "Generalized Automatic Modulation Classification for OFDM Systems under Unseen Synthetic Channels," IEEE Transactions on Wireless Communications, vol. 23, no. 9, pp. 11931–11941, Sep. 2024.

[13] H. Zhang, F. Zhou, H. Du, Q. Wu, and C. Yuen, "Revolution of Wireless Signal Recognition for 6G: Recent Advances, Challenges and Future Directions," arXiv preprint, arXiv:2503.08091, Mar. 2025.

[14] M. C. Park and D. S. Han, "Deep Learning-Based Automatic Modulation Classification with Blind OFDM Parameter Estimation," IEEE Access, vol. 9, pp. 108305–108317, 2021.

[15] S. K. Jagatheesaperumal, I. Ahmad, M. Höyhtyä, S. Khan, and A. Gurtov, "Deep Learning Frameworks for Cognitive Radio Networks: Review and Open Research Challenges," arXiv preprint, arXiv:2410.23949, Oct. 2024.

[16] E. Vijay and K. Aparna, "RNN-BIRNN-LSTM Based Spectrum Sensing for Proficient Data Transmission in Cognitive Radio," e-Prime – Advances in Electrical Engineering, Electronics and Energy, vol. 6, p. 100378, Dec. 2023.

[17] T. Xu and Y. Ma, "Signal Automatic Modulation Classification and Recognition in View of Deep Learning," IEEE Access, vol. 11, pp. 114623–114637, 2023.

[18] T. Zhang, C. Shuai, and Y. Zhou, "Deep Learning for Robust Automatic Modulation Recognition Method for IoT Applications," IEEE Access, vol. 8, pp. 117689–117697, 2020.

[19] B. Xu et al., "Towards Explainability for AI-Based Edge Wireless Signal Automatic Modulation Classification," Journal of Cloud Computing, vol. 13, no. 1, p. 10, Jan. 2024.

[20] Y. Wang et al., "An Improved Modulation Recognition Algorithm Based on Fine-Tuning and Feature Re-Extraction," Electronics, vol. 12, no. 9, p. 2134, May 2023, doi: 10.3390/electronics12092134.

[21] S. Mohsen, A. M. Ali, and A. Emam, "Automatic modulation recognition using CNN deep learning models," Multimed Tools Appl, vol. 83, no. 3, pp. 7035–7056, Jan. 2024, doi: 10.1007/s11042-023-15814-y.

[22] Z. Elkhatib, F. Kamalov, S. Moussa, A. B. Mnaouer, M. C. E. Yagoub, and H. Yanikomeroglu, "Radio Modulation Classification Optimization Using Combinatorial Deep Learning Technique," IEEE Access, vol. 12, pp. 17552–17570, 2024, doi: 10.1109/ACCESS.2024.3357628.

[23] Y. Qu, Z. Lu, R. Zeng, J. Wang, and J. Wang, "Enhancing Automatic Modulation Recognition through Robust Global Feature Extraction," arXiv preprint, arXiv:2401.01056, Jan. 2024.

[24] J. D. Ruikar, D.-H. Park, S.-Y. Kwon, and H.-N. Kim, "HCTC: Hybrid Convolutional Transformer Classifier for Automatic Modulation Recognition," Electronics, vol. 13, no. 19, p. 3969, Oct. 2024, doi: 10.3390/electronics13193969.

[25] Q. Zheng et al., "MobileRaT: A Lightweight Radio Transformer Method for Automatic Modulation Classification in Drone Communication Systems," Drones, vol. 7, no. 10, p. 596, Sep. 2023, doi: 10.3390/drones7100596.

[26] A. Faysal, M. Rostami, R. G. Roshan, H. Wang, and N. Muralidhar, "NM former: A Transformer for Noisy Modulation Classification in Wireless Communication," Oct. 30, 2024, arXiv: arXiv: 2411.02428.doi: 10.48550/arXiv.2411.02428.