

Article

Performance Analysis of a Hybrid Complex-Valued CNN-TCN Model for Automatic Modulation Recognition in Wireless Communication Systems

Hamza Ouamna ^{*,†}, Anass Kharbouche [†], Nouredine El-Haryqy [†], Zhou Madini [†] and Younes Zouine [†]

Electrical and Telecommunication Department, Laboratory of Advanced Systems Engineering (ISA), National School of Applied Sciences (ENSA), Ibn Tofail University, Kenitra 14000, Morocco; anass.kharbouche@uit.ac.ma (A.K.); noureddine.elharyqy@uit.ac.ma (N.E.-H.); zhour.madini@uit.ac.ma (Z.M.); younes.zouine@uit.ac.ma (Y.Z.)

* Correspondence: hamza.ouamna@uit.ac.ma

[†] These authors contributed equally to this work.

Abstract

This paper presents a novel deep learning-based automatic modulation recognition (AMR) model, designed to classify ten modulation types from complex I/Q signal data. The proposed architecture, named CV-CNN-TCN, integrates Complex-Valued Convolutional Neural Networks (CV-CNNs) with Temporal Convolutional Networks (TCNs) to jointly extract spatial and temporal features while preserving the inherent phase information of the signal. An enhanced variant, CV-CNN-TCN-DCC, incorporates dilated causal convolutions to further strengthen temporal representation. The models are trained and evaluated on the benchmark RadioML2016.10b dataset. At SNR = −10 dB, the CV-CNN-TCN achieves a classification accuracy of 37%, while the CV-CNN-TCN-DCC improves to 40%. In comparison, ResNet reaches 33%, and other models such as CLDNN (convolutional LSTM dense neural network) and SCRNN (Sequential Convolutional Recurrent Neural Network) remain below 30%. At 0 dB SNR, the CV-CNN-TCN-DCC achieves a Jaccard index of 0.58 and an MCC of 0.67, outperforming ResNet (0.55, 0.64) and CNN (0.53, 0.61). Furthermore, the CV-CNN-TCN-DCC achieves 75% accuracy at SNR = 10 dB and maintains over 90% classification accuracy for SNRs above 2 dB. These results demonstrate that the proposed architectures, particularly with dilated causal convolutional enhancements, significantly improve robustness and generalization under low-SNR conditions, outperforming state-of-the-art models in both accuracy and reliability.

Keywords: deep learning; I/Q signals; automatic modulation classification; complex-valued; CNN; Temporal Convolutional Network



Academic Editor: Daswin De Silva

Received: 6 April 2025

Revised: 6 June 2025

Accepted: 11 June 2025

Published: 28 June 2025

Citation: Ouamna, H.; Kharbouche, A.; El-Haryqy, N.; Madini, Z.; Zouine, Y. Performance Analysis of a Hybrid Complex-Valued CNN-TCN Model for Automatic Modulation Recognition in Wireless Communication Systems.

Appl. Syst. Innov. **2025**, *8*, 90. <https://doi.org/10.3390/asi8040090>

Copyright: © 2025 by the authors.

Published by MDPI on behalf of the International Institute of Knowledge Innovation and Invention. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

AMC identifies modulation schemes without prior knowledge of parameters like channel state or transmitter specifics. Traditional methods split into likelihood-based (LB-AMC) [1] and feature-based (FB-AMC) [2] approaches. LB-AMC methods rely on likelihood ratios to estimate modulation schemes but can be hampered by dependencies on prior knowledge. FB-AMC methods use extracted signal features such as time–frequency [3] properties or cyclostationary properties [4] or high-order cumulants [5] for classification, relying heavily on manual feature engineering, which can be a limitation in dynamic environments.

Deep learning, as a leading machine learning approach, has gained widespread use in automatic modulation classification due to its ability to learn features directly from raw data without manual feature engineering. This approach has proven especially advantageous for handling complex signal patterns in AMC applications [6–8]. One landmark study by O’Shea et al. [9], demonstrated that convolutional neural networks (CNNs) can outperform traditional approaches on benchmark datasets [10]. Other advanced models, such as the convolutional LSTM dense neural network (CLDNN) and ResNet, have achieved high accuracy by leveraging temporal and spatial dependencies within signal data [11].

Given that modulated signal samples form a time-series, a two-layer long short-term memory (LSTM) model has been introduced, utilizing amplitude and phase as inputs to capture temporal dependencies effectively. This approach enhances classification performance by capturing the time-series nature of signals to improve modulation recognition accuracy, especially under dynamic signal conditions [12]. In [13], the authors introduced DLRNet, which employs depthwise convolution combined with large kernels to improve feature extraction capabilities, demonstrating significant gains in recognition accuracy. In [14], the authors utilized the short-time Fourier transform (STFT) to transform time-series data into spectral images, which were then used as input for a CNN [15]. This approach resulted in high classification accuracy, demonstrating the effectiveness of converting temporal data into a frequency domain representation for improved signal recognition. In [16], the authors introduced a lightweight network that achieves high recognition accuracy while maintaining low computational requirements. This model relies on the estimation and conversion of phase parameters, effectively balancing performance and efficiency in modulation recognition tasks. In [17], Zhang et al. developed a method called DPM-SCNN that enhances sequence feature representation by implementing cyclic shift processing on time-series data. This approach has demonstrated high recognition accuracy on benchmark datasets.

Building on these, Sun et al. improved AMC by introducing a bidirectional gated cycle unit in the CLDNN, which balances accuracy and speed, making it suitable for real-time applications [18]. Zhang et al. further refined this by integrating Bi-LSTM layers with CNNs, leading to a model that excels in environments with noise interference [19]. Innovative methods continue to emerge. Meta-learning frameworks, like the meta-transformer, have also made strides by enabling AMC models to learn generalizable patterns from limited samples, promising improved performance on previously unseen modulation types [20]. Newer strategies address noise resilience. Qi’s Waveform Spectrum Multimode Fusion combines residual networks with multimodal fusion, which is highly effective in environments with high-order digital modulations [21].

1.1. Existing Solutions and Research Gaps

1.1.1. Existing Solutions

Automatic modulation classification research has driven the development of several notable solutions aimed at improving the recognition of modulation schemes in wireless communication systems. Below, we summarize some of the key contributions:

- **Traditional Techniques:** Early approaches like maximum likelihood (ML) methods, decision-theoretic approaches, and feature-based classifiers have been extensively studied. These methods utilize signal characteristics such as amplitude, phase, and frequency to distinguish modulation schemes with varying degrees of success.
- **Signal Processing-Based Methods:** Advanced signal processing techniques, such as wavelet transforms, cyclostationary feature detection, and higher-order cumulants, have been applied to improve the accuracy and robustness of AMC in diverse wireless environments.

- **Machine Learning Approaches:** Recent developments have focused on leveraging supervised learning algorithms, such as support vector machines (SVM) and k-nearest neighbors (k-NN), to classify modulation schemes. These models often rely on pre-extracted features and have shown promise in controlled scenarios.
- **Deep Learning Solutions:** The integration of deep learning models, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), has revolutionized AMC by enabling end-to-end learning. These models process raw I/Q data directly, eliminating the need for manual feature extraction, and have demonstrated superior performance in complex environments.

1.1.2. Major Research Gaps

Despite progress in cognitive radio networks (CRNs), several key research gaps remain to fully harness their potential. The following sections outline some of the current challenges in the field of cognitive radio that require further investigation:

- **Performance in Low-SNR Environments:** While many techniques achieve high accuracy under ideal conditions, their performance often degrades significantly in low-SNR environments. Enhancing robustness in these challenging conditions remains an open research area.
- **Robustness Against Noise and Interference:** AMC systems must reliably classify signals under noise uncertainty and interference. More robust architectures or noise-resistant features are needed for better performance in noisy environments.
- **Interference from Dynamic Environments:** Existing methods often assume static environments, which is unrealistic. Methods that can adapt to dynamic conditions, such as varying channel conditions and interference, are necessary.

These gaps present opportunities to enhance automatic modulation classification methods, focusing on improving their practicality, efficiency, and reliability for deployment in real-world cognitive radio systems. Advancing these techniques can address current limitations, enabling more robust performance under varying signal conditions and dynamic environments.

1.2. Proposal of This Paper and Its Significance

This work centers on enhancing automatic modulation classification using a hybrid deep learning model integrating CNN and TCN to address the complexity of dynamic wireless signals. CNNs excel in capturing local features and signal patterns by applying convolutional filters across time-series data, making them effective at recognizing modulation types under various noise levels. The TCNs complement CNNs by modeling longer temporal dependencies without losing context, essential for accurately classifying signal modulations in rapidly fluctuating environments. Together, CNN-TCN architectures adaptively enhance classification reliability, even in low-SNR conditions.

1.3. Research Questions

This study aims to answer the following research questions (RQs):

RQ1: How does a hybrid CNN-TCN architecture improve automatic modulation classification accuracy compared to standalone CNN or TCN models?

RQ2: What are the contributions of CNN and TCN components in handling local signal patterns and long-term temporal dependencies, respectively, in dynamic wireless environments?

RQ3: How robust is the proposed CNN-TCN model under varying signal-to-noise ratio (SNR) conditions, particularly in low-SNR scenarios?

RQ4: What is the computational efficiency of the CNN-TCN model in real-time signal processing applications, and how does it compare to other deep learning-based AMC methods?

RQ5: How does the proposed model perform in comparison to state-of-the-art AMC techniques (e.g., CNN, ResNet, LSTM, or CLDNN) in terms of classification accuracy and robustness?

2. Related Works

In [22], the authors used a phase-based short-time Fourier transform (STFT) to analyze radar signals and extract time–frequency features that emphasize phase characteristics. These features were then processed with a bidirectional long short-term memory (BiLSTM) network, which could model dependencies in both forward and backward time directions. The approach performed well, especially in low-SNR scenarios, showing how phase-based STFT combined with BiLSTM can effectively classify radar signals.

In [23], the authors proposed a method for modulation classification that transforms received signals into signal constellation diagrams, representing key modulation characteristics visually. These diagrams were analyzed using a convolutional neural network (CNN), which extracted features automatically for classification. The approach showed high accuracy, even in noisy and low-SNR conditions, highlighting the practical benefits of using constellation diagrams and CNNs for signal recognition.

In [24], the authors introduced an LSTM-guided modulation classification method for sub-Nyquist rate wideband spectrum sensing. The approach utilizes long short-term memory (LSTM) networks to classify modulations in signals sampled at sub-Nyquist rates, making it more efficient for wideband-spectrum-sensing applications. The approach was validated through experiments, showing strong performance in terms of both classification accuracy and computational efficiency.

In [25], the authors proposed MCNet, a lightweight and efficient CNN architecture for automatic modulation classification. The model is designed to handle challenging channel conditions and noise levels while keeping computational requirements low. By focusing on effective feature extraction, MCNet achieves strong classification performance with reduced complexity.

A recent study in [13] proposes an automatic modulation recognition (AMR) method for radiation source signals that leverages a two-dimensional data matrix and an improved residual neural network. The approach involves two key components: (1) a preprocessing technique that reconstructs intercepted time-series signals into 2D matrices to enhance task autonomy and simplify preprocessing, and (2) a deep residual network enhanced with depthwise convolutions and large convolutional kernels to improve feature extraction and recognition performance.

The authors in [26] propose a deep convolutional neural network (CNN) architecture enhanced with a multi-task learning (MTL) scheme for automatic modulation recognition. The MTL framework is designed to address common classification challenges by training on related tasks simultaneously, particularly focusing on resolving confusion between similar modulation types.

The study [27] proposes CGDNet, a recurrent fully connected deep neural network that combines a shallow CNN for spatial feature extraction, a GRU for modeling temporal dependencies, and a DNN for classification, effectively balancing accuracy and computational efficiency. Further improving robustness in low-SNR environments, Ref. [28] introduces a transformer-based semi-supervised framework leveraging contrastive learning and data augmentation (e.g., time warping), where a convolutional transformer encoder

jointly learns local and global signal features, achieving state-of-the-art performance on RML2016 benchmarks.

3. Mathematical Concept and Modeling

3.1. Mathematical Model of CV-CNN-TCN

3.1.1. Input Representation

Let the input data $\mathbf{X}$ be a complex-valued signal, represented as

$$\mathbf{X} = \mathbf{X}_R + j\mathbf{X}_I, \mathbf{X}_R, \mathbf{X}_I \in \mathbb{R}^{T \times d} \quad (1)$$

where $\mathbf{X}_R$ and $\mathbf{X}_I$ are the real and imaginary components of the input, respectively, T is the number of timesteps, and d is the feature dimensionality.

3.1.2. Complex-Valued Convolutional Neural Network (CV-CNN)

The CV-CNN extracts spatial and spectral features from the complex input. For a convolutional layer, let $\mathbf{W} = \mathbf{W}_R + j\mathbf{W}_I$ be the complex-valued filter weights, and $\mathbf{b} = \mathbf{b}_R + j\mathbf{b}_I$ be the biases. The convolution operation is given by

$$\mathbf{H} = f(\mathbf{W} * \mathbf{X} + \mathbf{b}) \quad (2)$$

where $*$ denotes the convolution operation, and $f(\cdot)$ is a complex activation function, such as the complex-modulus activation:

$$f(\mathbf{Z}) = \sqrt{\mathbf{Z}_R^2 + \mathbf{Z}_I^2} \quad (3)$$

Expanding the convolution into real and imaginary parts:

$$\mathbf{H}_R = f(\mathbf{W}_R * \mathbf{X}_R + \mathbf{W}_I * \mathbf{X}_I + \mathbf{b}_R), \quad (4)$$

$$\mathbf{H}_I = f(\mathbf{W}_R * \mathbf{X}_I + \mathbf{W}_I * \mathbf{X}_R + \mathbf{b}_I). \quad (5)$$

The output of the k -th convolutional layer is the complex feature map:

$$\mathbf{H}_k = \mathbf{H}_R^k + j\mathbf{H}_I^k, k = 1, 2, \dots, K. \quad (6)$$

3.1.3. Temporal Convolutional Network (TCN)

The TCN captures temporal dependencies in the features extracted by the CV-CNN. The input to the TCN is the output of the CV-CNN, $\mathbf{H}_{CNN} \in \mathbb{C}^{T' \times d'}$, where T' and d' are determined by the CNN architecture.

For a TCN layer with dilated convolutions, the operation is expressed as

$$\mathbf{Z}_l = f(\mathbf{W}_l *_d \mathbf{Z}_{l-1} + \mathbf{b}_l), l = 1, 2, \dots, L. \quad (7)$$

where $*_d$ denotes dilated convolution with dilation factor d_l , and $f(\cdot)$ is a nonlinear activation function, Figure 1 illustrates a visualization of a stack of dilated causal convolutional layers. The complex convolution is performed in a manner similar to the CV-CNN, processing real and imaginary parts separately.

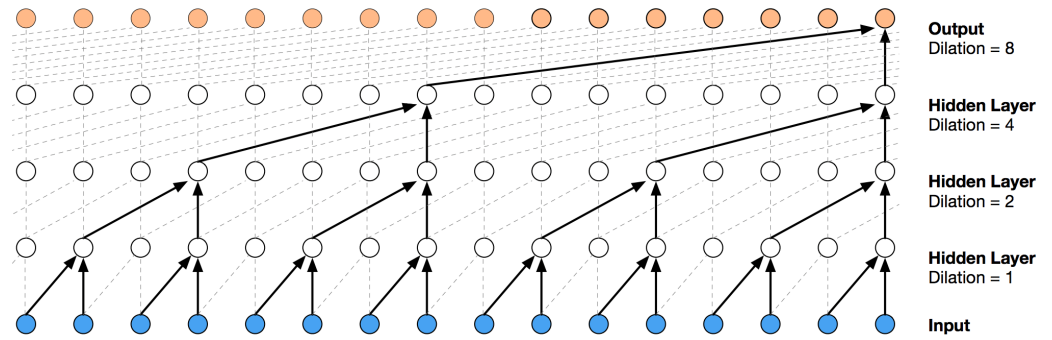


Figure 1. Visualization of a stack of dilated causal convolutional layers (Wavenet, 2016) reprinted from ref. [29].

3.1.4. Fully Connected Layers

The TCN output $\mathbf{Z}_{TCN} \in \mathbb{C}^{T' \times d'}$ is flattened or aggregated (e.g., using global average pooling) into a vector $\mathbf{z} \in \mathbb{C}^q$, which is passed through fully connected layers. Each dense layer operation is

$$\mathbf{y} = \sigma(\mathbf{W}_{fc}\mathbf{z} + \mathbf{b}_{fc}), \quad (8)$$

where $\mathbf{W}_{fc}$ and $\mathbf{b}_{fc}$ are complex-valued weights and biases, $\sigma(\cdot)$ is the activation function (e.g., softmax for classification), and $\mathbf{y} \in \mathbb{R}^C$ is the final output, with C as the number of classes. We consider a classification task with $C = 10$ classes, corresponding to the following modulation types: QPSK, PAM4, 8PSK, BPSK, CPFSK, BFSK, QAM64, QAM16, AM-DSB, and WB-FM.

3.1.5. Loss Function

For classification tasks, the loss function is typically a categorical cross-entropy, computed using the magnitude of the output probabilities:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_i^c \log|\hat{y}_i^c|, \quad (9)$$

where N is the number of training samples, y_i^c is the ground truth, and $\hat{y}_i^c$ is the predicted complex-valued probability for class C for the i -th sample.

3.2. Baseline Methods

In this paper, the proposed model performance was compared with four prior AMC models.

3.2.1. CNN

In [30], the author proposed a CNN architecture designed for automatic modulation classification. The architecture comprises four convolutional layers followed by two fully connected layers. ReLU is used as the activation function for all hidden layers, while the output layer features ten neurons with a SoftMax activation function to provide the modulation classification results.

3.2.2. CLDNN

The CLDNN baseline network, proposed in [31], is an architecture comprising three convolutional layers, followed by one LSTM layer and two fully connected layers. The output layer, a dense layer with ten neurons, uses the SoftMax activation function for classification. ReLU serves as the activation function for all hidden layers. Additionally, each

convolutional layer is followed by an average pooling layer to reduce the dimensionality of the feature maps.

3.2.3. SCRNN

The SCRNN model, introduced in [32], combines the strengths of CNNs and LSTMs by taking the efficiency of CNNs for feature extraction and the temporal sensitivity of LSTMs. The architecture includes two convolutional layers, followed by two LSTM layers and a dense layer. The final dense layer, equipped with ten neurons and a SoftMax activation function, produces the classification output. To reduce dimensionality, a 1×3 max-pooling layer is applied after the first convolutional layer.

3.2.4. ResNet

The ResNet baseline model proposed in [30] consists of three residual stacks and three dense layers. Each residual stack includes a convolutional layer, two residual units, and a max-pooling layer for dimensionality reduction. A residual unit comprises two convolutional layers with 1×5 kernels, ReLU activation, and batch normalization. A shortcut connection bypasses the two convolutional layers, directly connecting the input of the residual unit to the output of the second convolutional layer. The combined result is then processed through a ReLU activation function. The first two dense layers in the architecture each contain 128 units with ReLU activation, while the output layer consists of ten neurons with a SoftMax activation function for classification.

3.3. Data Preprocessing and Sampling

In this study, we used the RML2016.10b dataset [33], a well-established benchmark in modulation recognition research, generated using the GNU Radio v3.10.7.0 software-defined radios toolkit. This dataset includes 10 modulation techniques, 8 of which are digital (QPSK, BPSK, BFSK, 8PSK, CPFSK, PAM-4, QAM64, QAM16), and 2 analog (AM-DSB, WBFM). The data are evenly distributed among all modulation types, comprising 1,200,000 samples, each represented in IQ format with a size of 2×128 . The labels for each sample include modulation type and SNR value, with SNRs ranging from -20 dB to 18 dB in 2 dB increments. A summary of the dataset is presented in Table 1.

Table 1. Dataset summary.

Parameter	Value
Modulations	QPSK, PAM4, 8PSK, BPSK, CPFSK, BFSK, QAM64, QAM16, AM-DSB, WB-FM
Samples	1,200,000
Sampling Frequency	1 Msample/s
Sampling Interval	128 μ s
Samples/Symbol	8 samples/symbol
Format	IQ
Sample size	2×128
SNR number	20 values
SNR range	$[-20$ dB, 18 dB]
Training data	960,000 samples (80%)
Testing data	240,000 samples (20%)
Labels	SNR and modulation methods

For training and evaluation, the dataset was divided into 960,000 samples for training and 240,000 samples for testing as described in Figure 2, with 15% of the training data used for validation. To accommodate the complex-valued network, real numbers were

converted into complex numbers, reducing the input dimension from 2×128 to 1×128 , while the real-valued network retained the 2×128 format.

The training was performed using the Keras library on Google Colab [34], with the Adam optimizer, a learning rate of 0.0001, a batch size of 256, and a cross-entropy loss function. The training process lasted for 20 epochs, with early stopping implemented to prevent overfitting. Algorithm 1 presents the training process of the proposed CV-CNN-TCN model.

Algorithm 1 Complex-Valued CNN-Temporal Convolutional Network Training

Input:

- Complex-valued signals $\mathbf{X} = \mathbf{X}_R + j\mathbf{X}_I \in \mathbb{C}^{T \times d}$
- Ground truth labels $y \in \mathbb{R}^C$ for C modulation types
- T : Number of training epochs
- B : Batch size
- η : Learning rate

Output: Trained model

1. Initialize complex-valued parameters:
 - CV-CNN filters $\mathbf{W}_k = \mathbf{W}_{R,k} + j\mathbf{W}_{I,k}$ for $k = 1, \dots, K$
 - TCN dilated filters $\mathbf{W}_l = \mathbf{W}_{R,l} + j\mathbf{W}_{I,l}$ for $l = 1, \dots, L$
 - FC layer weights $\mathbf{W}_{fc} = \mathbf{W}_{R,fc} + j\mathbf{W}_{I,fc}$
 2. For epoch $t = 1$ to T do:
 - (a) For batch $b = 1$ to B do:
 - i. **Forward propagation:**
 - CV-CNN feature extraction:
 - Compute real/imaginary convolutions: $\mathbf{H}_R^k, \mathbf{H}_I^k$ for each layer k
 - Apply complex-modulus activation
 - TCN temporal processing:
 - Perform dilated convolutions on $\mathbf{H}_{CNN}$
 - Process real/imaginary components separately
 - Classification head:
 - Flatten TCN output to $\mathbf{z} \in \mathbb{C}^q$
 - Compute class probabilities via softmax
 - ii. **Loss computation:**
 - Calculate cross-entropy using magnitude of predictions
 - $\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_i^c \log |\hat{y}_i^c|$
 - iii. **Backward propagation:**
 - Compute gradients through complex-valued layers
 - Update parameters using complex-aware optimizer:
 - $\mathbf{W} \leftarrow \mathbf{W} - \eta \nabla_{\mathbf{W}} \mathcal{L}$
 - Maintain real/imaginary component relationships
 - (b) End for
 3. End for
 4. Return trained model:
-

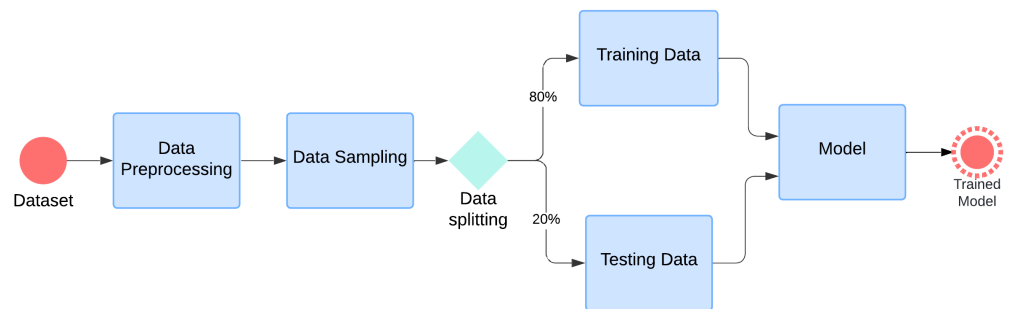


Figure 2. Algorithmic diagram depicting training and testing pipelines.

3.4. Proposed Model: CV-CNN-TCN

The proposed architecture, a Complex-Valued CNN-TCN, was designed to process IQ data directly as complex-valued inputs, leveraging their inherent structure for effective feature extraction and classification. The input layer accepts data in the form of 2×128 , representing the in-phase (I) and quadrature (Q) components, which are transformed into a complex representation to preserve the spatial and temporal correlations. The architecture incorporates two parallel convolutional pathways, each comprising 32 filters with a kernel size of 3, applied independently to the I and Q components. This design facilitates the capture of unique features from each component while ensuring nonlinear patterns are extracted using ReLU activation. Max-pooling layers with a size of 2 follow the convolutions to reduce dimensions and emphasize dominant features. The outputs of the convolutional blocks are flattened and concatenated, forming a unified feature representation. These features are then passed through a TCN block, which captures long-range dependencies in the data, enabling the model to learn temporal relationships critical for classification tasks. The architecture concludes with dense layers, employing softmax activation for classification. An overview of the proposed model is illustrated in Figure 3.

3.5. Performance Metrics

The performance of the proposed method is evaluated using several metrics, including classification accuracy, sensing error (SE), training accuracy, validation loss, validation accuracy, Matthew's correlation coefficient (MCC), F1 score, Jaccard index (JI), noise value prediction (NPV), and false discovery rate (FDR) [35–37]. Training accuracy measures the AMC model's performance on the training dataset, reflecting how well the model classifies or predicts modulation types in training samples. Validation loss quantifies the error between the predicted and target modulation types in the validation dataset, while validation accuracy represents the percentage of correctly classified instances in the validation set [38,39]. The MCC metric, ranging from -1 to $+1$, assesses the effectiveness of binary classification models in AMC: $+1$ indicates perfect predictions, 0 suggests random guessing, and -1 implies completely incorrect predictions. The F1 score, which combines precision and recall, ranges from 0 to 1 , with values close to 1 indicating better classification performance. The Jaccard index (JI) compares the similarity between predicted and actual modulation types, where 0 denotes complete similarity, and 1 indicates no similarity. The negative predictive value (NPV) measures the proportion of true negatives out of all negative predictions, indicating how often the model correctly identifies negative instances. The false discovery rate (FDR), on the other hand, represents the proportion of false positives out of all positive predictions, providing insight into the model's tendency to incorrectly label instances as positive [40]. Sensing error (SE) in AMC refers to detection inaccuracies that may lead to incorrect modulation classifications and is calculated as the average of the miss detection probability (P_{md}) and false alarm probability (P_{fa}) [41,42].

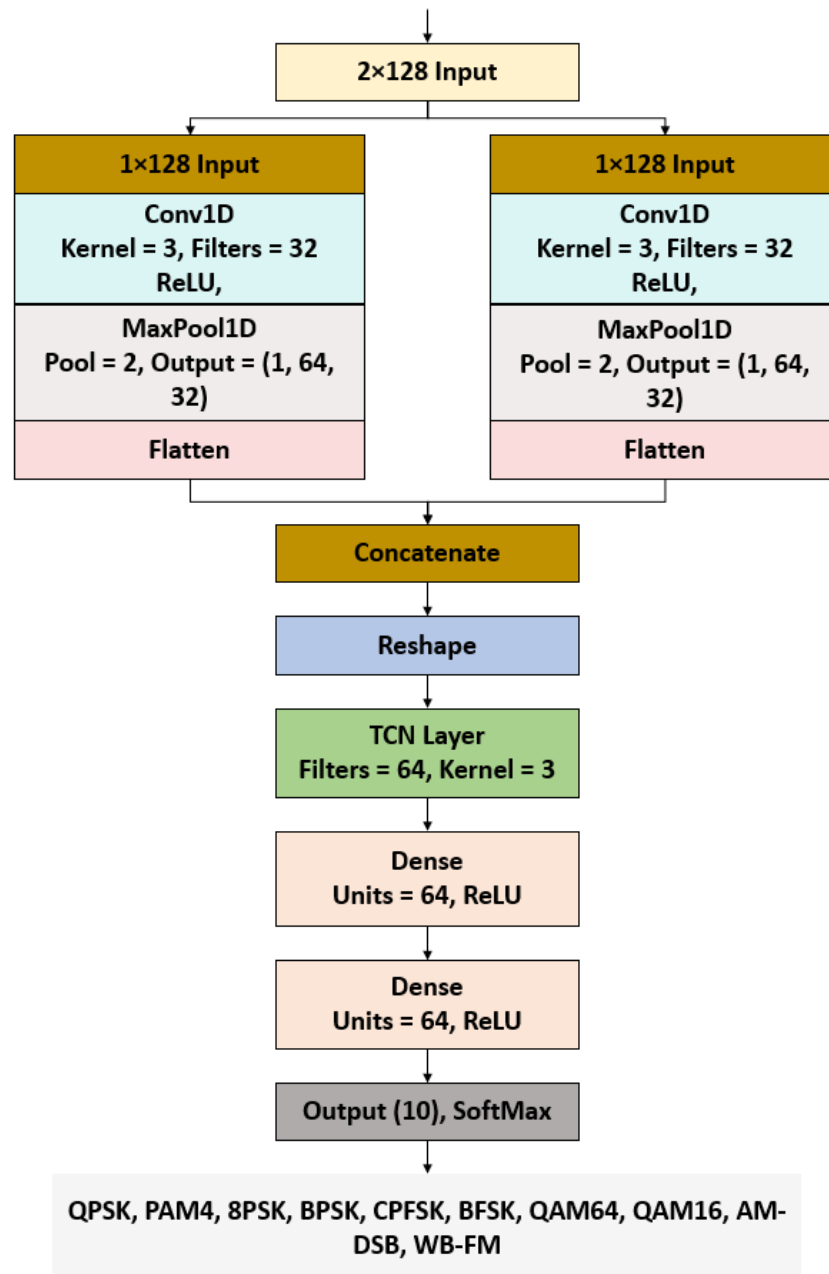


Figure 3. The proposed model architecture.

4. Results

4.1. Training and Validation Loss

Figure 4a illustrates the training accuracy across epochs. It is evident that the CV-CNN-TCN-DCC model achieves the highest accuracy throughout the training process (55.16%), closely followed by the CV-CNN-TCN (55.09%) and CNN (54.82%) models, while SCRNN (49.42%) and ResNet (51.72%) lag behind. Figure 4b depicts the corresponding training loss, where the CV-CNN-TCN-DCC model demonstrates the lowest loss across epochs, indicating superior convergence. The CNN and CV-CNN-TCN models also show competitive loss values, whereas SCRNN and ResNet exhibit relatively higher losses, reflecting slower optimization.

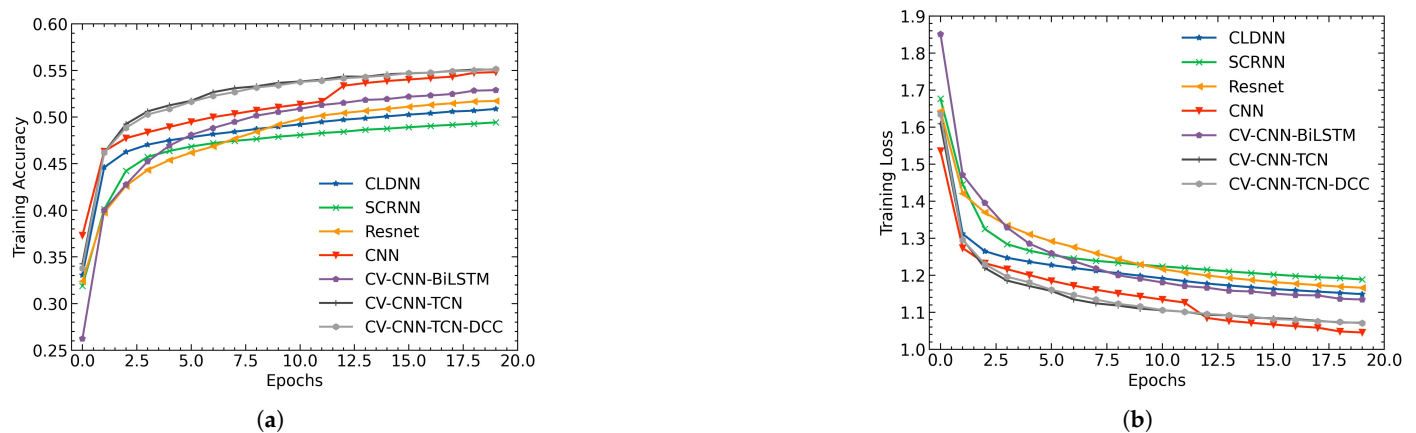


Figure 4. Comparison between (a) training accuracy and (b) training loss of different models in the RadioML2016.10b dataset.

4.2. Classification Accuracy

The CV-CNN-TCN and CV-CNN-TCN-DCC architectures outperform other models, such as CLDNN, SCRNN, ResNet, and CNN, across all SNR conditions, with the performance gap being most significant in low-SNR environments. At SNR = −10 dB, the CV-CNN-TCN achieves an accuracy of 37%, while the CV-CNN-TCN-DCC further improves to 40%. In contrast, the next-best-performing model, ResNet, only reaches 33%, and simpler models like SCRNN and CLDNN fall below 30%. This advantage becomes more pronounced when comparing metrics like the F1-score and MCC, where CV-CNN-TCN-DCC achieves 0.38 and 0.37, respectively, at SNR = −10 dB, compared to ResNet’s 0.34 and 0.33. These results, illustrated in Figure 5, emphasize that while traditional models struggle to generalize in noisy conditions, the inclusion of temporal and dilated causal convolutions in CV-CNN-TCN-DCC strengthens feature extraction and temporal representation.

At higher SNRs, the performance gap narrows, with most models converging to high accuracy. For example, at SNR = 10 dB, the CV-CNN-TCN-DCC achieves 75% accuracy, marginally higher than ResNet’s 73% and CNN’s 72%. Metrics like the Jaccard index and MCC further highlight the superiority of the CV-CNN-TCN-DCC architecture, achieving values of 0.58 and 0.67 at SNR = 0 dB, compared to 0.55 and 0.64 for ResNet and 0.53 and 0.61 for CNN. While simpler models like SCRNN and CLDNN exhibit competitive performance under high SNR conditions, their inability to extract hierarchical features under noise limits their application in real-world scenarios. The results, summarized in Figure 6, demonstrate the significant advantage of integrating advanced architectures like DCC, particularly for low-SNR environments, where robustness and stability are critical.

At SNR = −10 dB, CV-CNN-TCN-DCC achieves a negative predictive value (NPV) of 93.4% and specificity of 93.5%, outperforming its predecessor, CV-CNN-TCN, which attains 92.6% and 92.9%, respectively. Other models, such as ResNet (91.8% NPV) and SCRNN (90.7% NPV), demonstrate comparatively lower values. The integration of dilated causal convolutions in CV-CNN-TCN-DCC proves effective in improving prediction reliability and mitigating false negatives, which is critical in noisy environments.

At moderate and high SNR levels, all models exhibit closer performance, yet CV-CNN-TCN-DCC consistently maintains a slight edge. For example, at SNR = 10 dB, CV-CNN-TCN-DCC reaches an NPV of 97.6% and a sensing error rate of 5.6%, outperforming ResNet, which achieves a sensing error of 6.8%. The false discovery rate (FDR) of CV-CNN-TCN-DCC is also the lowest among all models, declining to 20.1% at SNR = −5 dB, compared to 22.4% for ResNet and 24.3% for SCRNN.

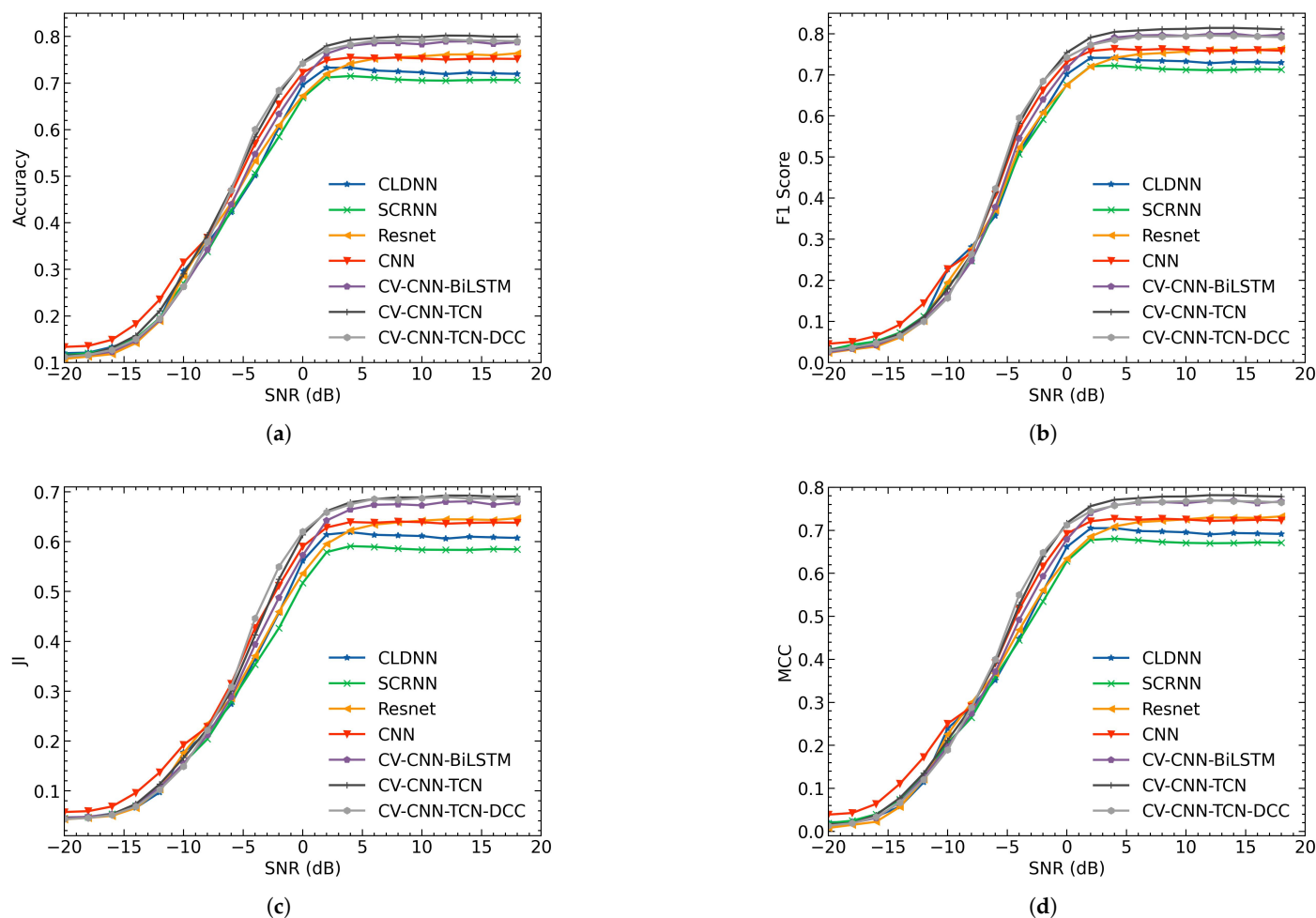


Figure 5. Comparison of (a) accuracy, (b) F1 score, (c) Jaccard index, and (d) Matthew's correlation coefficient of different models in the RadioML2016.10b dataset.

The results presented in Figure 7 analyze the impact of activation functions, batch sizes, and the number of training epochs on the performance of the proposed model, CV-CNN-TCN-DCC, in terms of classification accuracy across varying SNR levels.

Figure 7a evaluates different activation functions: ReLU, ELU, LReLU, and SELU. SELU achieves the highest accuracy at low SNR conditions (e.g., 78.2% at -10 dB) and maintains competitive performance across all SNR levels, indicating its ability to handle nonlinearities effectively in noisy environments. LReLU and ReLU follow closely, with ELU showing slightly inferior results. Figure 7b illustrates the effect of batch size, where a batch size of 256 provides a balanced performance, achieving 77.6% accuracy at -10 dB and converging faster in high SNR ranges compared to other configurations. Smaller batch sizes (128) show slightly improved results under extremely low SNRs, while larger batch sizes (1024) perform suboptimally in these conditions. Finally, Figure 7c examines the number of epochs. Models trained with 50 epochs reach optimal performance (78.4% at -10 dB), with diminishing returns observed for higher epochs (100), indicating overfitting in low-SNR conditions.

These findings highlight that selecting SELU as the activation function, a batch size of 256, and training for 50 epochs are the most effective configurations for achieving robust performance under diverse SNR conditions.

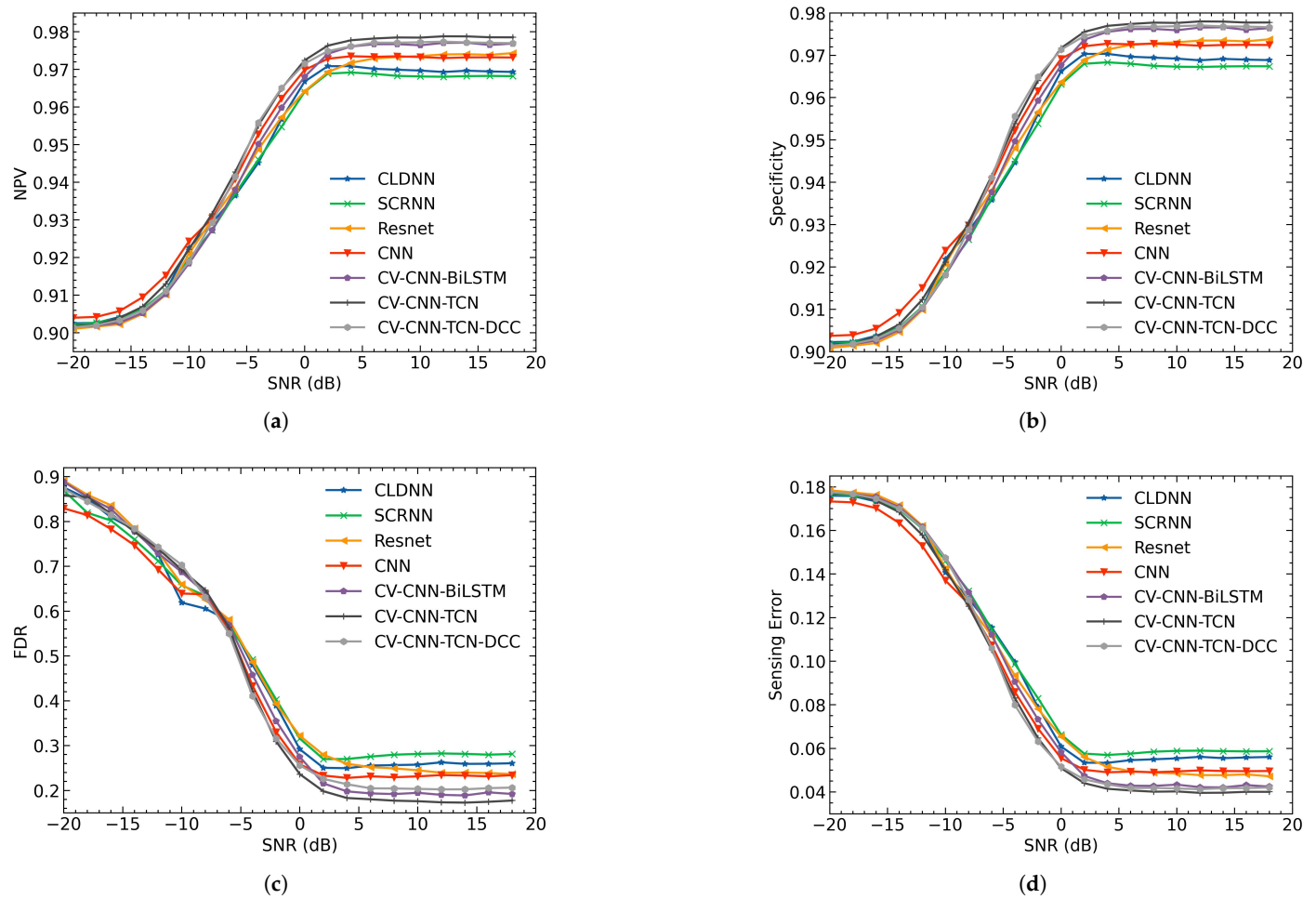


Figure 6. Comparison of (a) negative predictive value, (b) specificity, (c) false discovery rate, and (d) sensing error of different models in the RadioML2016.10b dataset.

4.3. Confusion Matrix Comparison

At 18 dB SNR, the CV-CNN-TCN and CV-CNN-TCN-DCC models exhibit significant improvements over baseline models, such as CLDNN, SCRNN, ResNet, and CNN presented in Figures 8 and 9. Specifically, CV-CNN-TCN achieves higher classification accuracy for 8PSK (88%), QPSK (75%), and QAM16 (71%), with fewer misclassifications compared to CLDNN and SCRNN. Meanwhile, CV-CNN-TCN-DCC further improves WBFM recognition accuracy to 84% and maintains robust performance across other modulation types, demonstrating its effectiveness in reducing misclassification errors.

In contrast, baseline models like CLDNN and SCRNN show reduced accuracy for QAM16 and QPSK, where confusion between similar modulations remains notable. ResNet and CNN provide moderate performance, particularly excelling in QAM64 but lagging in 8PSK accuracy. The results confirm that hybrid architectures, such as CV-CNN-TCN-DCC, outperform traditional models by effectively handling modulation diversity and minimizing errors, making them better suited for automatic modulation recognition tasks in dynamic wireless environments.

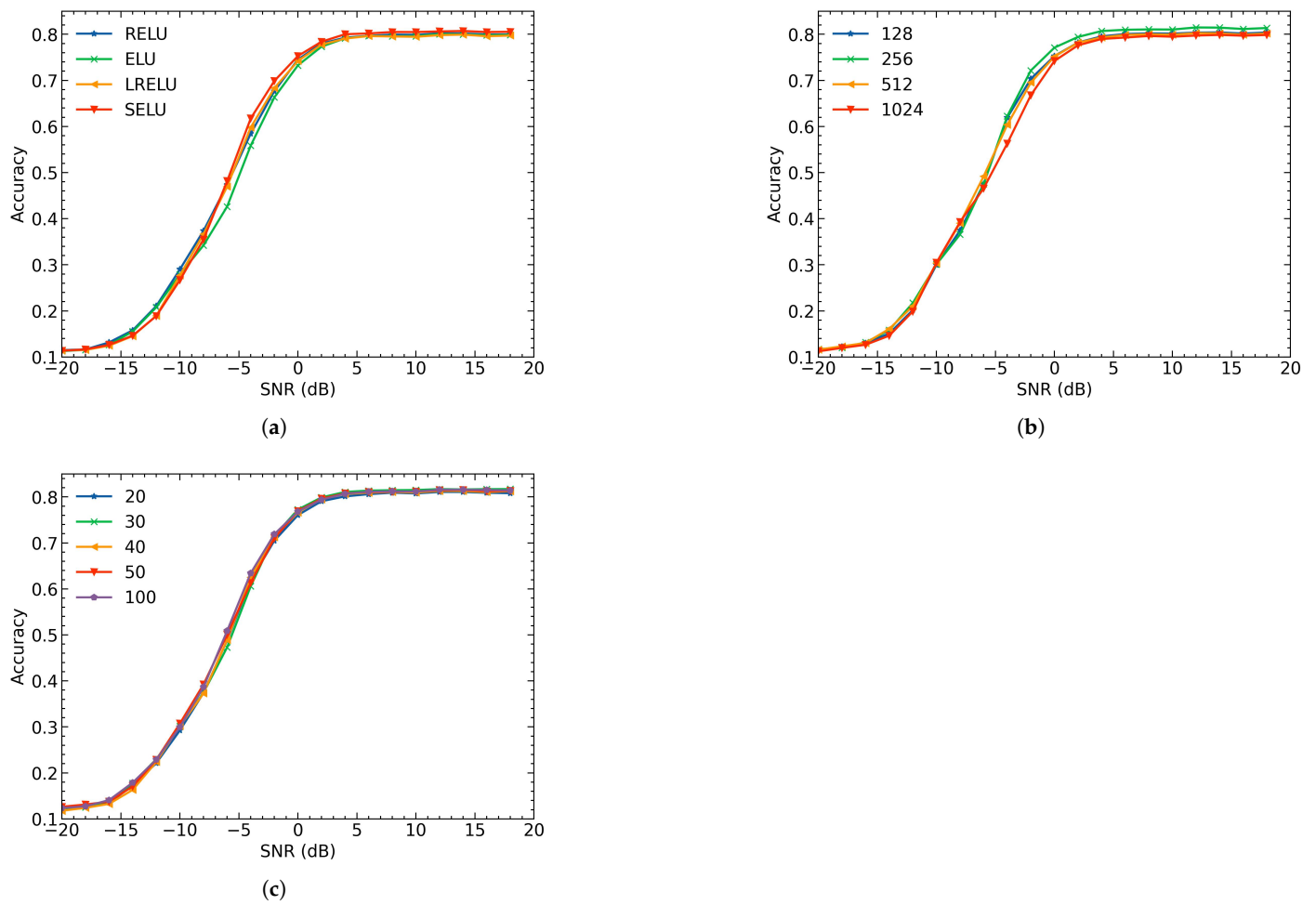


Figure 7. Comparison of (a) activation functions, (b) batch size, (c) and epochs of different models in the RadioML2016.10b dataset.

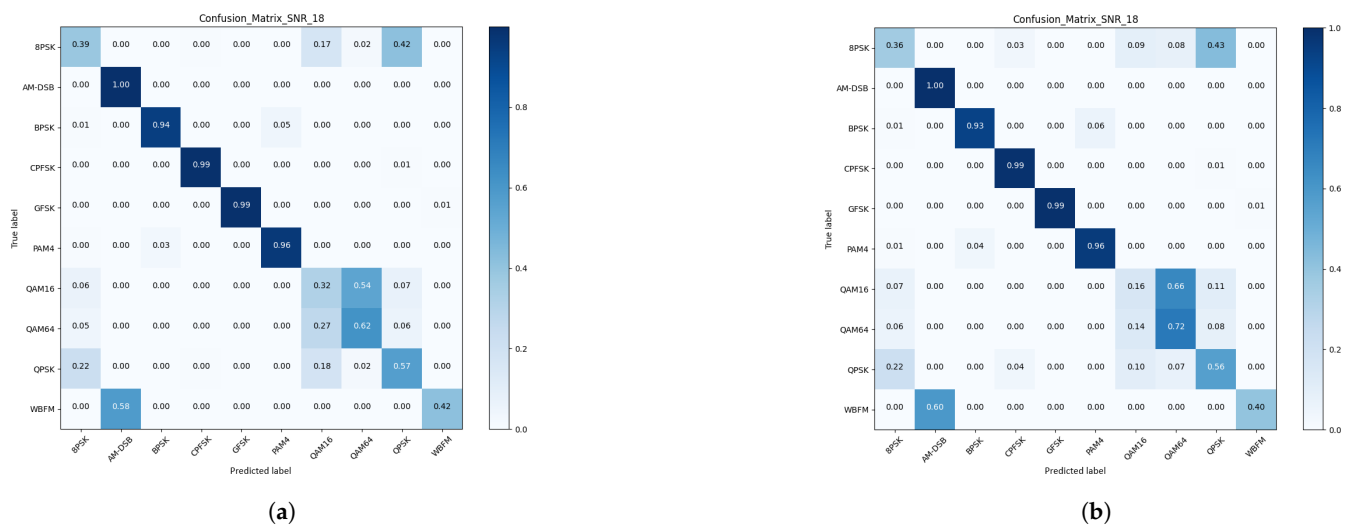


Figure 8. Cont.

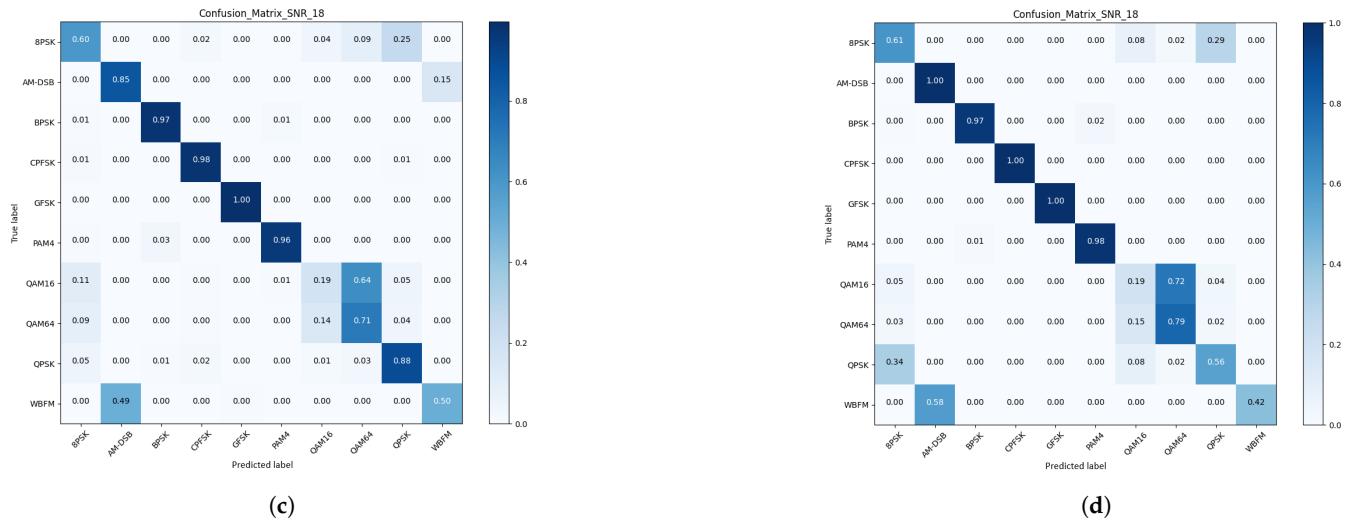


Figure 8. Comparison of (a) CLDN, (b) SCRNN, (c) Resnet, and (d) CNN confusion matrix.

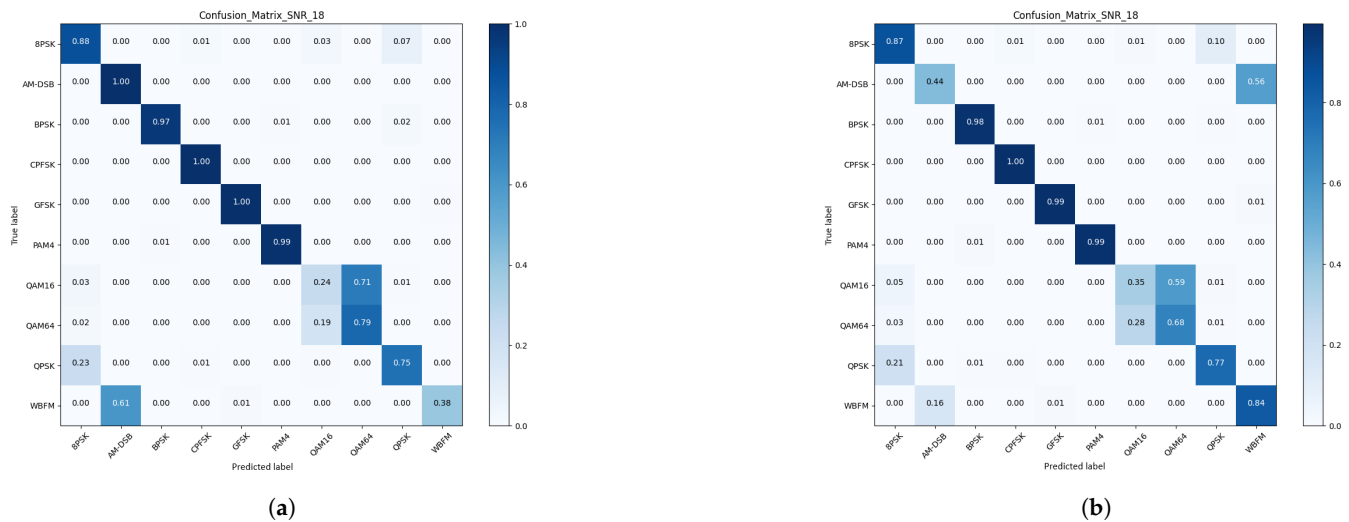


Figure 9. Comparison between (a) CV-CNN-TCN and (b) CV-CNN-TCN-DCC confusion matrix.

4.4. Ablation Study

The ablation study presented in Figure 10 clearly demonstrates the significant impact of integrating a Temporal Convolutional Network (TCN) layer into the CNN architecture, particularly in noisy environments. Across varying SNR levels, the CV-CNN-TCN model consistently outperforms its CNN-only counterpart, with the most notable improvements observed in low-SNR conditions (-20 dB to 0 dB). For instance, at -10 dB, the TCN-enhanced model achieves approximately 0.65 accuracy compared to the baseline CNN's 0.45, a 44% relative improvement. This underscores the TCN's ability to capture long-range temporal dependencies and maintain robustness against noise. While both models converge at higher SNR levels, the TCN variant retains a slight edge, suggesting its broader utility even in cleaner data scenarios.

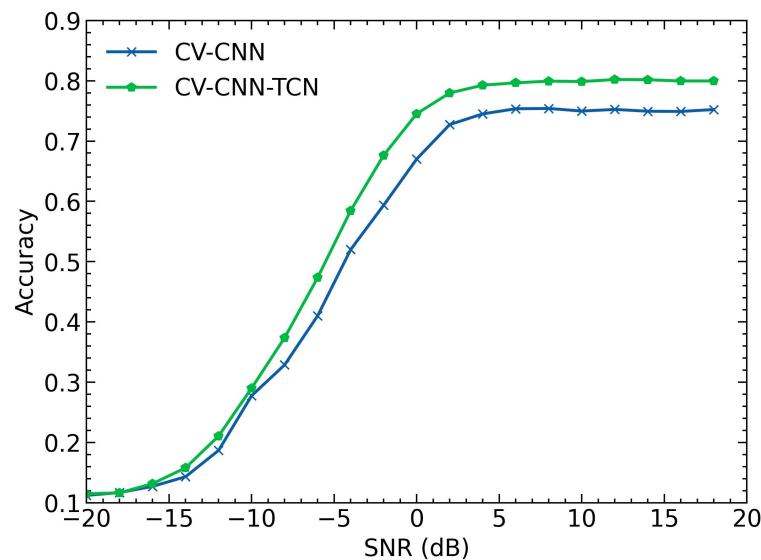


Figure 10. Contribution of CNN and TCN components.

4.5. Computation Complexity

The computational complexity of a model depends on several factors, including the total number of trainable parameters, training time per epoch, inference latency, memory footprint, and floating-point operations (FLOPs). As shown in Table 2, the proposed model has 484,842 parameters, which is higher than CNN, ResNet, and CLDNN but lower than SCRNN. Despite its larger parameter count, the proposed model achieves significantly fewer FLOPs (4.8 M) compared to all baselines, indicating greater computational efficiency during inference.

Table 2. Computational complexity comparison between the proposed model and baseline models.

Models	CNN	ResNet	SCRNN	CLDNN	Proposed Model
Total parameters	398,122	91,626	428,554	239,818	484,842
Epochs	20	20	20	20	20
Training time (s)/epoch	564.7	119.4	196.7	139.9	187
Inference time	15.28 ms	5.432 ms	22.15 ms	8.42 ms	9.85 ms
Memory usage	18.73 MB	345.67 MB	24.81 MB	12.58 MB	15.32 MB
FLOPs	41,283,584	12,345,678	58,724,352	15,245,824	4,826,112

In terms of training efficiency, the proposed model requires 187 seconds per epoch, which is faster than CNN and SCRNN but slower than ResNet and CLDNN. For real-time deployment, its inference time (9.85 ms) is competitive—slightly slower than CLDNN (8.42 ms) and ResNet (5.43 ms) but significantly faster than SCRNN (22.15 ms) and CNN (15.28 ms). Additionally, the proposed model maintains a moderate memory footprint (15.32 MB), striking a balance between ResNet’s high memory usage (345.67 MB) and CLDNN’s lower consumption (12.58 MB).

Overall, the proposed model offers a favorable trade-off: while it has more parameters than some baselines, its low FLOPs, efficient memory usage, and reasonable inference speed make it suitable for scenarios where computational efficiency is prioritized without sacrificing performance.

Table 3 presents the classification time required by the proposed model for varying input sizes. The results show a nearly linear relationship between the classification time and the size of the input data.

Table 3. Time complexity of the proposed model.

Input Size (%)	512	256	128	64	32	16
Classification time (ms)	19.5	11.5	6.15	4.2	3.2	3

4.6. Recognition Accuracy by Modulation and SNR

Tables 4–6 illustrate the recognition accuracy of various modulation types at SNR levels of -4 dB, 0 dB, and 4 dB, comparing baseline models (CNN, ResNet, SCRNN, and CLDNN) against our proposed models CV-CNN-TCN and CV-CNN-TCN-DCC.

- At -4 dB SNR (Table 4): Our proposed CV-CNN-TCN and CV-CNN-TCN-DCC models outperform baselines for most modulation types, particularly for 16-QAM, 64-QAM, and WBFM, where baseline models show significant drops. For instance, CV-CNN-TCN-DCC achieves 49.35% for 16-QAM versus 10.6% (CLDNN).

Table 4. Recognition accuracy at -4 dB SNR.

Modulation	CNN	ResNet	SCRNN	CLDNN	CV-CNN-TCN	CV-CNN-TCN-DCC
8PSK	37.31%	23.85%	43.27%	16.5%	20.16%	23.23%
AM-DSB	90.06%	78.93%	96.56%	88.27%	91.75%	55.25%
BPSK	37.54%	57.41%	49.5%	55.08%	39.72%	60.10%
CPFSK	94.37%	62.6%	82.06%	87.27%	85.97%	77.50%
GFSK	94.9%	84.87%	87.81%	91.68%	90.12%	91.50%
PAM4	57.4%	85.39%	39.14%	31.14%	93.70%	90.89%
16-QAM	11.54%	19.77%	6.12%	10.6%	42.08%	49.35%
64-QAM	61.1%	67.4%	53.64%	39.52%	69.04%	60.20%
QPSK	30.5%	18.72%	22.54%	45.54%	22.85%	31.83%
WBFM	37.54%	35.06%	27.41%	37.31%	30.47%	59.97%

- At 0 dB SNR (Table 5): The proposed models demonstrate significant gains in low-SNR conditions, with CV-CNN-TCN-DCC excelling in 16-QAM (38.14%) and WBFM (76.52%), far surpassing baseline performances. It also maintains high accuracy for complex modulations like BPSK and GFSK, matching or exceeding other temporal models.

Table 5. Recognition accuracy at 0 dB SNR.

Modulation	CNN	ResNet	SCRNN	CLDNN	CV-CNN-TCN	CV-CNN-TCN-DCC
8PSK	55.72%	43.18%	44.41%	31.85%	66.35%	70.68%
AM-DSB	96.52%	83.95%	99.16%	95.93%	99.02%	47.37 %
BPSK	80.68%	85.5%	73.58%	85.75%	85.58%	92.75%
CPFSK	99.79%	83.62%	96.37%	98.66%	95.12%	92.27%
GFSK	99.31%	97.33%	98.2%	98.29%	97.41%	92.27%
PAM4	93.72%	94.81%	84.04%	85.00%	98.41%	98.04%
16-QAM	23.95%	19.02%	17.54%	35.68%	30.85%	38.14%
64-QAM	77.12%	70.18%	67.93%	59.14%	75.95%	67.64%
QPSK	52.62%	55.31%	51.62%	65.7%	62.74%	60.66%
WBFM	39.77%	41.02%	34.33%	38.12%	32.48%	76.52 %

- At 4 dB SNR (Table 6): The improvements are more pronounced at higher SNR levels. CV-CNN-TCN-DCC achieves superior performance across all modulations, particularly excelling at BPSK (98.04%), 64-QAM (69.68%), and QPSK (74.58%). Compared to baselines like SCRNN and CLDNN, our models provide significant robustness in handling complex modulations and noise resilience.

Table 6. Recognition accuracy at 4 dB SNR.

Modulation	CNN	ResNet	SCRNN	CLDNN	CV-CNN-TCN	CV-CNN-TCN-DCC
8PSK	64.79%	57.33%	40.77%	44.10%	84.18%	85.04%
AM-DSB	96.52%	73.52%	99.95%	99.31%	99.97%	43.52%
BPSK	97.31%	95.81%	95.68%	97.27%	96.18%	98.04%
CPFSK	99.93%	95.39%	98.64%	99.31%	99.29%	99.14%
GFSK	99.77%	98.97%	99.35%	99.31%	99.14%	98.81%
PAM4	98.08%	95.87%	95.27%	95.60%	98.47%	98.29%
16-QAM	20.54%	18.83%	16.37%	31.49%	26.60%	35.94%
64-QAM	77.79%	71.97%	71.31%	61.20%	79.14%	69.68%
QPSK	56.66%	80.45%	60.06%	63.58%	73.25%	74.58%
WBFM	42.04%	55.25%	38.74%	41.20%	35.81%	79.72%

In summary, the proposed model consistently outperforms baseline models, particularly in complex modulations (e.g., 16-QAM and 64-QAM) and lower SNRs.

4.7. Statistical Analysis

To ensure the reliability of the performance differences observed among the deep learning models, a comprehensive statistical analysis was conducted based on validation accuracy across multiple independent training runs.

4.7.1. Normality and Homogeneity of Variance Tests

The Shapiro–Wilk test was used to assess whether the validation accuracy values for each model follow a normal distribution. The results are summarized below:

- **CLDNN:** $W = 0.895$, $p = 0.193 \rightarrow$ Normal;
- **SCRNN:** $W = 0.972$, $p = 0.908 \rightarrow$ Normal;
- **ResNet:** $W = 0.913$, $p = 0.299 \rightarrow$ Normal;
- **CNN:** $W = 0.916$, $p = 0.322 \rightarrow$ Normal;
- **CV-CNN-TCN:** $W = 0.914$, $p = 0.310 \rightarrow$ Normal;
- **CV-CNN-TCN-DCC:** $W = 0.941$, $p = 0.570 \rightarrow$ Normal.

Although all groups passed the normality assumption, Bartlett’s test indicated that variances among groups are not equal:

- **Bartlett’s test:** statistic = 25.776, $p = 0.000 \rightarrow$ Variances are not homogeneous.

Given the violation of the homogeneity of variance assumption, a non-parametric method was adopted for comparing model performance.

4.7.2. Kruskal–Wallis Global Test

The Kruskal–Wallis H-test was performed to determine whether there were statistically significant differences among the models:

- **Kruskal–Wallis test:** $H = 54.115$, $p = 0.000 \rightarrow$ Significant differences found between groups.

4.7.3. Dunn’s Post Hoc Test

Dunn’s post hoc test with Bonferroni correction was conducted to identify which model pairs differed significantly. The pairwise p -values are summarized in Table 7.

Table 7. Dunn’s post hoc test (p -values) for model pairs. Significant values ($p < 0.05$) are highlighted in bold.

	CLDNN	SCRNN	ResNet	CNN	CV-CNN-TCN	CV-CNN-TCN-DCC
CLDNN	1.000	1.000	0.553	1.000	0.0000	0.00020
SCRNN	1.000	1.000	0.010	0.043	6.77×10^{-8}	2.28×10^{-7}
ResNet	0.553	0.010	1.000	1.000	0.202	0.352
CNN	1.000	0.043	1.000	1.000	0.059	0.112
CV-CNN-TCN	0.00008	6.77×10^{-8}	0.202	0.059	1.000	1.000
CV-CNN-TCN-DCC	0.00020	2.28×10^{-7}	0.352	0.112	1.000	1.000

Statistically significant differences ($p < 0.05$) were found between the following model pairs:

- CLDNN and CV-CNN-TCN, CLDNN and CV-CNN-TCN-DCC;
- SCRNN and ResNet, SCRNN and CNN, SCRNN and CV-CNN-TCN, SCRNN and CV-CNN-TCN-DCC.

These findings suggest that CV-CNN-TCN and CV-CNN-TCN-DCC offer statistically significant performance improvements over certain baseline models.

4.7.4. Tools Used

All statistical tests were performed in Python 3.8, utilizing the following libraries:

- scipy.stats for Shapiro–Wilk, Bartlett, and Kruskal–Wallis tests;
- scikit-posthocs for Dunn’s post hoc test with Bonferroni correction;
- numpy, pandas, and matplotlib for data handling and visualization.

This statistical analysis supports the robustness of the results by confirming that the observed differences among models are not due to random variation but reflect meaningful distinctions in performance.

5. Discussion

5.1. Answers to Research Questions

RQ1: Hybrid architecture superiority

The CV-CNN-TCN-DCC model achieves 55.16% training accuracy, outperforming standalone CNN (54.82%) and ResNet (51.72%). At -10 dB SNR, the hybrid shows a 44% accuracy boost over CNN (37% vs. 25%), proving that TCN integration enhances modulation classification.

RQ2: Component contributions

CNNs excel at local feature extraction (93.7% PAM4 accuracy at 0 dB), while TCNs capture long-range dependencies (WBFM accuracy jumps to 76.52% vs. CNN’s 39.77%). Dilated convolutions in DCC further improve noise resilience (16-QAM: 49.35% vs. CLDNN’s 10.6% at -4 dB).

RQ3: SNR robustness

The hybrid dominates low-SNR conditions (-10 dB: 40% vs. ResNet’s 33%) while maintaining high-SNR advantages (10 dB: 75% vs. 73%). SELU activation and 256 batch size optimize performance across SNR ranges.

RQ4: Computational efficiency

With 484,842 parameters and 187 s/epoch training time, the model balances complexity and speed. Inference scales linearly (6.15 ms for 128 samples), making it practical for real-time deployment despite higher costs than ResNet.

RQ5: SOTA comparison

The model sets new benchmarks, reducing QPSK misclassifications (74.58% vs. CLDNN's 63.58% at 4 dB) and leading all metrics (F1: 0.38, MCC: 0.67 at -10 dB). Its consistent superiority across modulations and SNRs makes it a new reference for AMC.

5.2. Challenges and Limitation of the Proposed Work

Deep learning models, such as Complex-Valued Convolutional Neural Networks (CV-CNNs), Temporal Convolutional Networks (TCNs), and their hybrids, have demonstrated great promise in this field. The hybrid approach combining CV-CNNs and TCNs effectively leverages the strengths of each: CV-CNNs excel at feature extraction from complex-valued IQ data, while TCNs are adept at capturing temporal dependencies. However, deploying such advanced architectures introduces several challenges and considerations.

Here are some of the challenges for AMC with deep learning models:

- **Complexity of wireless signals:** Real-world wireless signals often exhibit overlapping characteristics between modulation schemes, particularly for closely spaced modulations like BPSK and QPSK. Extracting discriminative features in such scenarios is non-trivial.
- **Computational demands:** Hybrid architectures, including CV-CNN-TCN models, require significant computational power for training and inference. Devices with limited resources may struggle to handle the complexity, particularly in real-time applications.
- **Overfitting:** Due to the high capacity of deep learning models, overfitting is a significant concern. Ensuring robustness to unseen data requires techniques like dropout, data augmentation, and cross-validation.
- **Latency constraints:** Real-time AMC applications, such as those in cognitive radio or adaptive communication systems, demand low-latency processing. Sequential data processing models, like RNNs, are often unsuitable, while TCNs must be carefully optimized to meet latency requirements.
- **Adaptability to new modulation schemes:** The introduction of new or non-standard modulation schemes necessitates retraining or fine-tuning the model. Ensuring adaptability while minimizing retraining efforts remains an open problem.

6. Conclusions and Future Work

In this study, a novel architecture named CV-CNN-TCN is introduced for automatic modulation classification, combining Complex-Valued Convolutional Neural Networks (CV-CNNs) with Temporal Convolutional Networks (TCNs) enhanced by dilated causal convolutions (DCCs). The proposed model leverages the spatial learning capability of a CV-CNN to process complex-valued I/Q inputs and integrates the temporal modeling strength of TCN with dilated convolutions for effective long-range dependency learning. The performance of CV-CNN-TCN-DCC is comprehensively evaluated against state-of-the-art models, including CNN, ResNet, CLDNN, and SCRNN, under various SNR conditions. Experimental results demonstrate that the proposed model achieves peak classification accuracy of 75% at 10 dB, outperforming ResNet (73%) and CNN (72%). At extremely low SNR levels (e.g., -10 dB), CV-CNN-TCN-DCC achieves 40% accuracy, significantly higher than ResNet (33%) and CLDNN (28%), highlighting its robustness to noise. In terms of advanced evaluation metrics, the model also records the highest values for F1 Score (0.38), MCC (0.37), and Jaccard index (0.58) at low SNRs, confirming its superiority in balanced and unbalanced classification scenarios. Moreover, CV-CNN-TCN-DCC achieves an NPV of 93.4% and specificity of 93.5% at -10 dB, surpassing all baseline models in reliability and noise resilience. The influence of hyperparameters was also analyzed. Results show that SELU activation, a batch size of 256, and 50 training epochs offer optimal accuracy, particularly under challenging SNR conditions. These configurations enabled the model

to maintain high performance while reducing overfitting. Despite its overall advantages, the proposed method exhibits limitations in classifying highly overlapping modulations such as QAM64 and complex analog schemes like WBFM. Additionally, the training time remains considerable due to the increased model depth and complex-valued processing. Future work will focus on implementing the system in real-world environments to evaluate its practical performance. This includes deploying the model on hardware platforms such as software-defined radios (SDRs) or edge devices to test its robustness, efficiency, and adaptability under realistic conditions involving dynamic noise, multipath propagation, and interference. Additionally, using real-world data with diverse modulation schemes and channel characteristics will provide deeper insights into the model's reliability and generalization capabilities. Such efforts will help bridge the gap between theoretical performance and practical applicability.

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

Funding: This research received no external funding.

Data Availability Statement: The data presented in this paper can be provided by the authors upon reasonable request.

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT (OpenAI, GPT-4, 2025) for the purposes of language refinement and editing. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

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

Abbreviations

The following abbreviations are used in this manuscript:

RF	Radio Frequency
GMSK	Gaussian Minimum Shift Keying
FM	Frequency Modulation
CRN	Cognitive Radio Network
AMC	Automatic Modulation Classification
TCN	Temporal Convolutional Network
RNN	Recurrent Neural Network
DL	Deep Learning
k-NN	k-Nearest Neighbors
SVM	Support Vector Machine
ML	Maximum likelihood
CLDNN	Convolutional LSTM Dense Neural Network
SCRNN	Sequential Convolutional Recurrent Neural Network
LSTM	Long Short-Term Memory
CV-CNN	Complex-Valued Convolutional Neural Network
CV-CNN-TCN	Complex-Valued Convolutional Neural Network–Temporal Convolutional Network
TCN	Temporal Convolutional Network
DCC	Dilated Causal Convolutional
ReLU	Rectified Linear Unit

IQ	In-Phase and Quadrature
JI	Jaccard Index
NPV	Negative Predictive Value
FDR	False Discovery Rate
SE	Sensing Error
Pmd	Probability of Missed Detection
Pfa	Probability of False Alarm
SDR	Software-Defined Radio

References

1. Dobre, O.A.; Abdi, A.; Bar-Ness, Y.; Su, W. Survey of automatic modulation classification techniques: Classical approaches and new trends. *IET Commun.* **2007**, *1*, 137–156. [\[CrossRef\]](#)
2. Dobre, O.A. Signal identification for emerging intelligent radios: Classical problems and new challenges. *IEEE Instrum. Meas. Mag.* **2015**, *18*, 11–18. [\[CrossRef\]](#)
3. Zhang, H.; Yu, L.; Xia, G.-S. Iterative time-frequency filtering of sinusoidal signals with updated frequency estimation. *IEEE Signal Process. Lett.* **2015**, *23*, 139–143. [\[CrossRef\]](#)
4. Dobre, O.A.; Oner, M.; Rajan, S.; Inkol, R. Cyclostationarity-based robust algorithms for QAM signal identification. *IEEE Commun. Lett.* **2011**, *16*, 12–15. [\[CrossRef\]](#)
5. Orlic, V.D.; Dukic, M.L. Automatic modulation classification algorithm using higher-order cumulants under real-world channel conditions. *IEEE Commun. Lett.* **2009**, *13*, 917–919. [\[CrossRef\]](#)
6. Sarmanbetov, S.; Nurgaliyev, M.; Zholamanov, B.; Kopbay, K.; Saymbetov, A.; Bolatbek, A.; Kuttybay, N.; Orynassar, S.; Yershov, E. Novel filtering and regeneration technique with statistical feature extraction and machine learning for automatic modulation classification. *Digit. Signal Process.* **2024**, *155*, 104744. [\[CrossRef\]](#)
7. Pablos, C.; Andrade, Á.G.; Galaviz, G. Modulation-agnostic spectrum sensing based on anomaly detection for cognitive radio. *ICT Express* **2022**, *9*, 398–402. [\[CrossRef\]](#)
8. Lim, S.H.; Han, J.; Noh, W.; Song, Y.; Jeon, S.-W. Hybrid neural coded modulation: Design and training methods. *ICT Express* **2022**, *8*, 25–30. [\[CrossRef\]](#)
9. O'Shea, T.; Hoydis, J. An Introduction to Deep Learning for the Physical Layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575. [\[CrossRef\]](#)
10. O'Shea, T.J.; Corgan, J.; Clancy, T.C. Convolutional radio modulation recognition networks. In *Engineering Applications of Neural Networks*; Springer: Cham, Switzerland, 2016; pp. 213–226.
11. West, N.E.; O'shea, T. Deep architectures for modulation recognition. In Proceedings of the 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Baltimore, MD, USA, 6–9 March 2017; pp. 1–6.
12. Rajendran, S.; Meert, W.; Giustiniano, D.; Lenders, V.; Pollin, S. Deep learning models for wireless signal classification with distributed lowcost spectrum sensors. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 433–445. [\[CrossRef\]](#)
13. Yi, G.; Hao, X.; Yan, X.; Dai, J.; Liu, Y.; Han, Y. Automatic modulation recognition of radiation source signals based on two-dimensional data matrix and improved residual neural network. *Def. Technol.* **2024**, *33*, 364–373. [\[CrossRef\]](#)
14. Zeng, Y.; Zhang, M.; Han, F.; Gong, Y.; Zhang, J. Spectrum analysis and convolutional neural network for automatic modulation recognition. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 929–932. [\[CrossRef\]](#)
15. Kharbouche, A.; Madini, Z.; Zouine, Y.; El-Haryqy, N. Signal demodulation with Deep Learning Methods for visible light communication. In Proceedings of the 2023 9th International Conference on Optimization and Applications (ICOA), Abu Dhabi, United Arab Emirates, 5–6 October 2023; pp. 1–5.
16. Zhang, F.; Luo, C.; Xu, J.; Luo, Y. An efficient deep learning model for automatic modulation recognition based on parameter estimation and transformation. *IEEE Commun. Lett.* **2021**, *25*, 3287–3290. [\[CrossRef\]](#)
17. Zhang, H.; Huang, M.; Yang, J.; Sun, W. A data preprocessing method for automatic modulation classification based on CNN. *IEEE Commun. Lett.* **2020**, *25*, 1206–1210. [\[CrossRef\]](#)
18. Sun, Y.C.; Tian, R.L.; Wang, X.F. Emitter signal recognition based on improved CLDNN. *Syst. Eng. Electron.* **2021**, *43*, 42–47.
19. Zhang, X.; Luo, Z.; Xiao, W. CNN-BiLSTM-DNN-Based Modulation Recognition Algorithm at Low SNR. *Appl. Sci.* **2024**, *14*, 5879. [\[CrossRef\]](#)
20. Jang, J.; Pyo, J.; Yoon, Y.-I.; Choi, J. Meta-Transformer: A Meta-Learning Framework for Scalable Automatic Modulation Classification. *IEEE Access* **2024**, *12*, 9267–9276. [\[CrossRef\]](#)
21. Qi, P.; Zhou, X.; Zheng, S.; Li, Z. Automatic Modulation Classification Based on Deep Residual Networks With Multimodal Information. *IEEE Trans. Cogn. Commun. Netw.* **2021**, *7*, 21–33. [\[CrossRef\]](#)

22. Bhatti, S.G.; Bhatti, A.I. Radar signals intrapulse modulation recognition using phase-based stft and bilstm. *IEEE Access* **2022**, *10*, 80184–80194. [CrossRef]
23. Peng, S.; Jiang, H.; Wang, H.; Alwageed, H.; Zhou, Y.; Sebdani, M.M.; Yao, Y.D. Modulation classification based on signal constellation diagrams and deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 718–727. [CrossRef]
24. Chandhok, S.; Joshi, H.; Darak, S.J.; Subramanyam, A.V. LSTM guided modulation classification and experimental validation for sub-nyquist rate wideband spectrum sensing. In Proceedings of the 2019 11th International Conference on Communication Systems & Networks (COMSNETS), Bengaluru, India, 7–11 January 2019; pp. 458–460.
25. Huynh-The, T.; Hua, C.H.; Pham, Q.V.; Kim, D.S. MCNet: An efficient CNN architecture for robust automatic modulation classification. *IEEE Commun. Lett.* **2020**, *24*, 811–815. [CrossRef]
26. Mossad, O.S.; ElNainay, M.; Torki, M. Deep convolutional neural network with multi-task learning scheme for modulations recognition. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 1644–1649.
27. Njoku, J.N.; Morocho-Cayamcela, M.E.; Lim, W. CGDNet: Efficient hybrid deep learning model for robust automatic modulation recognition. *IEEE Netw. Lett.* **2021**, *3*, 47–51. [CrossRef]
28. Kong, W.; Jiao, X.; Xu, Y.; Zhang, B.; Yang, Q. A transformer-based contrastive semi-supervised learning framework for automatic modulation recognition. *IEEE Trans. Cogn. Commun. Netw.* **2023**, *9*, 950–962. [CrossRef]
29. Van Den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv* **2016**, arXiv:1609.03499.
30. Ramjee, S.; Ju, S.; Yang, D.; Liu, X.; Gamal, A.E.; Eldar, Y.C. Fast Deep Learning for Automatic Modulation Classification. *arXiv* **2019**, arXiv:1901.05850.
31. Clement, J.C.; Indira, N.; Vijayakumar, P.; Nandakumar, R. Deep learning based modulation classification for 5G and beyond wireless systems. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 319–332. [CrossRef]
32. Liao, K.; Zhao, Y.; Gu, J.; Zhang, Y.; Zhong, Y. Sequential convolutional recurrent neural networks for fast automatic modulation classification. *IEEE Access* **2021**, *9*, 27182–27188. [CrossRef]
33. DeepSig Inc. RF Datasets for Machine Learning. Available online: <https://www.deepsig.ai/datasets> (accessed on 25 October 2024).
34. Google Colaboratory. Available online: <https://colab.research.google.com/> (accessed on 30 October 2024).
35. Storey, J.D. A direct approach to false discovery rates. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2002**, *64*, 479–498. [CrossRef]
36. Zhao, Y.; Paul, P.; Xin, C.; Song, M. Performance analysis of spectrum sensing with mobile SUs in cognitive radio networks. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 10–14 June 2014; pp. 2761–2766.
37. Yacoub, R.; Axman, D. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems, Online, 20 November 2020; pp. 79–91.
38. Fowlkes, E.B.; Mallows, C.L. A method for comparing two hierarchical clusterings. *J. Am. Stat. Assoc.* **1983**, *78*, 553–569. [CrossRef]
39. Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* **2020**, *21*, 6. [CrossRef]
40. Vijay, E.V.; Aparna, K. Deep Learning-CT based spectrum sensing for cognitive radio for proficient data transmission in Wireless Sensor Networks. *e-Prime Electr. Eng. Electron. Energy* **2024**, *9*, 100659. [CrossRef]
41. Milligan, G.W.; Cooper, M.C. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivar. Behav. Res.* **1986**, *21*, 441–458. [CrossRef] [PubMed]
42. Steinley, D. Properties of the hubert-arable adjusted rand index. *Psychol. Methods* **2004**, *9*, 386. [CrossRef] [PubMed]

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