

RESEARCH ARTICLE

Robust Automatic Modulation Classification in Low Signal to Noise Ratio

TO TRUONG AN¹ AND BYUNG MOO LEE², (Senior Member, IEEE)

¹Sejong University, Seoul 05006, South Korea

²Department of Intelligent Mechatronics Engineering and Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, South Korea

Corresponding author: Byung Moo Lee (blee@sejong.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korea Government (MSIT) under Grant NRF-2020R1F1A1048470, and in part by the Faculty Research Fund of Sejong University in 2022.

ABSTRACT In a non-cooperative communication environment, automatic modulation classification (AMC) is an essential technology for analyzing signals and classifying different kinds of signal modulation before they are demodulated. Deep learning (DL)-based AMC has been proposed as an efficient method of achieving high classification performance. However, most current DL-AMC methods have limited generalization capabilities under varying noise conditions, especially at low signal-to-noise ratios (SNRs). Therefore, these methods can not be directly applied to practical systems. In this paper, we propose a threshold autoencoder denoiser convolutional neural network (TADCNN), which consists of a threshold autoencoder denoiser (TAD) and a convolutional neural network (CNN). TADs reduce noise power and clean input signals, which are then passed on to CNN for classification. The TAD network generally consists of three components: the batch normalization layer, the autoencoder, and the threshold denoise. The threshold denoise component uses an auto-learning threshold sub-network to compute thresholds automatically. According to experiments, AMC with TAD improved classification accuracy by 70% at low SNR compared with a model without a denoiser. Additionally, our model achieves an average accuracy of 66.64% on the RML2016.10A dataset, which is 6% to 18% higher than the current AMC model.

INDEX TERMS Automatic modulation classification, autoencoder denoiser, denoise signal, convolutional neural network, automatic modulation classification.

I. INTRODUCTION

Automatic modulation classification (AMC) is an essential technology for analyzing signals in a non-cooperative communication environment [1]. According to the observations of the received signal, AMC determines what modulation is being used at the transmitter. Over the past few decades, it has had various applications in both military and civilian scenarios, such as cognitive radio, signals control, spectrum management, surveillance analysis, and intelligent software-defined radios [2], [3], [4]. It is true that significant progress has been achieved using

The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut¹.

AMC in the aforementioned applications. However, the method still faces several challenges, such as AMC at low SNR, dynamic spectrum management, interference from multiple noises, complex channel environments, and testbed [5]. Therefore, some recent works have tackled these issues. For example, the authors in [3], [6] successfully developed a testbed based on USRP-RIO and simulated a similar scenario to real-world MIMO communication. Currently, the number of communication devices has dramatically increased due to breakthrough innovations in technology and services such as 6G and massive Multiple-Input Multiple-Output (MIMO) [7]. Therefore, this development requires spectrum management to manage a significant amount of communication users. In order

to effectively manage spectral challenges, the creation of effective AMC techniques has been considered to be a crucial and promising strategy. Academic and industrial communication researchers have given considerable attention to this issue.

In general, there are two types of AMC algorithms: Likelihood-based (LB) and Feature-based (FB). The former can be formulated as a multi-hypothesis test. There is a comparison between the likelihood function of an unknown signal and a threshold of a known density function [8]. Nevertheless, it is computationally expensive to determine the decision threshold when there are many signals to be considered. Therefore, the LB-AMC is unfeasibly executed in real-time and inexpensive applications. FB methods are seen to be a good substitute for LB methods because they reduce the complexity of computations. The FB-AMC makes feature extraction and classification. In order to extract features, expert systems carefully extract various manual features, such as wavelet-based features [9], instantaneous features [10], and statistical features [3], [11].

Currently, deep learning (DL) has a large number of successes in various application domains, such as natural language processing (NLP) [12], computer vision [13], robotics [14], and signal processing [15]. DL has gained greater popularity due to its ability to manipulate big data effectively, as well as its ability to generalize learner models to a wide range of domains in the information sciences. For this reason, many researchers are giving a great deal of thought to creating a DL-based AMC. T. O'Shea et al. in [16] proposed convolutional neural networks to classify modulation samples using raw IQ samples. The time-frequency analysis was used to classify modulation in CNN-AMC in [17], [18], [19]. In [5], [20], [21], the authors demonstrated that a state-of-the-art long short-term memory AMC could be used to evaluate both the augmentation method of radio signals in the training phase as well as the inference phase using Gaussian noise, rotation, or flipping. In addition, some papers [22], [23], [24] used deep neural networks (DNNs) for AMC.

During the past few years, deep learning has made significant progress in AMC. Although existing AMC models perform well and provide high classification accuracy at high SNR, these models do not perform well when the SNR is low. Multiple noises in the channel environment, as well as the complexity of the channel environment, make AMC more challenging at low SNR. The performance of AMC can be improved by eliminating such noises and increasing the signal-to-noise ratio. Thus, it is possible to implement signal de-noising algorithms prior to applying AMC techniques [25], [26]. LRR algorithm, for example, is used to de-noise images of cyclic spectrums before AMC models are implemented [27]. It is also possible to implement de-noising signals by implementing image de-noising algorithms [28], [29].

Motivated by successful de-noising AMC models, in this study, we present a TADCNN, which combines a TAD

denoiser layer with a CNN layer for modulation recognition. In the proposed model, there is a de-noising layer to de-noise the raw IQ signal, and then the CNN model uses the denoised signal as an input for classification modulation. According to the experiments, the proposed method improves by 70% recognition accuracy under low SNR conditions while maintaining parameter utilization efficiency. The major contributions of this study include the following:

- To improve the accuracy of AMC at low SNR, we propose a DL-based AMC called TADCNN, which consists of a TAD denoiser and CNN.
- We propose a TAD denoiser to denoise the incoming signal automatically. In order to reduce noise power, the TAD can extract and remove the noise and unimportant features from the input signal.
- We also propose a novel threshold algorithm to denoise signals in order to enhance the denoising performance. The threshold can be learned automatically for each incoming signal.
- We examine the effect of the threshold algorithm on the modulation classification accuracy and also analyze the influence of parameters of the threshold algorithm on the overall accuracy.
- We evaluate the model's performance by contrasting it with recent works in terms of classification accuracy, the number of training parameters, and inference time.

The remaining sections of the paper can be found in the following order: Section II reviews some related automatic modulation classification models. There are two parts in Section III: the problem statement and the system model. In Section IV, we present a combined de-noising and CNN models for modulation classification. Results from the experiment and a discussion of the results are provided in Section V, and the final section, Section VI, provides the conclusion.

II. RELATED WORK

There has been extensive research conducted on AMC technology over the past decade in the areas of signal processing and communication. It has provided an intelligent solution for managing and monitoring the radio communications spectrum in both civil and military applications. Research in wireless communications has focused on developing a low-complexity and robust AMC model, especially to improve accuracy under low SNR conditions.

In this section, we review the related works of AMC, which used deep neural networks, convolutional neural networks, and recurrent neural networks (RNNs) to classify modulation.

A. CNN-BASED MODELS

A number of researchers used existing IQ signal datasets, such as RadioML 2016.10A, and RadioML 2016.10B, as the input to CNN. For example, K. Yashashwi, et al. [30] used RadioML 2016.10A for AMC models, and they proposed

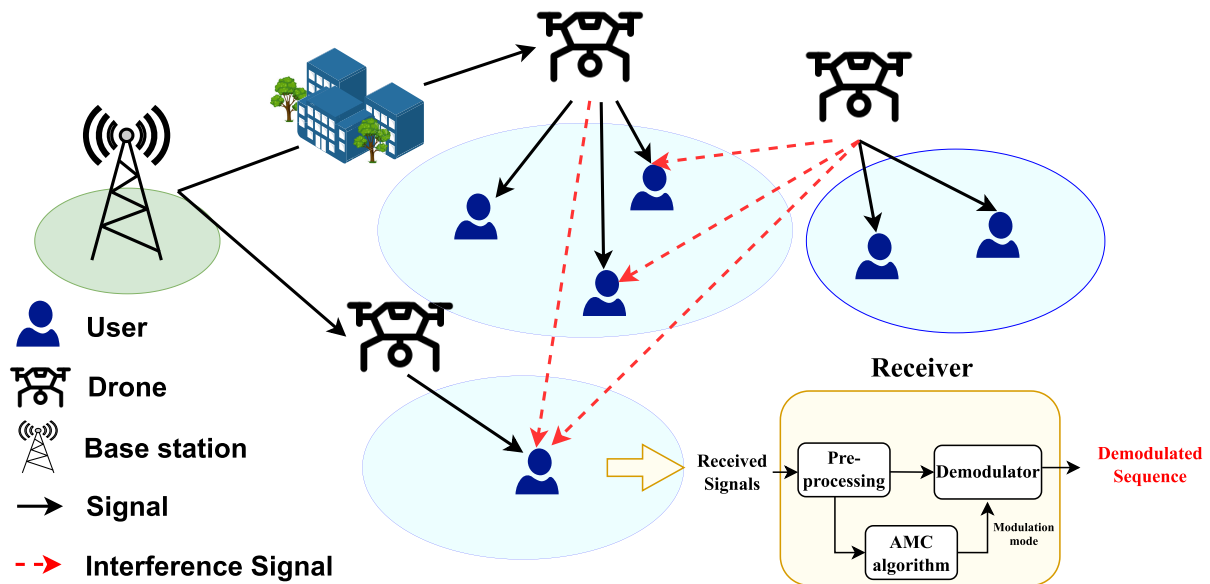


FIGURE 1. AMC-based receiver in non-cooperative communication systems.

a distorted signal correction model in order to enhance the classification accuracy of AMC. The model uses an artificial neural network to evaluate both carrier frequency offsets as well as phase offsets. On the other hand, in [31], this module eliminates the effects of both carrier frequency offsets and phase offsets. This is done by adjusting the frequency and phase of the signal before classification. In [32], attention mechanisms were exploited for the fusion of multiscale features extracted from the data. The categorical cross-entropy loss was optimized by using the SNR, and network performance was improved. In a study by Zeng et al. in [33], it was shown that the short-time Fourier transform could be utilized to convert radio signals into images of spectrograms for achieving high classification accuracy in noisy environments. The authors use the spectrogram images to input CNN to identify modulation. Using the open-source dataset RadioML2016.10A, which covers up to 11 types of modulation, experiments were conducted in order to determine the robustness of the method. In a paper [34] Fu, et al. proposed S-CNN, which decreased the space and time complexity with slightly lower classification accuracy.

In a study by B. Jdid et al. in [35] a novel and robust DL-AMR algorithm is presented by taking advantage of both contextual features and hand-crafted features for a particular signal-to-noise ratio range. The proposed method that is capable of performing robustly under varying noise regimes is proposed by addressing the core issues of feature extraction and selection criteria for features. The proposed DL-based AMR technique contributes significantly to wireless communication because it is able to solve AMC tasks with better classification accuracy while incurring lower computation

complexity due to the adoption of a simpler CNN model.

B. RNN-BASED MODELS

The methodology presented in [36] developed an approach with two-dimensional convolution, one-dimensional convolution, and long short-term memory (LSTM) called MCLDNN. This approach can be used to combine spatial features from both in-phase and quadrature components. In addition, based on MCLDNN, a light and simple structure known as PET-CGDNN was introduced [37]. This model incorporates parameter estimation and transformation (PET), two-dimensional convolution layers, and gated recurrent units (GRUs). CNN and LSTM radio modulation classifiers were used in [38] for the purpose of visualizing the features extracted by using class activation vectors. On the one hand, CNN captures similar radio features regardless of input format. On the other hand, the LSTM classifiers discriminate between modulation types similarly to expert knowledge. The Dual Path Networks (DPNs) were proposed in [39] as a method of combining AMC and recovery symbols. An architecture based on deep learning and linear signal processing is used to estimate the parameters of signals. In addition, it is used to correct distortions caused by multi-path fading and carrier frequency offset. The model contains residual blocks, LSTMs, and RNNs. In brief, five separate outputs were generated from received samples. It is significant to note that every output has a different loss function. Consequently, the training loss consists of a combination of various loss functions. According to the results, the model is capable of estimating signal distortions with high accuracy and also capable of outperforming

TABLE 1. Comparison between our work and related works.

Publication	Denoise	Novel threshold algorithm	CNN	ReNet	LSTM
X. Fu <i>et al.</i> in [36]	x	x	✓	x	x
J. Xu <i>et al.</i> in [38]	x	x	x	x	✓
F. Zhang <i>et al.</i> in [39]	x	x	x	x	✓
T. Huynh-The <i>et al.</i> in [46]	x	x	x	✓	x
J. Krzyzostet <i>et al.</i> in [47]	x	x	✓	x	x
Our work	✓	✓	✓	✓	x

many deep Learning methods regarding the accuracy of classification.

C. DNN-BASED MODELS

First and foremost, Jingreng Lei *et al.* in [40] combined DNN and a Wiener filter to reduce the noise of the input signal. As the data input to the subsequent DNN classifier, the signal cycle spectrum is extracted and intercepted as two-dimensional profiles of the cycle frequency axis. Next, it is proposed in [41] that the method utilizes a large hybrid deep neural network (HDNN) and a layered resnet network (LRN). The former consists of layers of Resnet and LSTM-RNN. In order to reduce the amount of time and storage consumed in DL, a cross-model approach was proposed. This means that real-time applications can meet the requirement of low complexity. There is a combination of deep belief network (DBN)-support vector machine (SVM) classifiers proposed in [42]. In order to extract relevant features from the received signal, the method utilized stacked RBM networks to construct a DBN. A robust automatic modulation classification was proposed by Huynh-The *et al.* in [43] that multiple convolutional blocks with residual connections and asymmetric convolution kernels might be used to obtain a robust structure, and this structure, known as MCNet, also had outstanding results.

For clarification, Table 1 summarizes the comparison between our work and related works. According to the table, we can make the following observations:

- The main difference between our work and existing works is that our paper has a denoise function in order to denoise the signal.
- Our work presents a novel threshold algorithm for automatic threshold learning.
- We present a novel DL algorithm with denoise, CNN, and ResNet. This model significantly improves overall accuracy, especially at low SNR.

III. PROBLEM STATEMENT AND SYSTEM MODEL

A. PROBLEM STATEMENT

Modulation plays an important role in the field of wireless communication. In this process, the properties of signals with a high frequency (known as carrier signals) are changed depending on the characteristics of a baseband signal. It is typical for the process of modulation to match the characteristics of the signal with that of the channel in order to provide accurate transmission of information between transmitters and receivers located at a distance from one another. Signals

received by receivers in a wireless communication system are given as follows:

$$y(t) = h(t) * X(t) + n \quad (1)$$

where $h(t)$ represents the communication channel impulse response, $X(t)$ denotes a noise-free transmission signal, and n represents the additive white Gaussian noise (AWGN) with zero mean and variance σ^2 .

The communication channel impulse response is described as follows:

$$h(t) = a * \delta(t - \tau) \quad (2)$$

where a is the amplitude of the signal, δ denotes an impulse function, t is the time interval, and τ represents the channel multi path delay.

Overall, the classification of modulations is an M-class recognition task, where M denotes the types of modulation of a transmitted signal $X(t)$ based on information provided by the received signal $y(t)$. Therefore, the primary purpose of modulation classification is to identify the characteristics of the received signal and to determine the exact type of modulation. In general, modulation classification algorithms are evaluated based on the speed and accuracy of their classification model.

The application of machine learning and deep learning to the classification of modulation is possible. Previous works have shown significant improvements and blossoming results. For example, there are algorithms that require prior signal information, such as carrier frequency, baud rate, and offset timing [30]. Aside from that, there are some ML and DL-based AMCs that are difficult to implement in real-time due to their computational complexity [46].

In recent years, many AMC methods have been proposed to enhance performance. An existing AMC approach achieves high accuracy by requiring high SNR values, which is not consistent with realistic scenarios. Classifying modulations at low SNR is a challenging task. At low signal levels, the DL-based AMC struggles to classify modulations because of various background noises, including frequency selective fading, local oscillator offsets, doppler offsets, additive white Gaussian noise, impulsive noise, co-channel interference, and adjacent channel interference.

B. SYSTEM MODEL

It is considered that a standard non-cooperative communication system is in use in which digital modulation signals are transmitted over a wireless channel by transmitters with interference and AWGN. There is no prior information provided to the receivers regarding the types of modulation, symbol rates, and so on. Once these signals are detected by the receivers, the system proceeds to reprocess the signals, which includes down conversion, band pass filtering, and digitization.

As a result of preprocessing, we can obtain baseband signals, after which the signals are passed through an AMC module to identify the type of modulation. Figure 1 illustrates



FIGURE 2. General process of DL-AMC.

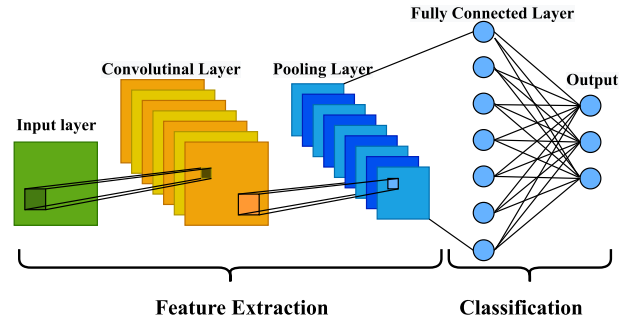


FIGURE 3. CNN architecture.

an AMC-based receiver in non-cooperative communication systems.

This paper focuses on building a deep-learning AMC model for determining the exact modulation type. Hence, the problem of deep learning-based AMC can be expressed as follows:

$$\hat{y} = \operatorname{argmax}_f(y|X; W) \tag{3}$$

where \hat{y} represents the predicted modulation type of DL-AMC, y denotes the truth modulation type, and W is the weight of AMC model. The goal of the DL-AMC technique is to design a low-complexity DL model with high accuracy. Figure 2 shows the general process of DL-AMC. Therefore, in the next section, we propose a DN-CNN model for AMC with the aim of achieving a light structure with high accuracy in both high and low SNR.

IV. THE PROPOSED AMC METHOD

In this section, we present a denoise layer, which uses Auto-Encoder and threshold to denoise input signals, and a CNN classifier architecture.

A. BASIC COMPONENTS

The CNN algorithm is a deep learning algorithm that consists of multiple layers. In terms of multi-dimensional inputs (for example, 2D images), CNN is comparable to the brain of a human being. Furthermore, CNN has a significantly smaller number of parameters than fully connected networks of the same size, and they are less susceptible to diminishing gradients. Figure 3 shows an example of CNN architecture. Nevertheless, the number of trainable parameters is a crucial factor in determining the level of complexity and memory requirements of the CNN model.

It is important to note that both the denoise layer and the proposed CNN share the same basic component, including the convolutional layer, activation functions, batch normalization functions, pooling functions, dropout, and so

TABLE 2. The common activation in DL.

Name	Equation
ReLU	$ReLU(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x \leq 0. \end{cases} \tag{4}$
SeLu	$SeLu(x) = \lambda \begin{cases} x, & \text{if } x > 0, \\ \alpha e^x - \alpha, & \text{if } x \leq 0. \end{cases} \tag{5}$
Softmax	$Softmax(x) = \frac{e^{x_i}}{\sum_{j=0}^k e^{x_k}} \tag{6}$
Sigmoid	$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{7}$

on. These components of the systems are described as follows [25].

Convolutional layer: The purpose of this layer is to create a map that involves learning features from the prior layers and convoluting them with learnable kernels. In the following step, activation functions are applied to these kernels' outputs in order to create an activation map. Table 2 provides several activation functions such as rectified linear units (ReLus), scaled exponential linear units (SeLus), sigmoid, and softmax. Moreover, convolutional kernels are used in convolutional layers, which require significantly fewer parameters than the transformation matrices in fully connected layers. As a result of a smaller number of trainable parameters, deep learning methods will be less likely to suffer from overfitting.

Consequently, it will be easier to achieve relatively high accuracy when testing datasets. Convolutional layer outputs can be described as follows:

$$y_j = \sum_{i \in S_j} x_i * a_{ij} + b_j \tag{8}$$

where x_i corresponds to the i^{th} feature map channel for the input data, y_j corresponds to the j^{th} output channel for the feature map, a represents the convolutional kernel, there is a bias called b , and S_j is the number of channels used to calculate the j^{th} channel of feature maps.

Batch normalization layer: This is a method for normalizing features in every batch. The purpose of batch normalization (BN) is to reduce covariance shift, where features continue to change distribution during training. The process of BN can be described as follows:

$$\mu = \frac{1}{m} \sum_{n=1}^m x_n \tag{9}$$

$$\sigma^2 = \frac{1}{m} \sum_{n=1}^m (x_n - \mu)^2 \tag{10}$$

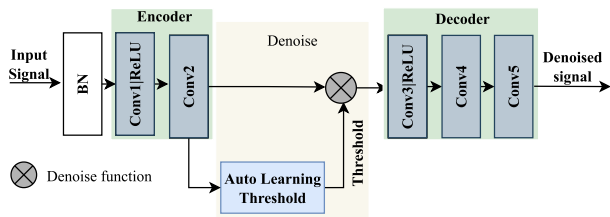


FIGURE 4. Structure of the TAD.

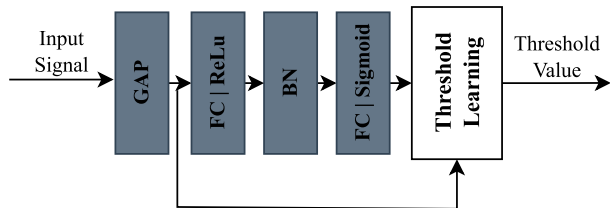


FIGURE 5. Structure of the automatic learning Threshold.

$$\hat{x}_n = \frac{x_n - \mu}{\sqrt{\sigma^2 + \varepsilon}} \tag{11}$$

where x_n and \hat{x}_n represent the input and the normalization input features of each observation in a minibatch, respectively. m is the number of neurons at layer, μ is the mean, and σ^2 is the standard deviation of this hidden activation. There is a constant, ε , which is close to zero.

Pooling layer: A pooling layer reduces the spatial size of representations in order to reduce the number of parameters and computations in the network, as well as allow feature maps to operate independently. A pooling layer can be either maximum pooling or average pooling.

Dropout layer: This layer is commonly used to reduce overfitting in neural networks. Neurons in hidden or visible layers are removed or dropped randomly when using the dropout technique.

B. STRUCTURE OF THRESHOLD AUTOENCODER DENOISER (TAD)

The structure of the TAD network consists of three components, including the BN layer, the autoencoder part, and the threshold denoise part. A threshold denoise sub-network is employed in the threshold denoise component, allowing for automatic threshold computation. The idea of an automatic threshold learning subsystem was introduced in [44] for fault diagnosis. The specific structure of TAD is shown in Figure 4.

Initially, the BN layer normalizes the input data distribution before the encoder layer in order to improve the generalization capabilities of TAD. For batch normalization, the variance magnitude and mean position are altered in order to improve the match between actual and resultant distributions and ensure the nonlinear expressiveness of the model. By backpropagating updates, the BN layer ensures that the input data distribution is relatively stable at each layer. This allows the model to learn more quickly and improves its generalization ability.

TABLE 3. The structure parameters of the autoencoder.

Layer	Output shape
Input	(2 x 128 x 1)
BN	(2 x 128 x 1)
Conv 1(filters 64, size (2,3) + ReLu)	(2 x 128 x 64)
Conv 2(filters 16, size (2,3))	(2 x 128 x 16)
Conv 3(filters 16, size (2,3) + ReLu)	(2 x 128 x 16)
Conv 4(filters 64, size (2,3))	(2 x 128 x 64)
Conv 5(filters 1, size (2,3))	(2 x 128 x 1)

Next, the threshold denoise part includes a global average pooling layer (GAP), two fully connected layers (FC), and two batch normalization layers. The GAP is added to the feature map in order to calculate its average values.

Last but not least, there is the autoencoder part, which can be divided into two groups: encoders and decoders. The former includes two Conv2D layers, while the latter consists of three Conv2D. Table 3 shows the structure parameters of the autoencoder part. As the activation function, we selected ReLu for the Conv1 and Conv3 layers. The filter size of Conv1 and Conv2 are the same. Similarly, Conv2 and Conv3 have the same number of filters. In the last Conv layer, the output is transformed into the input shape which readies for classification.

C. PRINCIPLES OF DENOISE

The noised signal in the encoder layer is transformed into a domain in which useful information can be extracted, while the irrelevant noise has near-zero values.

Furthermore, in the denoise part, the denoise function is used to transform features that are close to zero into zeros, removing any noise-related features. The denoise algorithm is described mathematically as follows:

$$y_i = \begin{cases} 0, & \text{if } |x| \leq \tau \\ Sgn(x)(|x| - \tau), & \text{if } |x| > \tau. \end{cases} \tag{12}$$

where x , y_i correspond to the input and output features, respectively, and τ refers to a threshold value. $Sgn(x)$ is the sign function of the input feature. The sign function can be expressed as follows:

$$Sgn(x) := \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0. \end{cases} \tag{13}$$

It is important to note that in modulation classification problems, we have a variety of modulation types, which means the noise level is different between each modulation. Hence, it is necessary to adjust the threshold value in accordance with the level of noise. In order to alleviate this problem, we propose a denoise function for automatically adjusting thresholds based on the level of noise.

The structure of the automatic learning threshold is also illustrated in Figure 5. Specifically, after the encoder layer,

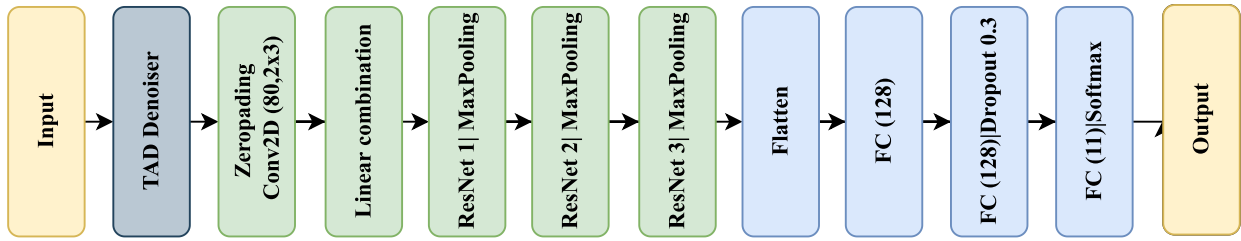


FIGURE 6. The structure of a proposal model.

TABLE 4. TADCNN architecture.

Layer Type	Input Size	Output Size	Details
Input	2 x 128 x 1	-	-
TAD Denoiser	2 x 128 x 1	2 x 128 x 1	$\rho = 1$, Kernel: 2 x 3, Strides = 1
Reshape	2 x 128 x 1	1 x 2 x 128	-
Zero Padding	1 x 2 x 128	1 x 4 x 132	Size: (1, 2)
Convolution	1 x 4 x 132	80 x 3 x 132	Activation: ReLu, Kernel: 2 x 3, Filter: 80
Linear combination	80 x 3 x 132	64 x 2 x 130	-
ResNet Block	32 x 2 x 130	32 x 2 x 130	Filter: 64
MaxPooling	32 x 2 x 130	32 x 1 x 130	Pooling Size: 1 x 1, Strides: (2,1)
ResNet Block	32 x 1 x 130	32 x 1 x 130	Filter: 32
MaxPooling	32 x 1 x 130	32 x 1 x 130	Pooling Size: 1 x 1, Strides: (2,1)
ResNet Block	32 x 1 x 130	32 x 1 x 130	Filter: 16
MaxPooling	32 x 1 x 130	32 x 1 x 130	Pooling Size: 1 x 1, Strides: (2,1)
Flatten	32 x 1 x 130	4160	-
Dense	4160	128	Activation = SeLu
Dense	128	128	Activation = SeLu, Alpha Dropout = 0.3
Dense	128	11	Activation = Softmax

*This Architecture has 671,440 Parameters.

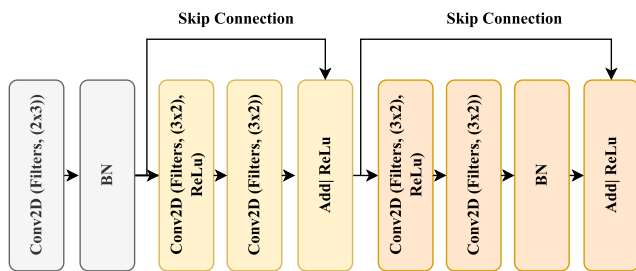


FIGURE 7. The architecture of ResNet Block.

the data is converted to a 1D vector by using the absolute operation and (GAP) in order to generate one dimensions vector. Then, this vector will be used for the generation of a scaling vector α in the range [0, 1]. A threshold τ is calculated by multiplying this scaling value by the mean one-dimensional vectors. Mathematically, the thresholds are described following:

$$\tau = \bar{\beta} * (1 + \rho) \tag{14}$$

$$\bar{\beta} = \alpha * \bar{y} \tag{15}$$

where $\bar{\beta}$ is the scaled average value of the feature map, which is calculated by the scale value α , and the average value \bar{y} of the input. ρ represents an adjustment parameter that is chosen from experiment results.

At the end of the process, the decoder part transforms the denoised feature map into a domain corresponding to the original signal, and then we get a cleaner signal.

D. TADCNN NETWORK STRUCTURE

The purpose of this section is to present a TADCNN model for modulation classification capable of achieving a high level of accuracy both at low and high SNR levels. As illustrated in Figure 6, TADCNN includes three main parts, which are TAD denoiser, CNN, and FC part.

The purpose of the TAD denoiser is to turn the unimportant features into zeros and to eliminate the noise from the signal. The noise reductions improve the classification accuracy in low SNR. In the CNN part, there are one Conv2D and one Linear combination, which was proposed in [45], as well as three ResNet blocks, which enable the neural networks to become deeper and avoid vanishing gradients. The architecture of the ResNet block can be found in Figure 7. Next,

in the Fully connected part, there are three dense layers that have sizes 128, 128, and 11, respectively. This part uses the knowledge obtained from CNN for classification modulation. Table 4 provides a detailed description of the architecture of TADCNN.

In addition, we apply EarlyStopping and Alpha Dropout layers to prevent overfitting. A major advantage of Alpha Dropout is that the mean and variance of the inputs remain unchanged. As part of the optimization process, Adam is employed as an optimizer, with the category cross entropy (CCE) function as a loss function. The loss function is shown as:

$$L_{CCE} = \sum_{i=1}^S Y_i \log(Y_p) \quad (16)$$

$$Y_p = f_{Softmax}(x_i) \quad (17)$$

where Y_i is the ground truth vector, and one-hot encoding can be used to accomplish this. Y_p represents the predicted vector. S represents sample types, and x_i is the i^{th} output of AMC model.

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. DATASET

RadioML 2016.10A dataset is widely accepted in the academic literature as a dataset for modulation classification and has been used in numerous studies. It consists of eight digital modulations and three analog modulations. There were 220k signals produced for a range of 20 signal-to-noise ratios from -20dB to 18dB, and they were divided into 1k signals per modulation per SNR. The 128 samples are passed through the network in a 2×128 vector in which the real and imaginary components of each complex time sample are separated. In addition, the dataset includes various types of background noise, such as frequency selective fading, local oscillator offsets, doppler offsets, additive white Gaussian noise, impulsive noise, co-channel interference, and adjacent channel interference, with further details provided in [47].

The RML2018.10A dataset, on the other hand, has a wide range of SNR values from -20dB to 30dB. Despite having a wide range of SNR values, this dataset has less background noise than that of the RadioML 2016.10A dataset, such as co-channel interference and adjacent channel interference [48]. In our work, we focus on improving accuracy at low SNR with various noises in the signal, so we chose the RadioML 2016.10A dataset for our simulation. Moreover, the RadioML 2016.10A and RML2018.10A have the same values at low SNR, which range from -20dB to 0dB.

B. IMPLEMENTATION DETAILS

In this section, we present an experimental setup for analyzing the performances of TADCNN using the RadioML 2016.10A dataset with 11 modulation schemes. We first train TADCNN on the RadioML 2016.10A dataset using a random selection method in which 70 percent of the dataset is used for training and 30 percent for testing the model. the

hyper-parameters for the training process are decided that the maximum number of epochs in the training process would be 100, the learning rate would be 0.001, and the batch size would be 1024. An EarlyStopping is applied in order to avoid overfitting. The adam optimization algorithm is employed in order to improve the learning process. Algorithm 1 shows the details of the training steps.

Algorithm 1 The Proposed TADCNN Method

Data: Training dataset, validation dataset

Result: Generalized model

Step 1: Randomly divide the dataset into training and validation datasets by 7 : 3;

Step 2: Conduct TADCNN according to Figure 6 and set a cross-entropy L_{CCE} as a loss function according to Eq. 16;

Step 3: Initialize the model Weight W ;

Step 4: Set hyper-parameters, as well as the number of maximum training epochs $N = 100$, the batch size $S = 1024$, and learning rate $\eta = 0.001$. ;

Step 5: Training TADCNN on the training dataset and update TADCNN's weight by Adam until the validation loss is not improved;

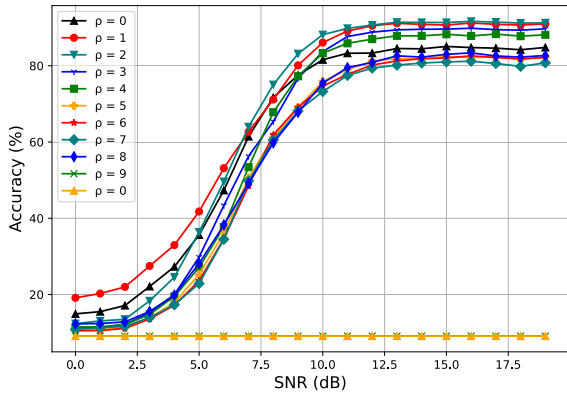
Return: Generalized model

The TADCNN model is developed using TensorFlow 2.9, a machine-learning framework developed by Google. Our models and experiments are conducted using Keras, Tensorflow, and NVIDIA GeForce RTX 2080 GPUs.

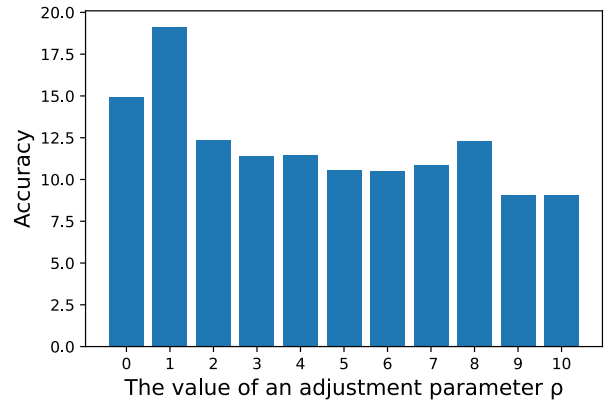
C. DENOISE PERFORMANCE ANALYSIS

We propose a TAD to clean the raw signal, which can improve classification accuracy. We use ρ in the thresholds formula, equation (14), as an adjustment parameter, which can affect the accuracy of the AMC. We investigate the impact of the adjustment parameter ρ on the classification accuracy of the AMC model. The results of the threshold experiments based on the baseline model TADCNN are shown in Figure 8(a). The purpose of denoiser is to enhance the accuracy of AMC models under conditions of low SNR. Therefore, we consider the accuracy of AMC with denoise in SNR = -20dB, shown in Figure 8(b). When the adjustment parameter ρ increases from 0 to 1, the accuracy rise by 4%. Unlikely, the accuracy decreases when ρ increases from 2 to 10. Especially when ρ is 9 and 10, the TADCNN model can not classify modulation. This is due to the fact that if the ρ value is large, then the threshold will also be large, leading to the input signal losing more information. For this reason, the ρ value should be from 1 to 2. Therefore, we set ρ to 1 for the TAD since it outperforms at low SNR.

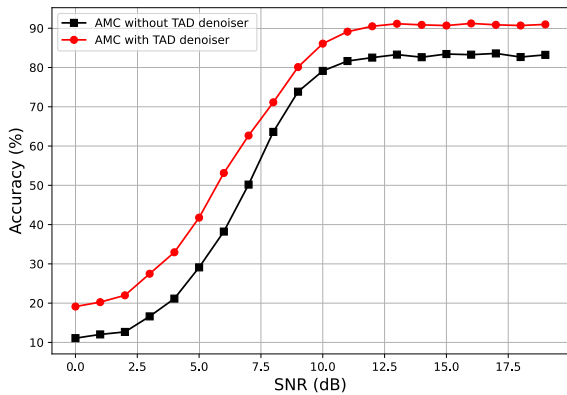
Figure 8(c) presents the effect of the TAD on the recognition accuracy of the proposed AMC. Without the TAD, the AMC model has a classification accuracy of 13% when SNR is around -20dB to -15dB. This accuracy is improved by 70% when it is combined with the TAD, which is shown



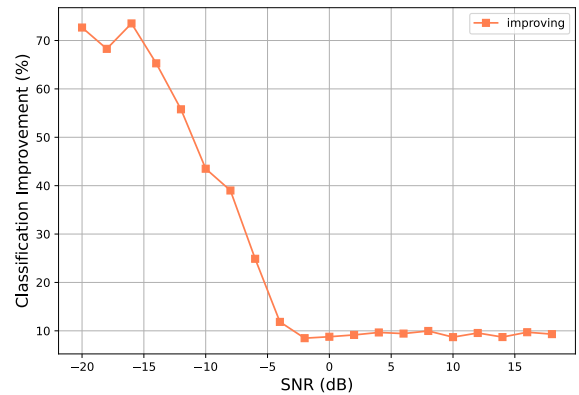
(a)



(b)



(c)



(d)

FIGURE 8. The threshold experiments based on the AMC model: (a) the accuracy of difference ρ , (b) the accuracy at SNR -20 dB, (c) the accuracy of AMC model with TAD and without TAD, (d) the improved accuracy of the AMC model when combined with TAD.

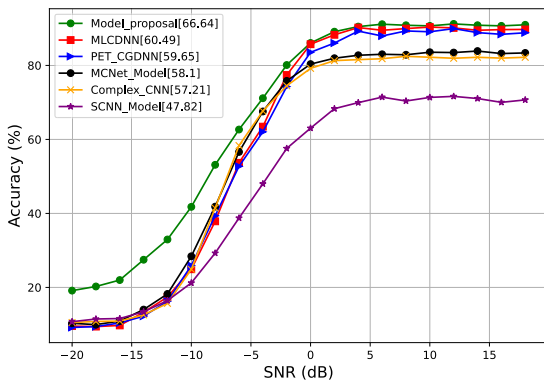


FIGURE 9. The classification accuracy of six different approaches. The average classification accuracy is represented by “****”.

in Figure 8(d). Similarly, TAD has also contributed to the improvement in the accuracy of AMC models in high SNR situations. At $+20$ dB, the architectures without the denoiser achieve a classification accuracy of around 83%, and the classification accuracy of the model rises from 83% to 91% when combining the TAD with the AMC model.

TABLE 5. Classification accuracy results of different models.

AMC Model	Classification Accuracy					Average
	-18dB	-10dB	0dB	10dB	18dB	
MCLDNN	0.09	0.25	0.86	0.90	0.90	60.49
PET-CGDNN	0.09	0.26	0.84	0.89	0.89	59.65
SCNN	0.1	0.18	0.61	0.69	0.68	47.21
MCnet	0.1	0.26	0.78	0.83	0.82	58.1
Complex CNN	0.1	0.22	0.79	0.82	0.82	57.21
Proposed (TADCNN)	0.21	0.41	0.86	0.91	0.91	66.64

Therefore, what stands out from the graph is that the architecture with the TAD achieves higher recognition accuracy in both cases and is especially important in the case of low SNR situations.

D. PERFORMANCE EVALUATION

In order to analyze the performance of TADCNN in terms of efficiency and complexity of modulation classification, we conduct a comparison with state-of-the-art AMC techniques. First of all, TADCNN is compared with five methods: Complex-CNN [45], MCnet [43], SCNN [34], MCLDNN [36], and PET-CGDNN [37]. For the sake of

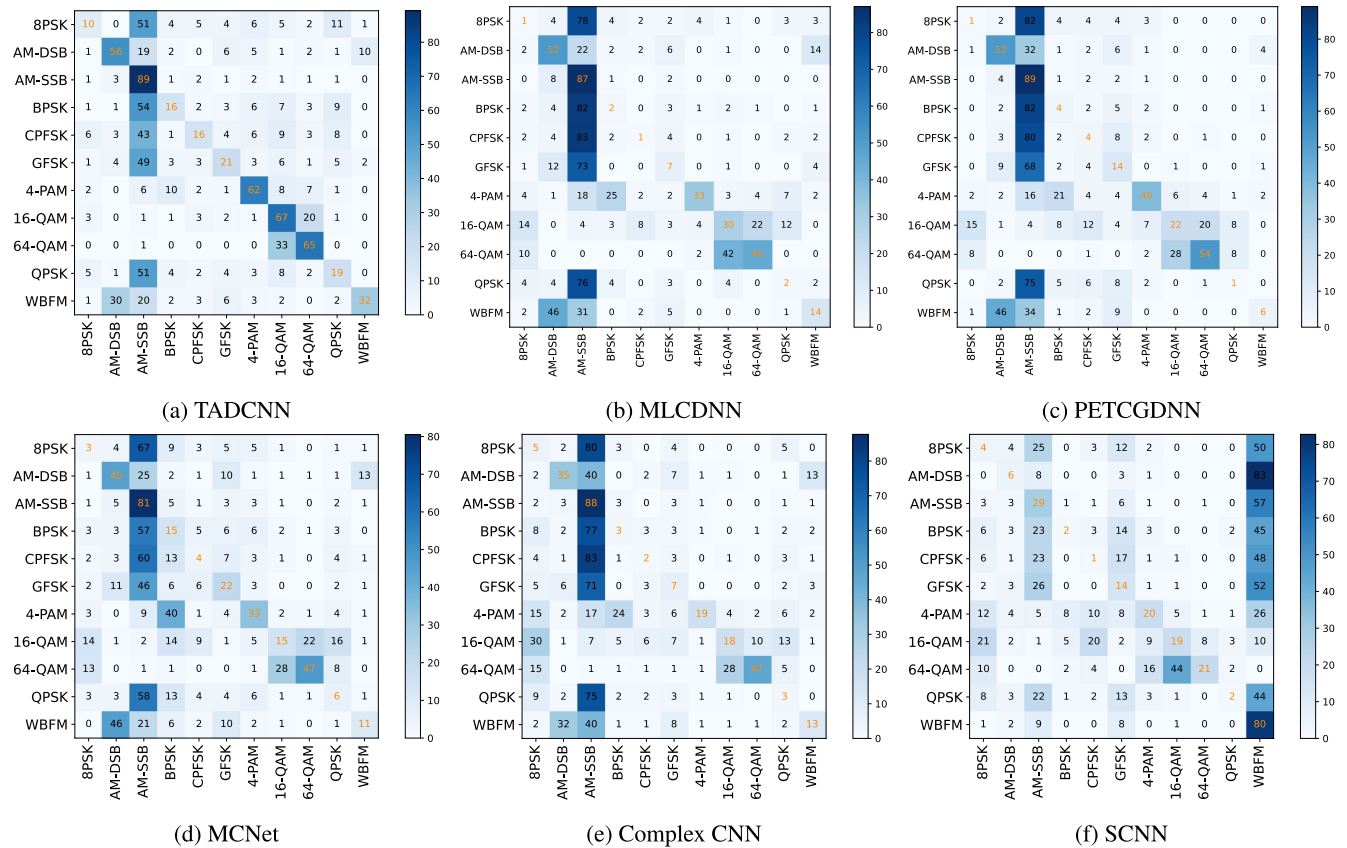


FIGURE 10. The confusion matrices of six AMC models at -10 dB SNR.

fair comparison, all models are trained and evaluated on the RadioML 2016.10A dataset in a similar implementation condition.

Table 5 compares different AMC methods at -18 dB, -10 dB, 0 dB, 10 dB, and 18 dB. It is evident from our results that the proposed TADCNN outperforms other AMC methods. The proposed method performs better than other methods at low SNR or high SNR. It should be noted that in low SNR, the proposed TADCNN method achieves two times as high as the accuracy of current AMC methods.

Figure 9 illustrates the classification performance of six models. The proposed TADCNN outperforms other methods of comparison at all SNRs. Additionally, our proposed TADCNN achieves an average classification accuracy of 66.64% at all SNRs, which is a 6.2%, 7%, 8.5%, and 9.4% improvement over MCLDNN, PET-GCDNN, MCnet, and Complex CNN respectively. Additionally, our model has a significantly higher accuracy score than SCNN by 18.8%.

Looking at the details in Figure 9, our model is significantly more accurate than the comparison methods at low SNR, from -20 dB to 0 dB. At -20 dB, for example, our model achieves nearly 20%, while the other approaches are nearly identical, 9 to 10%, which is twice as low as that of TADCNN. There may be a reason for this, as in conditions of low SNR, the noise power is large. Therefore, the signal of

modulation schemes is difficult to recognize, which reduces the overall classification accuracy.

On the other hand, the proposed model and MLCDDN produce similar results at 91% at high SNRs, for example, $SNR > 5$ dB, while MCnet, Complex CNN, and SCNN produce results ranging from 70% to 82%. When we test in high SNR conditions, the average power is large, which leads to the modulation scheme being easy to distinguish. Therefore, most of the models have performed well in High SNRs.

Figure 10 presents confusion matrices of all comparison models at -10 dB SNR. The TADCNN has the ability to boost the classification performance of all modulation modes. For instance, the accuracy of recognition reaches 16%, 12%, and 19%, respectively, for CPFSK, GFSK, and QPSK. In contrast, the other approaches are not able to classify the three modulation modes.

According to the study, 8PSK achieves the lowest accuracy of 10%, whereas other models achieve less than 5%. Next, the highest accuracy of TADCNN is observed for 16-QAM with an accuracy of 67%, which is confused with 64-QAM. Similar confusion exists in other models with 16-QAM and 64-QAM. The reason for this confusion is analyzed through signal visualization. Figure 11 shows the various signal constellations of the RadioML 2016.10A dataset at 12 dB. The signal constellations of each modulation type have

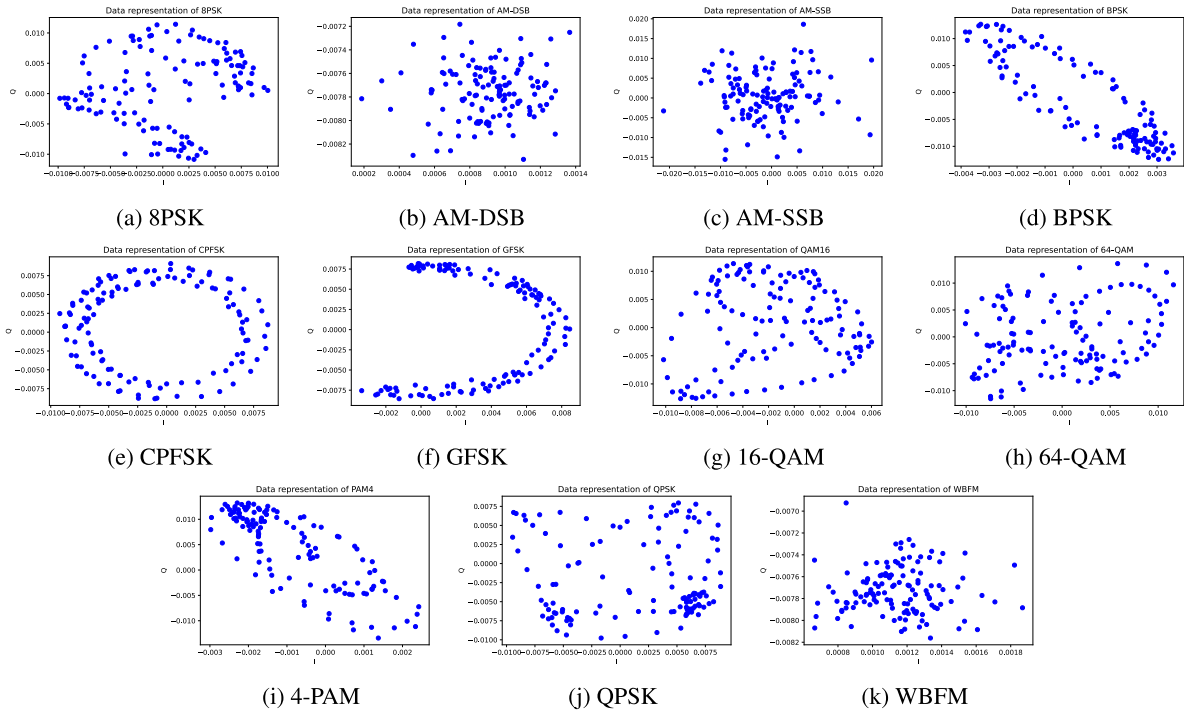


FIGURE 11. Signal constellation diagram.

different shapes, but the constellation diagrams of 16-QAM and 64-QAM are very similar, leading to low classification rates and confusion between 16-QAM and 64-QAM. In addition, AM-DSB and AM-SSB have some similarities in terms of their shape. Consequently, there is a small confusion regarding the classification of the two models is that the structures of the two modulations are highly similar, so it makes the AMC challenging to distinguish.

As well as evaluating the recognition accuracy, the performance of the proposed TADCNN is examined in terms of parameter numbers and the inference time, determining the complexity of an AMC model. In order to conduct a comprehensive complexity analysis, we rely primarily on Table 6 in our discussion. The table shows that the parameters for TADCNN are 671.4K, far fewer than those for ComplexCNN, which have 2.75M parameters. In spite of the fact that ComplexCNN has more parameters, it performs less well than our model. The two models with the lowest number of trainable parameters are PET-CGDNN and MCNet. The former has 71.9K parameters, and the latter has 90.8K parameters.

Furthermore, in terms of inference speed, the MCLDNN shows the highest inference time among the various models, which is 0.16 ms per sample. Although PET-CGDNN has the fewest parameters among the six models, it takes 0.063 ms as an inference time, which is the second-highest number. The inference speed of TADCNN is 0.045, which is over three times faster than MCLDNN. The SCNN model is the fastest due to its light structure. As a result, classification accuracy is the lowest.

TABLE 6. Computational Complexity of Different methods.

AMC Model	No. Parameters	Inference time per sample (ms)	Average accuracy(%)
MCLDNN	406,199	0.160	60.49
PET-CGDNN	71,871	0.063	59.65
SCNN	104,395	0.009	47.82
MCnet	90,763	0.021	58.1
Complex CNN	2,749,275	0.031	57.21
Proposed (TADCNN)	671,440	0.045	66.64

On the other hand, some recent works have been proposed to address the challenge associated with classifying modulation under varying noise conditions. The paper [35] proposed a novel technique to recognize modulation, which achieved great success. The proposed algorithm is able to determine the optimal threshold and most relevant features for splitting the wide-range SNR. After that, CNN is used to solve modulation classification tasks. It is capable of solving AMC tasks with higher classification accuracy while incurring lower computation complexity by utilizing a simpler CNN model. In contrast, our proposed AMC uses the denoise signal technique to overcome the challenge and also accomplishes remarkable results in classification, which is improved by 70% at low SNR.

VI. CONCLUSION

In this paper, we proposed a combination of threshold autoencoders and CNN architecture, denoted by TADCNN, in order to achieve high classification performance, especially in low SNR. The TAD was proposed to reduce noise power and improve TADCNN classification accuracy. When the model

with TAD is compared to the model without TAD, it can be found that at low SNR, the TAD model has been improved by 70%. Simulation results show that our method outperforms other comparison methods at all SNR levels. When SNR was -20dB to -15dB , the TADCNN's classification accuracy ranged from 19% to 25%, while the current AMC method performed just over 9%, far less than ours. In future work, we plan to reduce the complexity of TADCNN, but it still remains classification accuracy.

REFERENCES

- [1] Y. Wang, Q. Lu, Y. Jin, and H. Zhang, "Communication modulation signal recognition based on the deep multi-hop neural network," *J. Franklin Inst.*, vol. 358, no. 12, pp. 6368–6384, Aug. 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0016003221003604>
- [2] S. Kumar, R. Mahapatra, and A. Singh, "Automatic modulation recognition: An FPGA implementation," *IEEE Commun. Lett.*, vol. 26, no. 9, pp. 2062–2066, Sep. 2022.
- [3] H. B. Thameur, I. Dayoub, and W. Hamouda, "USRP RIO-based testbed for real-time blind digital modulation recognition in MIMO systems," *IEEE Commun. Lett.*, vol. 26, no. 10, pp. 2500–2504, Oct. 2022.
- [4] T. Huynh-The, T.-V. Nguyen, Q.-V. Pham, D. B. da Costa, G.-H. Kwon, and D.-S. Kim, "Efficient convolutional networks for robust automatic modulation classification in OFDM-based wireless systems," *IEEE Syst. J.*, early access, Oct. 3, 2022, doi: [10.1109/JSYST.2022.3207377](https://doi.org/10.1109/JSYST.2022.3207377).
- [5] 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.
- [6] H. B. Thameur and I. Dayoub, "Real-time in-lab test of eigenvalue-based spectrum sensing using USRP RIO SDR boards," *IEEE Commun. Lett.*, vol. 25, no. 3, pp. 1029–1032, Mar. 2021.
- [7] B. M. Lee and H. Yang, "Massive MIMO with massive connectivity for industrial Internet of Things," *IEEE Trans. Ind. Electron.*, vol. 67, no. 6, pp. 5187–5196, Jun. 2020.
- [8] 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.
- [9] K. M. Ho, C. Vaz, and D. G. Daut, "A wavelet-based method for classification of binary digitally modulated signals," in *Proc. IEEE Sarnoff Symp.*, Mar. 2009, pp. 1–5.
- [10] E. Moser, M. K. Moran, E. Hillen, D. Li, and Z. Wu, "Automatic modulation classification via instantaneous features," in *Proc. Nat. Aerosp. Electron. Conf. (NAECON)*, Jun. 2015, pp. 218–223.
- [11] 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.
- [12] C. Angelica, H. Purnama, and F. Purnomo, "Impact of computer vision with deep learning approach in medical imaging diagnosis," in *Proc. 1st Int. Conf. Comput. Sci. Artif. Intell. (ICCSAI)*, vol. 1, Oct. 2021, pp. 37–41.
- [13] H.-T. Nguyen, C. C. Cheah, and K.-A. Toh, "An analytic layer-wise deep learning framework with applications to robotics," *Automatica*, vol. 135, Jan. 2022, Art. no. 110007. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109821005331>
- [14] G. B. Tunze, T. Huynh-The, J.-M. Lee, and D.-S. Kim, "Sparsely connected CNN for efficient automatic modulation recognition," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15557–15568, Dec. 2020.
- [15] M. Zhang, Y. Zeng, Z. Han, and Y. Gong, "Automatic modulation recognition using deep learning architectures," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2018, pp. 1–5.
- [16] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu, "Data augmentation for deep learning-based radio modulation classification," *IEEE Access*, vol. 8, pp. 1498–1506, 2020.
- [17] Y. Guo, H. Jiang, J. Wu, and J. Zhou, "Open set modulation recognition based on dual-channel LSTM model," 2020, *arXiv:2002.12037*.
- [18] W. Shi, D. Liu, X. Cheng, Y. Li, and Y. Zhao, "Particle swarm optimization-based deep neural network for digital modulation recognition," *IEEE Access*, vol. 7, pp. 104591–104600, 2019.
- [19] W. Xie, S. Hu, C. Yu, P. Zhu, X. Peng, and J. Ouyang, "Deep learning in digital modulation recognition using high order cumulants," *IEEE Access*, vol. 7, pp. 63760–63766, 2019.
- [20] F. N. Khan, K. Zhong, W. H. Al-Arashi, C. Yu, C. Lu, and A. P. T. Lau, "Modulation format identification in coherent receivers using deep machine learning," *IEEE Photon. Technol. Lett.*, vol. 28, no. 17, pp. 1886–1889, Sep. 1, 2016.
- [21] R. Zhou, F. Liu, and C. W. Gravelle, "Deep learning for modulation recognition: A survey with a demonstration," *IEEE Access*, vol. 8, pp. 67366–67376, 2020.
- [22] R. Li, L. Li, S. Yang, and S. Li, "Robust automated VHF modulation recognition based on deep convolutional neural networks," *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 946–949, May 2018.
- [23] M. D. Ming Zhang and L. Guo, "Convolutional neural networks for automatic cognitive radio waveform recognition," *IEEE Access*, vol. 5, pp. 11074–11082, 2017.
- [24] Z. Zhang, C. Deng, W. Jin, and L. Peng, "Edge detection algorithm based on threshold function de-noising and wavelet neural network," in *Applied Mathematics, Modeling and Computer Simulation*. Amsterdam, The Netherlands: IOS Press, 2022, pp. 152–159.
- [25] S. Rodzin, V. Bova, Y. Kravchenko, and L. Rodzina, "Deep learning techniques for natural language processing," in *Proc. Comput. Sci. On-Line Conf.* Springer, 2022, pp. 121–130.
- [26] R. M. Samant, M. R. Bachute, S. Gite, and K. Kotecha, "Framework for deep learning-based language models using multi-task learning in natural language understanding: A systematic literature review and future directions," *IEEE Access*, vol. 10, pp. 17078–17097, 2022.
- [27] G. Truong, H. Le, E. Zhang, D. Suter, and S. Z. Gilani, "Unsupervised learning for maximum consensus robust fitting: A reinforcement learning approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 27, 2022, doi: [10.1109/TPAMI.2022.3178442](https://doi.org/10.1109/TPAMI.2022.3178442).
- [28] L. H. Pham, D. Nguyen-Ngoc Tran, H.-H. Nguyen, T. H.-P. Tran, H.-J. Jeon, H.-M. Jeon, and J. W. Jeon, "DeepACO: A robust deep learning-based automatic checkout system," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3107–3114.
- [29] K. Alibabaei, E. Assunção, P. D. Gaspar, V. N. G. J. Soares, and J. M. L. P. Caldeira, "Real-time detection of vine trunk for robot localization using deep learning models developed for edge TPU devices," *Future Internet*, vol. 14, no. 7, p. 199, Jun. 2022.
- [30] K. Yashashwi, A. Sethi, and P. Chaporkar, "A learnable distortion correction module for modulation recognition," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 77–80, Feb. 2019.
- [31] J. Shi, S. Hong, C. Cai, Y. Wang, H. Huang, and G. Gui, "Deep learning-based automatic modulation recognition method in the presence of phase offset," *IEEE Access*, vol. 8, pp. 42841–42847, 2020.
- [32] J. Nie, Y. Zhang, Z. He, S. Chen, S. Gong, and W. Zhang, "Deep hierarchical network for automatic modulation classification," *IEEE Access*, vol. 7, pp. 94604–94613, 2019.
- [33] Y. Zeng, M. Zhang, F. Han, Y. Gong, and J. Zhang, "Spectrum analysis and convolutional neural network for automatic modulation recognition," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 929–932, Jun. 2019.
- [34] X. Fu, G. Gui, Y. Wang, T. Ohtsuki, B. Adebisi, H. Gacanin, and F. Adachi, "Lightweight automatic modulation classification based on decentralized learning," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 1, pp. 57–70, Mar. 2022.
- [35] B. Jdid, W. H. Lim, I. Dayoub, K. Hassan, and M. R. B. Mohamed Juhari, "Robust automatic modulation recognition through joint contribution of hand-crafted and contextual features," *IEEE Access*, vol. 9, pp. 104530–104546, 2021.
- [36] J. Xu, C. Luo, G. Parr, and Y. Luo, "A spatiotemporal multi-channel learning framework for automatic modulation recognition," *IEEE Wireless Commun. Lett.*, vol. 9, no. 10, pp. 1629–1632, Oct. 2020.
- [37] F. Zhang, C. Luo, J. Xu, and Y. Luo, "An efficient deep learning model for automatic modulation recognition based on parameter estimation and transformation," *IEEE Commun. Lett.*, vol. 25, no. 10, pp. 3287–3290, Oct. 2021.
- [38] L. Huang, Y. Zhang, W. Pan, J. Chen, L. P. Qian, and Y. Wu, "Visualizing deep learning-based radio modulation classifier," *IEEE Trans. Cognit. Commun. Netw.*, vol. 7, no. 1, pp. 47–58, Mar. 2021.

- [39] S. Hanna, C. Dick, and D. Cabric, "Combining deep learning and linear processing for modulation classification and symbol decoding," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–7.
- [40] J. Lei, J. Zhou, Z. Jian, H. Liu, L. Zhang, Y. Feng, and Z. Wu, "A Wiener filter denoising based intelligent modulation recognition system," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2022, pp. 88–93.
- [41] H. Ma, G. Xu, H. Meng, M. Wang, S. Yang, R. Wu, and W. Wang, "Cross model deep learning scheme for automatic modulation classification," *IEEE Access*, vol. 8, pp. 78923–78931, 2020.
- [42] H. Wang, Z. Wu, S. Ma, S. Lu, H. Zhang, G. Ding, and S. Li, "Deep learning for signal demodulation in physical layer wireless communications: Prototype platform, open dataset, and analytics," *IEEE Access*, vol. 7, pp. 30792–30801, 2019.
- [43] T. Huynh-The, C. Hua, Q. Pham, and D. Kim, "MCNet: An efficient CNN architecture for robust automatic modulation classification," *IEEE Commun. Lett.*, vol. 24, no. 4, pp. 811–815, Apr. 2020.
- [44] M. Zhao, S. Zhong, X. Fu, B. Tang, and M. Pecht, "Deep residual shrinkage networks for fault diagnosis," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4681–4690, Jul. 2020.
- [45] J. Krzyston, R. Bhattacharjea, and A. Stark, "Complex-valued convolutions for modulation recognition using deep learning," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6.
- [46] H. Zhang, Y. Wang, L. Xu, T. A. Gulliver, and C. Cao, "Automatic modulation classification using a deep multi-stream neural network," *IEEE Access*, vol. 8, pp. 43888–43897, 2020.
- [47] T. O'Shea and N. West, "Radio machine learning dataset generation with gnu radio," in *Proc. GNU Radio Conf.*, 2016, vol. 1, no. 1, pp. 1–6. [Online]. Available: <https://pubs.gnuradio.org/index.php/grcon/article/view/11>
- [48] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.



TO TRUONG AN received the B.S. degree in mechanical engineering from the Ho Chi Minh City University of Technology and Education, Vietnam, in 2016. He is currently pursuing the master's degree with Sejong University, Seoul, South Korea. He had two years of industry experience as a Research Engineer at Samsung Electronics, Ho Chi Minh, Vietnam, in 2017. His research interests include machine learning and deep learning in wireless communication networks, 5G, 6G, and signal processing, with an emphasis on deep learning for wireless communications.



BYUNG MOO LEE (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of California at Irvine, Irvine, CA, USA, in 2006. He had ten years of industry experience including research positions at the Samsung Electronics Seoul Research and Development Center, Samsung Advanced Institute of Technology, Suwon, South Korea, and the Korea Telecom Research and Development Center, Seoul, South Korea. He is currently an Associate Professor with the Department of Intelligent Mechatronics Engineering, Sejong University, Seoul. During his industry experience, he participated in IEEE 802.16/11, Wi-Fi Alliance, and 3GPP LTE standardizations, and also participated in Mobile VCE and Green Touch Research Consortiums, where he made numerous contributions and filed a number of related patents. His research interests include wireless communications, signal processing, and machine learning applications. He was the Vice Chairperson of the Wi-Fi Alliance Display MTG from 2015 to 2016.

• • •