IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 9, 2021, accepted July 19, 2021, date of publication July 26, 2021, date of current version July 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3099222

Robust Automatic Modulation Recognition Through Joint Contribution of Hand-Crafted and Contextual Features

BACHIR JDID[1], WEI HONG LIM[1], (Senior Member, IEEE), IYAD DAYOUB[2,3], (Senior Member, IEEE), KAIS HASSAN[4], AND MOHD RIZON BIN MOHAMED JUHARI[1]

[1]Faculty of Engineering Technology and Built Environment, UCSI University, Kuala Lumpur 56000, Malaysia
[2]Université Polytechnique Hauts-de-France, CNRS, University of Lille, ISEN, Centrale Lille, UMR 8520, Institut d'Électronique de Microélectronique et de Nanotechnologie (IEMN), Département d'Opto-Acousto-Électronique (DOAE), F-59313 Valenciennes, France
[3]INSA des Hauts de France, F-59313 Valenciennes, France
[4]Laboratoire d'Acoustique de l'Université du Mans (LAUM), UMR CNRS 6613, Le Mans University, 72085 Le Mans, France

Corresponding authors: Mohd Rizon Bin Mohamed Juhari (mohdrizon@ucsiuniversity.edu.my) and Iyad Dayoub (iyad.dayoub@uphf.fr)

This work was supported by UCSI University through the UCSI University Research Excellence and Innovation Grant Scheme under Project REIG-FETBE-2020/025.

ABSTRACT Automatic modulation recognition (AMR) has become increasingly important in the field of signal processing, especially with the advancements of intelligent communication systems. Deep Learning (DL) technologies have been incorporated into the AMR field and they have shown outstanding performances against conventional AMR methods. The robustness of DL-based AMR methods under varying noise regimes is one of major concerns for the widespread utilization of this technology. Furthermore, most existing works have neglected the contributions of hand-crafted features (HCFs) in boosting the classification performances of DL-based AMR methods. In order to address the aforementioned technical challenges, a novel and robust DL-AMR method is proposed by leveraging the benefits of both contextual features (CFs) and HCFs for a specific range of signal-to-noise ratio (SNR). A novel feature selection algorithm is also proposed to search for the optimal sets of HCFs in order to reduce the dimensions of feature vectors without losing any important and relevant features. Simulation studies are performed to investigate the feasibility of proposed method in classifying 11 types of modulation schemes. Extensive performance analyses revealed the superiority of proposed method over baseline method in terms of the classification performance as well as the excellent capability of proposed feature selection algorithm in determining an optimal subset of HCFs.

INDEX TERMS Automatic modulation recognition (AMR), convolutional neural network (CNN), deep learning (DL), wireless signal classification.

I. INTRODUCTION

For the communication systems widely used in both military and civilian applications, the radio signals are encoded by predefined adaptive modulation schemes with respect to the specification of transmission channel. Receiver needs to correctly identify the types of modulation scheme adopted in order to ensure successful demodulation. Most often, the receiver has no prior information about the modulation scheme of received signals in blind detection. Automatic modulation recognition (AMR) is a popular technique used to provide the blind recognition of the modulation scheme. In literatures, existing AMR techniques are designed and implemented based on two main approaches: (1) likelihood-based (LB) and (2) feature-based (FB) [1]. Despite being able to achieve the optimum recognition rate, most LB approaches tend to suffer with technical drawbacks such as high computational complexity and strong dependency on the prior information of received signal [1]. In contrary to LB approaches, FB approaches are considered as the more practical solutions due to their ease of implementation for real-life scenario and ability to produce sub-optimal solutions. It is notable that the performance of FB approaches relies tremendously on the hand-crafted features (HCFs) manually extracted from

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

the received signals. Different combinations of HCFs are also found to have crucial roles in recognizing different types of modulation schemes. The incorporation of more advanced feature-learning methods for deep feature extraction is necessary to develop the more robust LB techniques in solving AMR problems.

Recently, different deep learning (DL) models such as the convolutional neural network (CNN), recurrent neural network (RNN), deep neural network (DNN) and etc. have shown their promising advantages to tackle the key emerging issues in research fields of computer vision, healthcare, robotics, internet of things, and communication systems. For instance, CNN is widely used as an image recognition method to solve multiple-object detection problems due to its promising ability to automatically extract multiple levels of features from inputs. Motivated by the successes of DL in different areas, numerous DL-based AMR techniques have been proposed based on CNN, RNN, DNN and other network architectures in recent years and they have been observed to achieve significant results [1].

Jdid *et al*. [1] presented a comprehensive survey on existing AMR methods developed with different machine learning (ML) and DL models. They compared these ML-based and DL-based AMR methods in terms of their network structures, training and testing hyperparameters, noise conditions, modulation pools and performances. O'Shea and West [2] employed GNU Radio to generate a synthetic dataset known as the RadioML 2016.10A that contains eleven types of digital and analog modulations. A new AMR method was developed by O'Shea and Corgan [3] by using CNN and its performance was investigated with the dataset proposed in [2]. O'Shea *et al.* [4] amended the tool described in [2] to generate RadioML 2018.01A and more details of this new dataset will be summarized in Section II-B. Their previous works [3], [5], [6] with adaptation of CNNs for AMR were extended to solve the new dataset, where in-depth investigation in terms of the design parameters, channel impairment and training dataset parameters were studied through various simulations including over-the-air measurement of AMR performance.

Another CNN-based AMR was proposed by Zhang *et al.* [7] by employing the Short-Time Fourier Transform (STFT) to convert the input signal spectrograms into image representation. Similarly, Sun *et al*. [8] adopted an image dataset with 10 different modulation schemes to build a CNN-based AMR from a popular deep learning model known as VGG-16 [9]. Zhang *et al.* [10], applied both of Wigner-Ville and Born-Jordan distributions to convert the received signals into two types of images. These converted images were then fed into two CNNs for feature extraction before fusing them with the selected HCFs in order to solve the classification problems.

The work reported by Peng *et al.* [11] exploited the adaptation of constellation diagrams and a popular CNN model called AlexNet [12] for AMR. Similarly, constellation diagrams were employed as the inputs of feature extractor in [13], whereas a support vector machine (SVM) was used as the classifier for AMR. Given that the qualities of datasets are crucial to improve the performance of AMR, the pre-processing technique such as data augmentation was employed by Tang *et al.* [14] to ensure sufficient data were available to construct the datasets required for the training and testing of DL-based AMRs. Particularly, constellation diagrams were converted into the contour stellar images in [14] to obtain more color features than those of [11]. Several issues encountered in training process such as the divergence of generator, overfitting of discriminator and mode collapse can degrade the overall performance of traditional generative adversarial nets [15]. Numerous measures were proposed to address the poor performance issues of GANs and simulation studies revealed that the classification accuracies can be improved through the extension of datasets.

Most existing FB-AMR approaches proposed in earlier era [16] were designed based on the extraction of HCFs. Meanwhile, the DL-based AMR approaches proposed in recent literatures were found to heavily rely on the ideas of using deep neural network architecture for feature extraction [17]. Extensive amounts of previous research works in image processing have shown the advantages of concatenating the contextual features (CFs) and HCFs [18]–[20]. In particular, a proper combination between HCFs with CFs is able to achieve the performance boosting via more diverse representation of features. To the best of authors' knowledge, limited investigations were performed to search for the optimal combinations of CFs and HCFs. This is because the quality of CFs is governed by the efficiency and abundance of the used datasets. Meanwhile, the value of each HCF may change with different signal-to-noise ratio (SNR) values. It is also noteworthy that some HCFs tend to have the same values for different types of modulations. These undesirable characteristics of HCFs tend to jeopardize the accuracies of classifiers through the distortion of their hyperplanes. In order to address these technical drawbacks, it is crucial to design an efficient feature selection algorithm that is able to build the robust feature set.

As mentioned earlier, the performances of DL-based AMR methods are governed by the qualities of datasets, hence the latter should be constructed carefully to include a varsity of real-life effects. While the larger number of training examples can lead to better performance of DL-based AMR methods, the involved training processes tend to be time-consuming. This issue may be resolved by splitting the original large dataset into multiple subsets of data for the training process. The hyperplanes of classifiers also tend to be distorted when dealing with datasets that have very wide ranges of SNR. Although training a classifier with the wide range of SNR can eliminate the requirements of SNR estimation, CNNs with highly complex network structures are required to extract the relevant features. The required memory storage of classifiers could increase significantly due the tremendous amounts of training parameters involved in its feature extractor. In this paper, a novel DL-based AMR method is proposed to tackle

the aforementioned issues, where the original SNR range is spilt into multiple ranges in order to train the good performing classifiers with lower complexity and smaller storage requirements.

The main purpose of AMR is to configure the parameters and modulation schemes of received signals in order to achieve a successful demodulation process. Greater accuracy of AMR process is expected to achieve through the incorporation of DL due to its excellent feature learning capability. This paper is devoted to study the feasibility of combining the robust set of HCFs with CFs in boosting the classification performance. Furthermore, this work also aims to investigate the performance differences of classification accuracy between simple CNNs trained with smaller SNR ranges and complex CNN trained with wider SNR ranges. For the former case, a set of HCFs are first constructed and the original SNR range is spilt into two smaller SNR ranges based on a predefined threshold. Then, these two sets of HCFs are concatenated with the CFs learned by the two CNNs deployed in their respective SNR ranges.

The main contributions of this paper are summarized as follows:

- Introduce new compositions of High-Order Cumulants (HOCs) that can be used to provide good complement for CFs in order to boost the classification performance.
- Propose a novel algorithm that is able to determine the optimal threshold and most relevant features for splitting the wide-range SNR.
- Introduce a novel criterion known as Classification Confusion Index (CCI) which is the main core of the proposed algorithm that is able to select an optimal subset of HCFs to be used for solving the classification tasks.
- Propose a novel DL-based AMR method that is able to achieve high classification accuracy without compromising the desirable characteristics of low computational complexity and storage requirement.

The findings of this paper are also expected to provide constructive insights for the following key questions:

- Is it efficient to spilt a wide-range of SNR into two smaller ranges?
- Can a narrower range of SNR reduce the complexity and depth of CNN architecture in learning more meaningful CFs?
- Is the presence of HCFs able to boost classification performance of a DL-based AMR method?

This work focuses on employing simple CNN architectures for learning CFs and concatenating these features with a set of HCFs in a wide-range of SNR that is split into two different ranges. The outline of this paper is summarized as follows. The background of AMR and specifications of RadioML 2018.01A dataset considered in current study are presented in Section II. Other required information such as the signal parameters and channel characteristics are also covered in this section. Section III explains the algorithm used to extract and select an optimal subset of HCFs, whereas the extraction process of CFs using CNNs are elaborated in Section IV. The overall mechanisms of proposed DL-based AMR method are then explained in Section V and the performance comparisons with baseline methods are presented in Section VI. Finally, the concluding remarks of current study are presented in Section VII.

II. AMR BACKGROUND

AMR is an essential process used to identify the type of modulation scheme adopted by received signals before recovering these received information through the demodulation process. In this section, a brief background of FB-AMR is first presented. Then, we wrap up this section by providing an overview of the most popular signals dataset, RadioML 2018.01A [4], in AMR field.

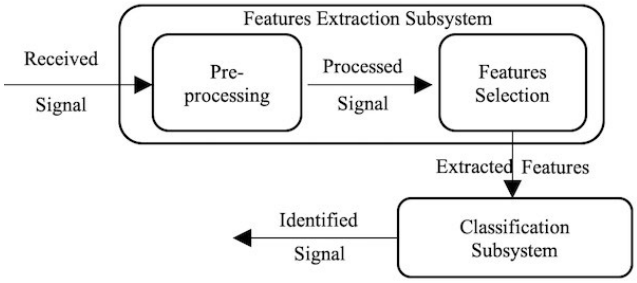

FIGURE 1. Block diagram of the Feature-based AMR method.

A. FB-AMR BACKGROUND

Fig. 1 shows the general block diagram of conventional FB-AMR methods that consist of three main phases:

- Pre-processing phase: Several variables such as Carrier Frequency Offset (CFO), baud rate, Phase Offset (PO), SNR and timing offsets can be obtained from the input signals via different pre-processing techniques. The quality of received signal can be enhanced in this phase before proceeding to feature extraction phase.
- Feature extraction phase: Feature extraction is first performed in this phase to obtain the HCFs or CFs. These extracted features can also be categorized as instantaneous time features, wavelet features and statistical features. Feature selection algorithm is subsequently used to choose the most relevant features in order to reduce the dimensional size of feature vector.
- Classification phase: This phase is responsible to decide the types of modulation scheme adopted by received signal with one or multiple classifiers that can be broadly categorized as traditional classifiers (e.g., classification tree), ML-based classifiers (e.g. SVM) and DL-based classifiers (e.g., CNN).

B. SIGNALS DATASET RadioML 2018.01A

The RadioML 2018.01A dataset proposed by O'Shea *et al.* [4] is considered in this study because it is one of the most challenging datasets for AMR problem.

TABLE 1. Summary of the parameters of RadioML 2018.01A Dataset.

Key	Value
Normal modulation pool (P)	{OOK, 4ASK, BPSK, QPSK, 8PSK, 16QAM, AM-SSB-SC, AM-DSB-SC, FM, GMSK, OQPSK}
Difficult modulation pool (P^d)	{OOK, 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APSK, 32APSK, 64APSK, 128APSK, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, AM-SSB-WC, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC, FM, GMSK, OQPSK}
SNR minimum value (S_n^{min})	$S_n^{min} = -20\ dB$
SNR maximum value (S_n^{max})	$S_n^{max} = +30\ dB$
SNR step	$2\ dB$
Number of SNR values (N_{snr})	$N_{snr} = (S_n^{max} - S_n^{min})/2 + 1 = 26$
Signal length (l)	$l = 1,024$
Total number of signals (N)	$N = 2,555,904$
Number of signals per type (N_t)	$N_t = 106,496$
Number of signals per type per SNR (N_t^s)	$N_t^s = 4,096$

This dataset consists of two different compositions of classes spread across a wide range of SNR values. The "Normal" classes of RadioML 2018.01A dataset consists of 11 modulation schemes commonly seen in the impaired environments, whereas the "Difficult" classes contain the 24 digital and analog modulation schemes. It is also notable that the RadioML 2018.01A dataset contains more than 2.5 million frames of modulation signals along with different synthetic simulated channel effects such as CFO, symbol rate offsets, delay spread and thermal noise. In this paper, the performance of proposed DL-based AMR method will be evaluated using the "Normal" classes. For the self-contained purpose, the parameters of RadioML 2018.01A dataset are presented in Table 1.

III. HCFs EXTRACTION

Extensive literature studies revealed that most existing FB-AMR methods employ different types of HCFs such as wavelet feature [21], spectral-based features [22], time-frequency features [23], [24] [25], instantaneous features [1] and statistical features [1]. The derivations of these HCFs are further explained in the following subsections.

A. INSTANTANEOUS FEATURES

Several instantaneous features can be extracted by employing specific parameters such as the instantaneous amplitude a_n, instantaneous phase ϕ_{NL} and instantaneous frequency f_N. Table 2 presents the derivation of the five instantaneous features [26] considered in this work.

B. HIGH ORDER STATISTICS FEATURES

Both of the Higher-Order Moments (HOMs) and HOCs [27], [28] are reported to be the best candidates for signal recognition. Mathematically, the HOMs of a signal x are

TABLE 2. The mathematical equations of the extracted instantaneous features.

Feature	Equation
Envelope variation	$m_A = \frac{\sigma^2}{\mu^2}$ (1) where σ and μ are the variance and mean, respectively, of a_n.
Standard deviation of the amplitude envelope	$E = \sqrt{\frac{1}{N_s - 1}\sum_{i=1}^{N_s}\left(a_n(i) - \frac{1}{N_s}\sum_{i=1}^{N_s} a_n(i)\right)^2}$ (2) where N_s is the sample number.
Kurtosis of the normalized-centered instantaneous amplitude	$\mu_{42}^a = \frac{E\left(a_{cn}^4[n]\right)}{\left(E\left\{a_{cn}^2[n]\right\}\right)^2}$ (3) where a_{cn} is the normalized and centered instantaneous amplitude of the incoming signal at the time instant. $a_{c_n} = a_n(i) - 1$, $a_n(i) = a_n(i)/m_a$, $m_a = 1/N_s . \sum_{i=1}^{N_s} a_n(i)$
Standard deviation of the absolute value of the normalized-centered instantaneous amplitude	$\sigma_{aa} = \sqrt{\frac{1}{N_i}\left(\sum_{i=1}^{N_s} a_{cn}^2(i)\right) - \left(\frac{1}{N_s}\sum_{i=1}^{N_s}\lvert a_{cn}(i)\rvert\right)^2}$ (4)
Maximum value of the power spectral density of the normalized-centered instantaneous amplitude	$\gamma_{\max} = \frac{\mathrm{MAX}\left(\lvert DFT\left[a_{cn}(i)^2\right]\rvert\right)}{N_s}$ (5) where DFT refers to the discrete Fourier transform of modulated signal.

defined as:

$$M_{km}(x) = E[x^{k-m}(x^*)^m] \tag{6}$$

where k is the order of the moment. The cumulant of order k of the zero-mean signal x is defined as:

$$C_{km}(\mathrm{x}) = \mathrm{Cum}[\underbrace{x, \ldots, x}_{(k-m)\text{ times}}, \underbrace{x^*, \ldots, x^*}_{m\text{ times}}] \tag{7}$$

The mathematical relation between cumulants and moments can be expressed as:

$$\mathrm{Cum}\left[x_1, \ldots, x_n\right] = \sum_{\Phi}(c-1)!(-1)^{c-1}\prod_{v\in\Phi} E\left(\prod_{i\in v} x_i\right) \tag{8}$$

where $\phi \in \{1, \ldots, n\}$, v refers to the list member in the partition ϕ; c is the number of elements in the partition ϕ. For instance, the cumulants of up to 6^{th} orders are defined as follows [26]:

$$C_{20} = M_{20} \tag{9}$$
$$C_{21} = M_{21} \tag{10}$$
$$C_{40} = M_{40} - 3M_{20}^2 \tag{11}$$
$$C_{41} = M_{41} - 3M_{21}M_{20} \tag{12}$$
$$C_{42} = M_{42} - |M_{20}|^2 - 2M21^2 \tag{13}$$
$$C_{60} = M_{60} - 15M_{40}M_{20} + 30M_{20}^3 \tag{14}$$
$$C_{63} = M_{63} - 9M_{41}M_{21} - 6M_{21}^3 \tag{15}$$

It is evident from (9) to (15) that the estimation of moments will lead to the estimation of the cumulants as well.

However, given a signal x with N samples, the moments are estimated as:

$$\widehat{\mathrm{M}}_{\mathrm{km}}(\mathrm{x}) = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} x^{\mathrm{k-m}}(\mathrm{i}) x^{*\,\mathrm{m}}(\mathrm{i}) \tag{16}$$

Without loss of generality, the normalized signal x is assumed to have a unity energy, i.e., $C_{21} = 1$. The normalization process is used to address the scaling issues of estimators. In practical, the self-normalized HOMs and HOCs are calculated as:

$$\tilde{M}_{km}(x) = \hat{M}_{km}(x)/\hat{M}_{21}^{k/2}(x) \tag{17}$$

$$\tilde{C}_{km}(x) = \hat{C}_{km}(x)/\hat{C}_{21}^{k/2}(x) \tag{18}$$

Estimating a moment of order k requires only around N complex addition and $k \times N$ complex multiplications. Theoretically, when k is close to N, the complexity of cumulant estimation is of order N^2. Practically, the cumulant order k is far smaller than N and, thus, the complexity of cumulant estimation will be of order N [16]. Notably, the computational cost incurred by feature calculation is of the same order with those in estimation of cumulants and moments. Therefore, the feature extraction process is considered to have very low complexity of $O(N)$ in practical scenarios [16]. In this work, we introduce a total of ten new cumulants compositions. These compositions were found by some simple experiments and they are defined as follows:

$$P_1 = |C_{20}|/|C_{21}| \tag{19}$$

$$P_2 = |C_{20}|/|C_{40}| \tag{20}$$

$$P_3 = |C_{20}|/|C_{C41}|^{\frac{1}{2}} \tag{21}$$

$$P_4 = |C_{21}|/|C_{62}|^{\frac{1}{3}} \tag{22}$$

$$P_5 = |C_{21}|/|C_{80}|^{\frac{1}{4}} \tag{23}$$

$$P_6 = |C_{63}|^{\frac{1}{3}}/|C_{80}|^{\frac{1}{4}} \tag{24}$$

$$P_7 = |C_{40}|/|C_{60}|^{\frac{1}{3}} \tag{25}$$

$$P_8 = \log|C_{20}| \tag{26}$$

$$P_9 = \log|C_{21}| \tag{27}$$

$$P_{10} = \log|C_{41}| \tag{28}$$

To this end, a total of 15 HCFs are obtained to construct the original HCFs set denoted as F, where:

$$F = \{m_A, E, \mu_{42}^a, \sigma_{aa}, \gamma_{\max}, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}\} \tag{29}$$

IV. CFs EXTRACTION USING CNNs

CNN is a popular deep learning architecture used to automatically extract CFs from the input data. A typical CNN architecture consists of several convolution (Conv) and pooling (Pool) layers connected in series along with at least one fully-connected (FC) layer. The depth of CNN architecture increases with numbers of convolution, pooling and fully-connected layers.

The convolution layers of CNN play essential role in feature extraction of input data, where these features can be learnt from a set of convolutional filters constructed by a group of neurons that are arranged as the rectangular gird. When the 2-dimensional (2D) input data are forward passed within CNN, a set of 2D activation maps are produced by sliding all convolution filters across the input data. During the training of CNN, the weight values of convolution filters can be updated through the forward propagation and backpropagation of error. This enables the activation of learned filters when encountering desired types of input signals.

Pooling layers are commonly located between the single or multiple convolutional layers. Max pooling and average pooling are two popular pooling layers adopted. The former pooling takes the maximum values within the kernels, whereas the latter one considers the average values within the kernels. The pooling layers of CNN can avoid network overfitting to certain extent by reduce the dimension sizes of feature progressively. The presence of pooling layers also enables CNN to becomes invariant to small translations of input, therefore improving its network accuracy. Finally, the fully connected layers are used as the classifier part of CNN and they consist of several nodes connected to all activations received from previous layer.

An appropriate activation function needs to be defined for all CNN layers except for its input layer. Both of the rectified linear unit (ReLU) and parametric rectified linear unit (PReLU) are commonly used in CNN by executing a threshold operation on each input element. Softmax is another activation function widely used to solve the multiclassification problems by normalizing the input vectors into a probability distribution consisting of several probabilities proportional to the exponentials of the input numbers and maps them into the (0, 1) interval.

V. THE PROPOSED DL-BASED AMR TECHNIQUE

This section is organized in a top-down manner. An overview of the proposed DL-based AMR method is first presented, followed by the explanations of technical details for each compositional sub-system in the subsequent subsections. Fig. 2 shows the block diagram of the proposed DL-based AMR method. Initially, the received signals are fed into feature extraction subsystem, where three main groups of HCFs are generated. The first group (G_1) is used for SNR splitter,[1] while the other two groups (G_2 and G_3) are used along with CFs for modulation schemes classification. For the HCFs belonging to G_1 that are fed into the SNR splitter sub-system, a specific threshold (Thr) is defined to divide the overall SNR range (R) into two smaller ranges of R_1 and R_2, where $R = R_1 \cup R_2$. Depending on the identified SNR range, these signals are then fed into the corresponding CNN for CFs extraction. The CFs extracted from CNN_1 and CNN_2 are then concatenated with the G_2 and G_3 groups respectively. Finally, the classification of modulation schemes can be performed by one of the SVMs denoted as SVM_1 and SVM_2 that are deployed at both SNR ranges, R_1 and R_2, respectively.

[1] An explanation for this will be provided later in section V-A4

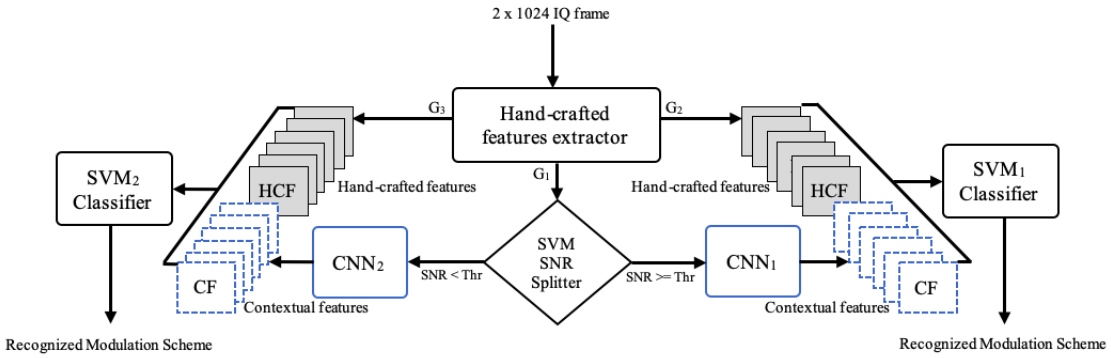

FIGURE 2. Block diagram of the proposed DL-based AMR technique.

A. NOVEL ALGORITHM FOR SNR WIDE-RANGE SPLITTING

In this subsection, we propose a novel algorithm to split the wide range of SNR values into two smaller ranges based on a set of HCFs and an SVM classifier. The SVM classifier that can be utilized as a SNR splitter is first presented, followed by the descriptions of novel algorithm used to select the most relevant HCFs by accurately splitting the overall SNR range based on the most effective splitting threshold. The proposed algorithm is then evaluated by using both "Normal" and "Difficult" classes. Finally, a new training method known as cross-SNR training is presented to reduce the classification errors produced by SNR splitter in order to enhance the classification accuracy of the proposed DL-based AMR method.

1) SNR SPLITTER

In practical, SVM classifiers can provide good computation speed and memory, and work relatively well when there is a clear margin of separation between few classes. Therefore, a fast linear SVM classifier is adopted as the SNR splitter of proposed DL-AMR method. For each signal obtained from the dataset, a vector of HCFs denoted as G_1 is extracted. In offline training, all HCFs vectors that are extracted from the signals with SNR values lesser than a predefined threshold (Thr) are labeled as L_1, whereas those with SNR values greater than Thr are labeled as L_2. After the SVM is trained, it will be integrated in the proposed AMR method as the SNR splitter. It will identify the range of SNR value of the received signals before HCFs and CFs extraction system. The complete pseudo-code of SNR splitter is presented using Algorithm 1. It is worth mentioning that the same criterion is used to label the modulation schemes of all signals with L_1 and L_2.

2) NOVEL HCFs SELECTION ALGORITHM FOR SPLITTING SNR RANGE

The total SNR range are divided into several consecutive ranges to investigate the variation of each HCF value under these ranges. This procedure aims to select the best features and threshold values that can split the overall SNR range into smaller ranges with higher splitting accuracy. The Coefficient of Variation (CV) measure is used to demonstrate the variation of each HCF value under the varying SNR conditions.

CV is defined as the ratio of standard deviation σ to mean μ and it is used to measure the dispersion of data sample around the mean value of population. CV is able to compare the degree of variations between different data series efficiently even through their mean values are drastically different from one another [29]. It is also notable that CV is sensitive to small changes in the mean if the latter has near zero value. For a group of signals denoted as S^m, the CV of k-th feature (f_k) is

Algorithm 1 Pseudo Code for SNR Splitter Function

Function `SNR-Splitter` $(F_p, N_{snr}, I_t^s, I_t, T_t^s, T_t, Thr, P_{ct})$**:**
- **Data:** Array of HCFs F_p for the modulations pool P.
- Testing signals per type T_t.
- Testing signals per type per SNR T_t^s.
- SNR threshold Thr.
- Total number of signals per type per SNR N_t^s.
- Training signals per type per SNR I_t^s.
- Training signals per type I_t.
- Number of SNR values N_{snr}.
- **Result:** Vector L_t of testing labels.
- Vector L_p of predicted labels.
- The splitting accuracy A_c.

for $i \leftarrow 1$ **to** *length of* P **do**
  for $j \leftarrow 1$ **to** *length of* $F_p(i)$ **do**
    if $j < Thr \times N_t^s$ **then**
      $L_{tr}((i-1) \times I_t^s + j) \leftarrow$ "L_1'';
      $L_t((i-1) \times T_t^s + j) \leftarrow$ "L_1'';
    else
      $L_{tr}((i-1) \times I_t^s + j) \leftarrow$ "L_2'';
      $L_t((i-1) \times T_t^s + j) \leftarrow$ "L_2'';
  Train SVM classifier using L_{tr}, then test it using L_t. Finally, store the accuracy in A_c and the predicted labels in L_p.

TABLE 3. CV values for all HCFs and SNR ranges.

Range	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	$\gamma_{\max}$	σ_{aa}	μ^a_{42}	m_A	E
S_1	0.522	1.859	0.704	0.268	0.202	0.349	0.590	0.189	0.756	0.227	0.190	0.029	0.084	0.082	0.021
S_2	0.523	1.727	0.713	0.266	0.202	0.350	0.588	0.188	0.758	0.228	0.190	0.029	0.084	0.082	0.021
S_3	0.521	1.712	0.712	0.267	0.204	0.353	0.588	0.188	0.756	0.229	0.190	0.029	0.085	0.082	0.021
S_4	0.521	1.972	0.707	0.267	0.203	0.352	0.586	0.187	0.753	0.229	0.190	0.029	0.085	0.082	0.021
S_5	0.522	2.271	0.702	0.265	0.202	0.351	0.588	0.857	0.753	0.228	0.189	0.029	0.084	0.083	0.021
S_6	0.524	2.122	0.705	0.264	0.202	0.352	0.587	0.193	0.760	0.228	0.191	0.029	0.084	0.083	0.021
S_7	0.575	2.102	0.748	0.264	0.202	0.352	0.585	0.218	0.851	0.229	0.191	0.029	0.084	0.085	0.023
S_8	0.749	2.078	0.887	0.263	0.200	0.350	0.592	0.309	1.11	0.235	0.218	0.029	0.083	0.097	0.032
S_9	0.891	2.167	0.826	0.282	0.198	0.367	0.605	0.804	1.365	0.323	0.694	0.032	0.081	0.15	0.059
S_{10}	0.935	2.089	0.729	0.373	0.242	0.335	0.689	0.694	1.515	0.545	1.462	0.052	0.081	0.317	0.118
S_{11}	0.933	1.453	0.669	0.481	0.372	0.345	0.877	0.788	1.609	0.668	1.831	0.101	0.085	0.673	0.216
S_{12}	0.930	1.887	0.673	0.530	0.530	0.494	1.235	0.732	1.704	0.635	1.935	0.179	0.121	1.148	0.337
S_{13}	0.931	1.944	0.687	0.562	0.664	0.645	1.761	0.701	1.812	0.661	1.97	0.278	0.175	1.496	0.449
S_{14}	0.934	1.808	0.694	0.604	0.768	0.764	2.279	0.701	1.914	0.728	1.983	0.375	0.215	1.659	0.533
S_{15}	0.935	1.973	0.702	0.651	0.853	0.860	2.667	0.717	1.995	0.792	1.988	0.448	0.239	1.727	0.589
S_{16}	0.936	2.263	0.708	0.693	0.927	0.942	2.925	0.734	2.051	0.838	1.991	0.495	0.254	1.754	0.625
S_{17}	0.936	2.092	0.710	0.728	1.002	1.022	3.122	0.749	2.084	0.865	1.993	0.523	0.264	1.767	0.648
S_{18}	0.936	1.967	0.710	0.753	1.075	1.10	3.261	0.759	2.10	0.879	1.996	0.539	0.291	1.772	0.661
S_{19}	0.937	2.018	0.713	0.766	1.15	1.178	3.353	0.764	2.108	0.884	1.999	0.548	0.500	1.773	0.669
S_{20}	0.937	3.683	0.714	0.775	1.224	1.255	3.629	0.766	2.111	0.887	1.998	0.552	1.192	1.775	0.672
S_{21}	0.937	3.701	0.711	0.783	1.283	1.317	3.834	0.768	2.113	0.889	1.999	0.554	1.984	1.774	0.675
S_{22}	0.937	2.142	0.711	0.789	1.327	1.362	3.688	0.769	2.113	0.889	2.001	0.555	2.414	1.774	0.676
S_{23}	0.937	2.126	0.711	0.794	1.356	1.392	3.49	0.770	2.113	0.889	2	0.555	2.583	1.774	0.676
S_{24}	0.937	2.109	0.711	0.797	1.371	1.407	3.632	0.770	2.113	0.889	2.001	0.555	2.641	1.774	0.676
S_{25}	0.937	3.524	0.711	0.795	1.376	1.412	3.593	0.770	2.113	0.889	2.001	0.555	2.67	1.774	0.676

calculated as follow:

$$\mathrm{CV_{km}} = \frac{\sigma_{km}}{\mu_{km}} \tag{30}$$

For simplicity, we denote the minimum and maximum values of SNR as r_1 and r_{L+1}, respectively. Hence, the overall range of SNR denoted as $[r_1, r_{L+1}]$ is equally split into L ranges as shown in Eq. (32). The group of signals for specific SNR range of $[r_{m-1}, r_m]$ are denoted as S^m as shown in Eq. (33). Hence, the signals are divided into a total of L groups as:

$$\mathrm{R} = \{[r_1, r_2], [r_2, r_3], \ldots, [r_{m-1}, r_m], \ldots, [r_L, r_{L+1}]\} \tag{31}$$

where $\mathrm{R_m}$ is calculated as:

$$\mathrm{r_m} = r_1 + \frac{(m-1)(r_{L+1} - r_1)}{L} \tag{32}$$

$$S^i = [r_{i-1}, r_i] : 1 < i < m+1 \tag{33}$$

There are many signals with SNR value equal to r_{m-1} are shared in any two consecutive groups of S^{m-1} and S^m. Under this circumstance, the values of each k-th feature f_k in these two groups are approximately same if both CV_{km-1} and CV_{km} values are approximately same. The second derivative of each CV_k vector is hence calculated to determine the SNR range where the CV of an HCF starts to vary. In principle, the positive and negative values of second derivative imply that a curve starts to concave up and concave down, respectively. The SNR range can be detected when a feature value starts to change drastically with increasing rate of variation along with the concave upward trends of curve. Table 3 shows the CV_{km} values of each k-th HCF for all groups of S^m, where the highlighted cells indicate the CV_k of particular feature is in a concave upward trend during S^m. Notably, two consecutive groups of CV_{km-1} and CV_{km} should be selected to obtain a center value of SNR as the required threshold. In order to fulfill the objectives of SNR splitting, it is also desirable to define a threshold value that is able to split the signal dataset into two subsets with similar sizes. The SNR ranges in which most of HCFs show the concave upward trend are then selected as threshold value for splitting.

Algorithm 2 presents the full pseudo code of HCFs and SNR threshold selection algorithm that considers the following inputs:

- A 3D array of HCFs denoted as F, where the first dimension represents the size of modulation pool (l_p), second dimension represents the total number of signals per type (N_t) and third dimension represents the HCFs count (l_f), respectively.
- Minimum value of SNR denoted as S_n^{min}.
- Maximum value of SNR denoted as S_n^{max}.
- Step size of SNR denoted as *step*.
- Percentages of signals used for testing denoted as P_{ct}.

The following variables are first initialized in Algorithm 2, i.e., number of SNR values (N_{snr}), number of testing signals per type per SNR (N_t^s), number of testing signals per type (N_t), number of training signals per type per SNR (I_t^s),

Algorithm 2 The Algorithm of HCFs and SNR Threshold Selection for SNR Splitter

Input: $F, N_t, S_n^{min}, S_n^{max}, step, P_{ct}$
Output: Selected HCFs group G_1 for SNR range splitter.
SNR Splitter accuracy A_c.
SNR splitter Threshold Thr.

$l_f \leftarrow$ *size of* 3^{rd} *dimension of* F;
$N_{snr} \leftarrow \frac{S_n^{max} - S_n^{min}}{2} + 1$;
$N_t^s \leftarrow N_t \,/\, N_{snr}$;
$T_t \leftarrow floor(\frac{P_{ct} * N_t}{100})$;
$T_t^s \leftarrow floor(\frac{P_{ct} * N_t^s}{100})$;
$I_t \leftarrow N_t - T_t$;
$I_t^s \leftarrow N_t^s - T_t^s$;
$Indices \leftarrow [1$ **to** $l_f]$;
for $k \leftarrow 1$ **to** l_f **do**
  for $m \leftarrow 1$ **to** $N_{snr} - 1$ **do**
    CV(m, k) $\leftarrow \frac{\sigma_F(m,k)}{\mu_F(m,k)}$
  $Z_k \leftarrow normalize(CV(:,k))$;
  Calculate $Z_k{}' \leftarrow \frac{\partial Z_k}{\partial x}$ and $Z_k{}'' \leftarrow \frac{\partial^2 Z_k}{\partial x^2}$
  for $j \leftarrow 1$**to** $N_{snr} - 2$ **do**
    if $Z_k{}''(j) > 0$ ***and*** $Z_k{}'(j) \times Z_k{}'(j+1) < 0$ **then**
      Remove the corresponding index of the feature f_k from *Indices*
    else if $Z_k{}''(j) < 0$ **then**
      $Z_k{}''(j) \leftarrow 0$;
    else if $Z_k{}''(j) > 0$ **then**
      $Z_k{}''(j) \leftarrow 1$;
  $Q_k \leftarrow Z_k{}''(1 : end-1) + Z_k{}''(2 : end)$;
  Remove all 1*s* from Q_k
  $A(:,k) \leftarrow Q_k$;
Store the sum of each row in A in a vector O
Apply K-means clustering on O.
Select the cluster K with maximum values.
$A_c \leftarrow 0$; **for** $t \leftarrow 1$**to** *length of* K **do**
  $T \leftarrow K(t) + 1$;
  $i \leftarrow 0$;
  for $n \leftarrow 1$**to** *length of Indices* **do**
    $idx \leftarrow Indices(n)$;
    if $Z_{idx}{}''(T)$ *is equal to* 1 *or* $Z_{idx}{}''(Thr - 1)$ *is equal to* 1 **then**
      $F_{indicies}(i++) \leftarrow idx$;
      $F_p(:,:,n) \leftarrow F(:,:,n)$;
  $[a, tl, pl] \leftarrow SNRSplitter(F_p, N_{snr}, N, Thr, P_{ct})$;
  if $a > SA$ **then**
    $Thr \leftarrow T$; $A_c \leftarrow a$; $L_t \leftarrow tl$; $L_p \leftarrow pl$;
    $G_1 \leftarrow F_{indicies}$;

the number of training signals per type (I_t) and the indices vector (*Indices*) that contains the indices of selected features. For each HCF, the *CVs* of features in all signal groups S^m are calculated using Eq. (30) and data normalization are subsequently performed on these CV values. The first and second derivatives of normalized CVs vector are then calculated. In order to prevent the errors in SNR splitting, the current feature needs to be excluded if it has relative minima or maxima when the CV vector is in a concave upward trend as indicated by the sign of first derivative. Each pair of consecutive signal groups (i.e., S^m and S^{m+1}) that indicate the changes of CV vector in an increasing rate of variation are then identified. Finally, the potential HCFs are fed into a SNR splitter function in order to determine the most relevant HCFs and best threshold (Thr) that can lead to the best splitting accuracy.

TABLE 4. SNR Splitter potential thresholds and the corresponding HCFs and splitting accuracies for the "Normal" classes.

Potential SNR threshold (dB)	Splitting Accuracy (%)	Selected HCFs
-4	94.94	$P_4, P_5, P_7, P_9, P_{10}, \gamma_{\max}, \sigma_{aa}, m_A, E$
-2	97.07	$P_4, P_5, P_7, P_{10}, \gamma_{\max}, \sigma_{aa}, m_A, E$
0	98.84	$P_4, P_5, P_6, P_7, \sigma_{aa}, m_A, E$
+2	99.49	$P_5, P_6, P_7, \sigma_{aa}, m_A, E$

TABLE 5. SNR splitter potential thresholds and the corresponding HCFs and splitting accuracies for the "Difficult" classes.

Potential SNR threshold (dB)	Splitting Accuracy (%)	Selected HCFs
-4	95.09	$P_1, P_4, P_5, P_7, P_9, P_{10}, \gamma_{\max}, \sigma_{aa}, m_A, E$
-2	96.65	$P_4, P_5, P_7, P_{10}, \gamma_{\max}, \sigma_{aa}, m_A, E$
0	98.16	$P_4, P_5, P_7, \sigma_{aa}, m_A, E$

3) EVALUATION OF PROPOSED SNR-SPLITTING ALGORITHM

The proposed SNR-splitting algorithm is evaluated on both "Normal" and "Difficult" classes of adopted dataset. Tables 4 and 5 show some potential splitting thresholds obtained by the proposed algorithm together with their corresponding selected HCFs and splitting accuracy for both "Normal" and "Difficult" classes, respectively. For "Normal" class, $Thr = +2\ dB$ is identified as the best SNR splitting threshold that can produce the selected HCFs of G_1 with the highest splitting accuracy of 99.49%.

$$G_1 = \{P_5, P_6, P_7, \sigma_{aa}, m_A, E\} \tag{34}$$

Fig. 3 shows the variation of mean envelope for each selected HCF in G_1 along with the overall SNR range for all modulation schemes considered in the "Normal" class. Each selected HCF of modulation schemes are observed to have different values around the splitting threshold of $Thr = +2\ dB$ that divide the total SNR range into two smaller ranges. When the whole SNR ranges are considered

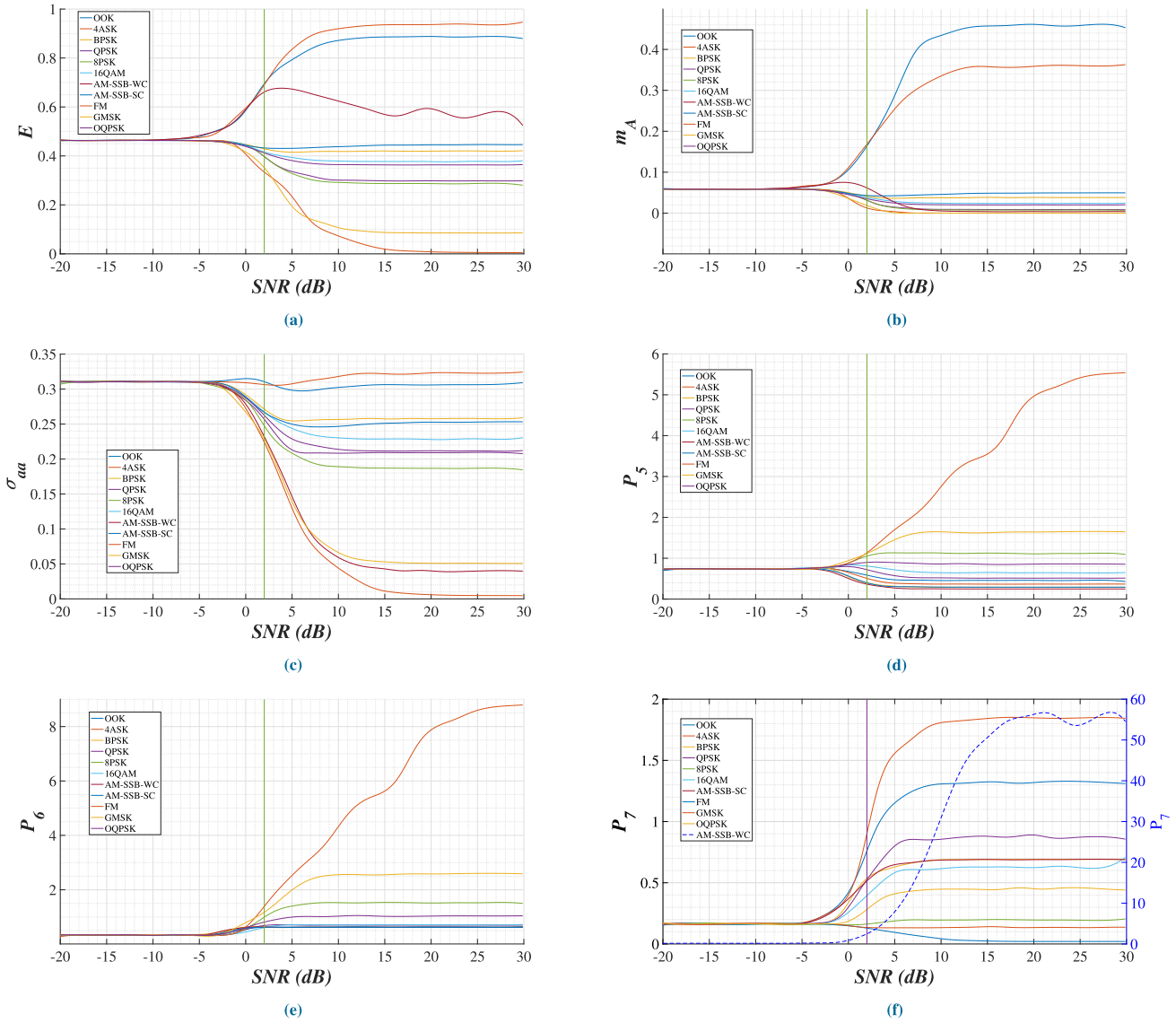

FIGURE 3. Values of (a) E, (b) m_A, (c) σ_{aa}, (d) P_5, (e) P_6, and (f) P_7 against SNR.

in training process, the hyperplanes of classifiers tend to be distorted and more complex CNN is needed to learn the most relevant CFs. On the other hand, the strategy of splitting the total SNR range into two smaller ranges can achieve higher accuracies by training the classifiers with more robust and relevant features extracted using the CNNs with lower complexity. The feasibility of this method will be further investigated in the following sections.

4) CROSS-SNR TRAINING METHOD

When a cross-SNR training method is used to train both CNNs on two different SNR ranges, at least one of the SNR value can be shared between these two models. The selection of shared SNR can be done by identifying the SNR ranges of splitter that produce majority of wrong labels. The distributions of wrong labels are subsequently presented in the histograms of Fig. 4 to identify the SNR values that produce most numbers of wrong labels. For our case, the CNN_1 and CNN_2 are trained on the SNR ranges of $R1 \cup [+2, +4]\,dB$ and $R2 \cup \{0\,dB\}$, respectively.

B. HCFs SELECTION FOR AMR

In this section, the characteristic of each HCF under different noise regimes is studied before designing a novel HCFs selection algorithm. The variance of each normalized HFC is first calculated for the overall SNR range. A careful study is then performed on the HCFs in the SNR ranges of R_1 and R_2 in order to identify the best combinations of HCFs that are able to boost the performance of AMR in the corresponding SNR ranges.

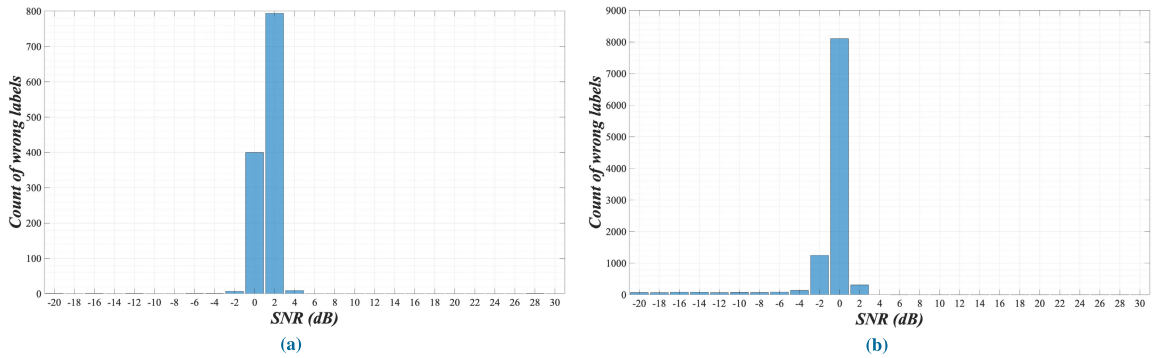

FIGURE 4. Histograms of the wrong labels resulted from SNR splitter for (a) "Normal" classes and (b) "Difficult" classes.

TABLE 6. The variance of normalized hand-crafted features.

Feature	Variance	Feature	Variance
$\gamma_{\max}$	0.4588	σ_{aa}	0.1771
μ^a_{42}	0.3631	m_A	0.5437
E	0.1430	P_1	0.5218
P_2	4.5219	P_3	0.3183
P_4	0.2865	P_5	0.1855
P_6	0.2567	P_7	0.6189
P_8	0.4413	P_9	0.7683
P_{10}	0.2829		

1) THE VARIANCE OF THE NORMALIZED HCFs

In order to construct a more robust and relevant feature subset, the behavior of each feature stored in original feature set F are investigated under different SNR values. Denote D_k as the variance of each k-th normalized HCF (F_k) for all modulation schemes under a specific SNR range, then:

$$D_k = \frac{1}{M \times N_t - 1} \sum_{m=1}^{M} \sum_{i=1}^{N_t} |F_k(i,m) - \mu_k(m)|^2 \quad (35)$$

where M is the size of modulation pool; N_t is the number of signals per modulation for the total SNR range; $F_k(i,m)$ is the normalized feature; $\mu_k(m)$ is the mean value of normalized feature F_k for m-th modulation and it is always equal to one. Both of $F_k(i,m)$ and $\mu_k(m)$ are calculated using Eqs. (36) and (37), respectively. Table 6 shows the variance of each normalized HCF stored in F.

$$F_k(i,m) = \frac{f_k(i,m)}{\frac{1}{N}\sum_{j=1}^{N} f_k(j,m)} \quad (36)$$

$$\mu_k(m) = \frac{1}{N} \sum_{i=1}^{N} F_k(i,m) = 1 \quad (37)$$

2) LIMITATIONS OF NORMALIZED VARIANCES FOR HCFs

Substantial amounts of literatures have revealed that it is not desirable to consider the features that can change with SNR during training process due to their high tendency of distorting the hyperplanes of classifiers and affecting their accuracies [30]. Wu *et al.* [30] considered the variance of normalized features to select robust features that can be used under varying noise conditions by identifying the cluster with minimum variance value. In contrary to [30], this paper does not only consider the minimum variance criterion in selecting the relevant features that can lead to best classification performance of modulation schemes. From practical point of view, some normalized features with relatively large variances can also be considered to boost the classification accuracy of proposed method because the value of an HCF can fall into different specific ranges when different modulation schemes are applied. In this context, it is also notable that the variance of normalized features [30] did not consider the mean value. Therefore, more investigations are required to construct the robust and relevant HCFs set.

3) CLASSIFICATION CONFUSION INDEX (CCI)

In practical, some features can have varying values under specific modulation scheme but they remain as constant values under other schemes. If the variance of normalized feature is the only criterion to be considered in selecting HCFs, some useful features that only demonstrate large variation in certain modulation schemes could be wrongly eliminated. In order to select the features that can lead to the best modulation recognition rate, the variance of all normalized means ($\mu'_k(m)$) is calculated as follows:

$$D_{\mu_k} = \frac{1}{M-1} \sum_{m=1}^{M} \left|\mu'_k(m) - E\left(\mu'_k(m)\right)\right|^2 \quad (38)$$

where $E\left(\mu'_k(m)\right)$ is the mean value of the normalized mean $\mu'_k(m)$ for the k-th feature F_k and under the m-th modulation scheme. Fundamentally, any features with larger values of D_{μ_k} imply for the larger distances between modulation schemes. Meanwhile, any feature with smaller variance D_k tends to have better robustness throughout the given SNR range. In other words, the better classification accuracy can be achieved through larger D_{μ_k} and smaller D_k. For any k-th feature, the associated classification confusion index of CCI_k can be calculated as:

$$CCI_k = \frac{D_k}{D_{\mu_k}} \quad (39)$$

TABLE 7. The variance of normalized HFCs for R_1.

Feature	Variance	Feature	Variance
$\gamma_{\max}$	0.2912	σ_{aa}	0.0015
μ_{42}^a	0.0079	m_A	0.01914
E	0.0025	P_1	0.8701
P_2	4.0764	P_3	0.6347
P_4	0.0838	P_5	0.0437
P_6	0.1528	P_7	0.4203
P_8	0.1170	P_9	1.0651
P_{10}	0.0923		

TABLE 8. The variance of normalized HFCs for R_2.

Feature	Variance	Feature	Variance
$\gamma_{\max}$	0.6005	σ_{aa}	0.2743
μ_{42}^a	0.2016	m_A	1.6820
E	0.2292	P_1	0.2742
P_2	1.9536	P_3	0.1123
P_4	0.0572	P_5	0.0467
P_6	0.0433	P_7	0.2830
P_8	0.2927	P_9	0.6493
P_{10}	0.1122		

After splitting the overall SNR range of $[r_1, r_{L+1}]$ into two smaller ranges of $[r_1, r_{Thr}]$ and $[r_{Thr+1}, r_{L+1}]$, the recalculation of variance for each normalized HCF under these two new SNR ranges can produce new clusters because the behavior of each feature can change under different SNR ranges. Table 7 and Table 8 show the variance of each normalized HCF produced under the SNR ranges of $R_1 = [-20, 0]\ dB$ and $R_2 = [2, 30]\ dB$, respectively. When K-means clustering is performed on all HCFs under the SNR range of R_1, the following four clusters are produced:

- $A1 = \{P_4, P_5, P_6, P_8, P_{10}, \sigma_{aa}, \mu_{42}^a, m_A, E\}$
- $B1 = \{P_7, \gamma_{\max}\}$
- $C1 = \{P_1, P_3, P_9\}$
- $D1 = \{P_2\}$

Meanwhile, the four clusters produced by K-means clustering in the SNR range of R_2 are:

- $A2 = \{P_3, P_4, P_5, P_6, P_{10}\}$
- $B2 = \{P_1, P_7, P_8, \sigma_{aa}, \mu_{42}^a, E\}$
- $C2 = \{P_9, \gamma_{\max}\}$
- $D2 = \{P_2, m_A\}$

The CCI_k of each k-th feature is calculated and the results of all features for SNR ranges of $R_1 = [-20, 0]dB$ and $R_2 = [2, 30]dB$ are presented in Table 10 and Table 9, respectively. The outliners that present in both SNR ranges are removed. For the SNR range of R_1, the HCFs of μ_{42}^a and, P_5 and P_6 are considered as the outliers and eliminated from the original feature set. K-means clustering is subsequently applied on the remaining features to produce the following four clusters:

- $A3 = \{P_1, P_3, P_8, P_9, \gamma_{\max}, m_A, E\}$
- $B3 = \{P_7, P_{10}\}$
- $C3 = \{P_4\}$
- $D3 = \{P_2, \sigma_{aa}\}$

TABLE 9. The CCIs of HCFs for R_1.

Feature	CCI	Feature	CCI
$\gamma_{\max}$	1.8813	σ_{aa}	42.3944
μ_{42}^a	1325.9667	m_A	4.6842
E	3.5785	P_1	2.6229
P_2	40.8835	P_3	4.3915
P_4	49.3454	P_5	91.6561
P_6	214.9963	P_7	21.6477
P_8	7.4465	P_9	1.1214
P_{10}	16.2419		

TABLE 10. The CCIs of HCFs for R_2.

Feature	CCI	Feature	CCI
$\gamma_{\max}$	0.1396	σ_{aa}	1.2346
μ_{42}^a	0.0650	m_A	0.5535
E	0.6293	P_1	0.2867
P_2	2.6910	P_3	0.2166
P_4	0.1320	P_5	0.0394
P_6	0.0339	P_7	0.0363
P_8	0.5489	P_9	0.1461
P_{10}	0.1685		

For the SNR range of R_2, P_2,and σ_{aa} are identified as the outliers and eliminated from the original feature sets. Four clusters are obtained from the remaining features by using K-means clustering, where:

- $A4 = \{P_5, P_6, P_7, \mu_{42}^a\}$
- $B4 = \{P_3, P_4, P_9, P_{10}, \gamma_{\max}\}$
- $C4 = \{P_1\}$
- $D4 = \{P_8, m_A, E\}$

The complete pseudocode of HCFs selection based on the CCI criterion is summarized using Algorithm 3. Evidently, different clusters of HCFs can be produced under the SNR ranges of R_1 and R_2 when the criteria of CCI and normalized variance are considered. The performance of these HCFs selection algorithms will be evaluated in the next sections.

VI. SIMULATION RESULTS

A. SIMULATION SETTINGS

RadioML 2018.01A dataset is selected for performance evaluation. The performances of proposed HCFs selection algorithm and proposed DL-based AMR method are compared with the methods used by O'Shea *et al.* [4]. The parameter settings required for simulation studies are presented in Table 11.

The method introduced by Wu *et al.* [30] to select a robust set of HCFs based on the normalized variance is considered for performance validation of the proposed HCFs selection algorithm. In addition, a Baseline (BL) AMR method presented in [4] is also selected for AMR performance validation of the proposed AMR based on the proposed HCFs selection algorithm and SNR splitter, where the former method considers a total of 28 HCFs including HOMs, HOCs and other features. Finally, both of the CNN/VGG and RN methods introduced in [4] are also considered for performance validation of the proposed DL-based AMR method.

Algorithm 3 Pseudo Code of HCFs Selection Based on the CCI Criterion

Function `GetClassificationFeatures` (F,Thr, N_{snr}) **:**
  Data: 3D array of HCFs $F(P, EF, FC)$ where P is the modulations pool, EF is the extracted features for all SNR values and FC is the features count.
  SNR Threshold (Thr) where the SNR total range is split into two ranges R_1 and R_2.
  SNR ranges (N_{snr}).
  Result: Features group F_1 for modulation classification on range R_1.
  Features group F_2 for modulation classification on range R_2.
  $Indices \leftarrow [1$ **to** $FC]$;
  for $k \leftarrow 1$ **to** FC **do**
    for $m \leftarrow 1$ **to** N_{snr} **do**
      if $m < Thr$ **then**
        $CCI_1(k) = \frac{D_k}{D_{\mu_k}}$;
      else
        $CCI_2(k) = \frac{D_k}{D_{\mu_k}}$;
  Remove outliers from CCI_1 and CCI_2.
  Apply K-means clustering on both CCI_1 and CCI_2.
  Select the cluster K_1 with minimum values in CCI_1.
  Select the cluster K_2 with minimum values in CCI_2.
  $F_1 \leftarrow$ corresponding indices of features in K_1;
  $F_2 \leftarrow$ corresponding indices of features in K_2;

TABLE 11. Experiment parameters and setting.

Parameters	Value
Input Signals	IQ samples at each SNR value for the "Normal" classes
Training signals per type per SNR I_t^s	3277
Training signals per type I_t	85, 202
Testing signals per type per SNR T_t^s	819
Testing signals per type T_t.	21, 294

B. SIMULATION RESULTS FOR HCFs SELECTION ALGORITHM

The effectiveness of the proposed HCFs selection algorithm in identifying the most relevant features is investigated in this section. First, we make a fair comparison between the performance of AMR based on the proposed HCFs selection algorithm and that based on the normalized variance used in [30]. The proposed CCI criterion and the normalized variance [30] are used separately to select the HCFs subset on the split SNR ranges R_1 and R_2 to be then fed into SVM_1 and SVM_2 classifiers, respectively. However, we propose to use the following methods for the comparison:

- HCFs-CCI-2cl: Adopt the proposed HCFs selection algorithm with two clusters of HCFs, i.e., $\{A_3, B_3\}$ and $\{A_4, B_4\}$ for R_1 and R_2, respectively.
- HCFs-VAR-2cl:Adopt the variance of normalized HCFs for HCFs selection [30] with two clusters of HCFs, i.e., $\{A_1, B_1\}$ and $\{A_2, B_2\}$ for R_1 and R_2, respectively.
- HCFs-CCI-1cl: Adopt the proposed HCFs selection algorithm with one cluster of HCFs, i.e., A_3 and A_4 for R_1 and R_2, respectively.
- HCFs-VAR-1cl:Adopt the variance of normalized HCFs for HCFs selection [30] with one cluster of HCFs, i.e., A_1 and A_2 for R_1 and R_2, respectively.

TABLE 12. Dimension of HCFs vector of the compared methods for R_1 and R_2.

Method	HCFs vector dimension on R_1	HCFs vector dimension on R_2
HCFs-CCI-2cl	9	9
HCFs-VAR-2cl	11	11
HCFs-CCI-1cl	7	4
HCFs-VAR-1cl	9	5
BL [4]	28	

On the other hand, the performances of the aforementioned methods are also compared with the BL method presented in [4]. Table 12 presents the dimension of HCFs vector produced by each compared HCFs selection algorithm for both SNR ranges of R_1 and R_2, whereas the classification performances of all compared methods are illustrated in Fig. 5. It is observed that the proposed HCFs selection algorithm with two clusters of HCFs (i.e., HCFs-CCI-2cl) has achieved the best classification accuracy among all compared methods. In addition, the proposed HCFs selection algorithm is also observed to outperform the compared HCFs selection algorithms based on normalized variance [29]. For instance, the classification accuracy of HCFs-CCI-2cl is more than 81% when the SNR is equal to 2 *dB* but HCFs-VAR-2cl can only produce the accuracy level around 60%. It is also notable that HCFs-CCI-2cl is able to produce the maximum classification accuracy rate of 97% at the high SNR regions, whereas the best classification accuracy rates obtained by both HCFs-VAR-2cl and BL [4] are only 96.7% and 94.6%, respectively. Similar observations are made when only one cluster of HCFs is considered, where the classification accuracy rate of HCFs-CCI-1cl is around 8% better than that of HCFs-VAR-1c1 throughout the SNR ranges. Referring to the simulation results presented in Table 12 and Fig. 5, it can be concluded that the proposed HCFs selection algorithm has more competitive performance for being able to select the most relevant HCFs and produce high classification accuracy with the lower dimension of HCFs vector.

C. SIMULATION RESULTS FOR THE PROPOSED DL-BAESD AMR TECHNIQUE

In this section, simulations are conducted for comparative studies between the proposed DL-based AMR method and

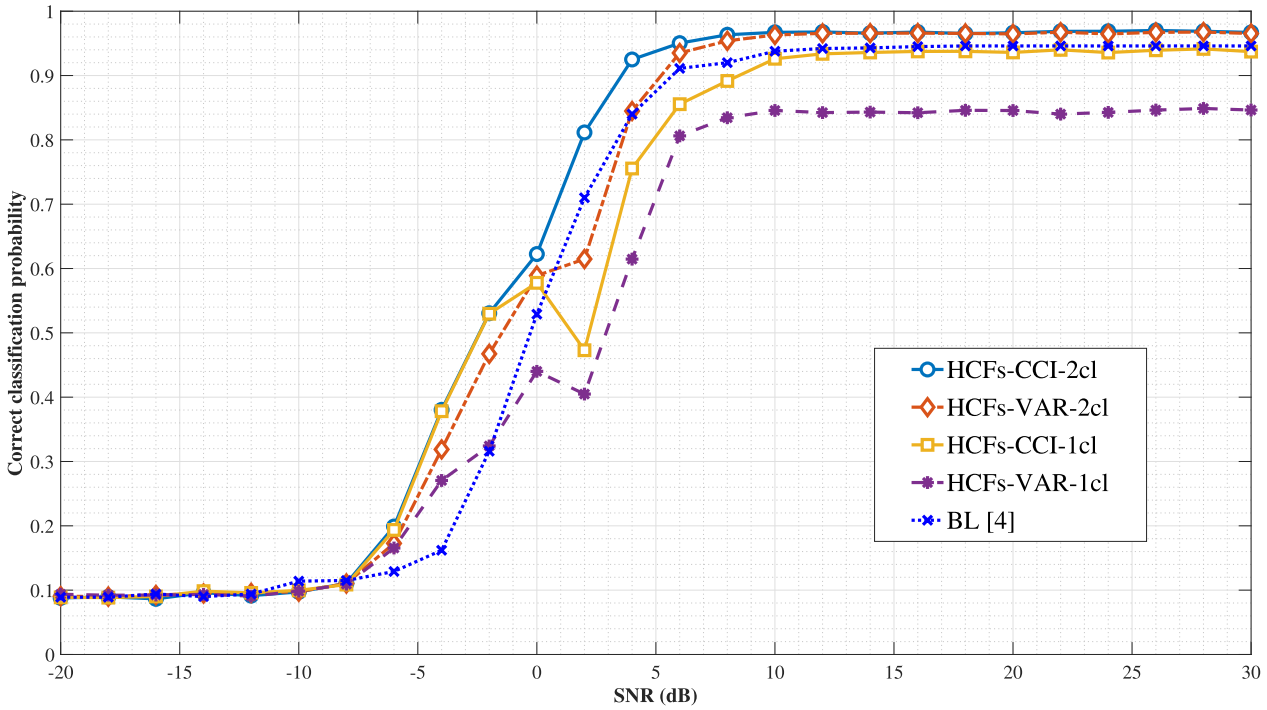

FIGURE 5. Performance comparison of the proposed HCFs selection algorithm, the normalized variance, and the baseline method in [4].

TABLE 13. Network trainable parameters.

Network	Trainable Parameters
CNN_1	145,856
CNN_2	27,328
CNN/VGG [4]	257,099
RN [4]	236,344

TABLE 14. Network structure for CNN_1 and CNN_2.

Layer	CNN_1 Output dimensions	CNN_2 Output dimensions
Input	2×1024	2×1024
conv	32×1024	16×1024
BN, RELU, Max Pool	32×512	16×512
conv	64×512	16×512
BN, RELU, Max Pool	64×256	16×256
conv	64×256	16×256
BN, RELU, Max Pool	64×128	16×128
FC/Softmax	11	11

both of the CNN/VGG and RN methods presented in [4]. The same parameter settings summarized in Table 11 are used in this performance analysis. Table 14 present the network structures of two CNN models denoted as CNN_1 and CNN_2 used to perform the contextual features extraction from signals belong to the SNR ranges of R_1 and R_2, respectively. Notably, both of CNN_1 and CNN_2 have fewer numbers of trainable parameters as compared to both CNN/VGG and RN employed in [4] as shown in Table 13.

CNN_1 is trained on 11 modulations schemes in the range of $[-20, +4]$ *dB*, while CNN_2 is trained on those in the range of $[0, +30]$ *dB*. Cross Entropy (CE) function is applied as loss

TABLE 15. Main hyper-parameters for CNNs training.

Network	Hyper-Parameters
mini batch size	1024
maximum epochs number	30
optimizer	SGDM
Loss function	cross entropy
Learning Rate	0.01

function, considering that AMR is essentially a multi-class classification task. The function of CE is shown as:

$$y_{CE} = -\sum_{C} y_c \log(\widehat{y}_c) \tag{40}$$

where y_c is the ground truth vector that can be achieved through one-hot encoding of sample label. $\widehat{y}_c$ is the predicted vector. C is the number of the samples' types. Table 15 shows some key hyper-parameters for the training process of both CNN_1 and CNN_2. The mini batch size of 1024 with 30 epochs are set for the training processes of both CNN models. In addition, a stochastic gradient descent with momentum (SGDM) optimizer is chosen to prevent the overfitting of networks. After constructing both CNN_1 and CNN_2 according to Table 14, their weights are initialized with learning rate equal to 0.01. Thereafter, CNN_1 and CNN_2 are trained on the corresponding training dataset and their weights are updated according to SGDM until validation loss is not improved in setting epochs.

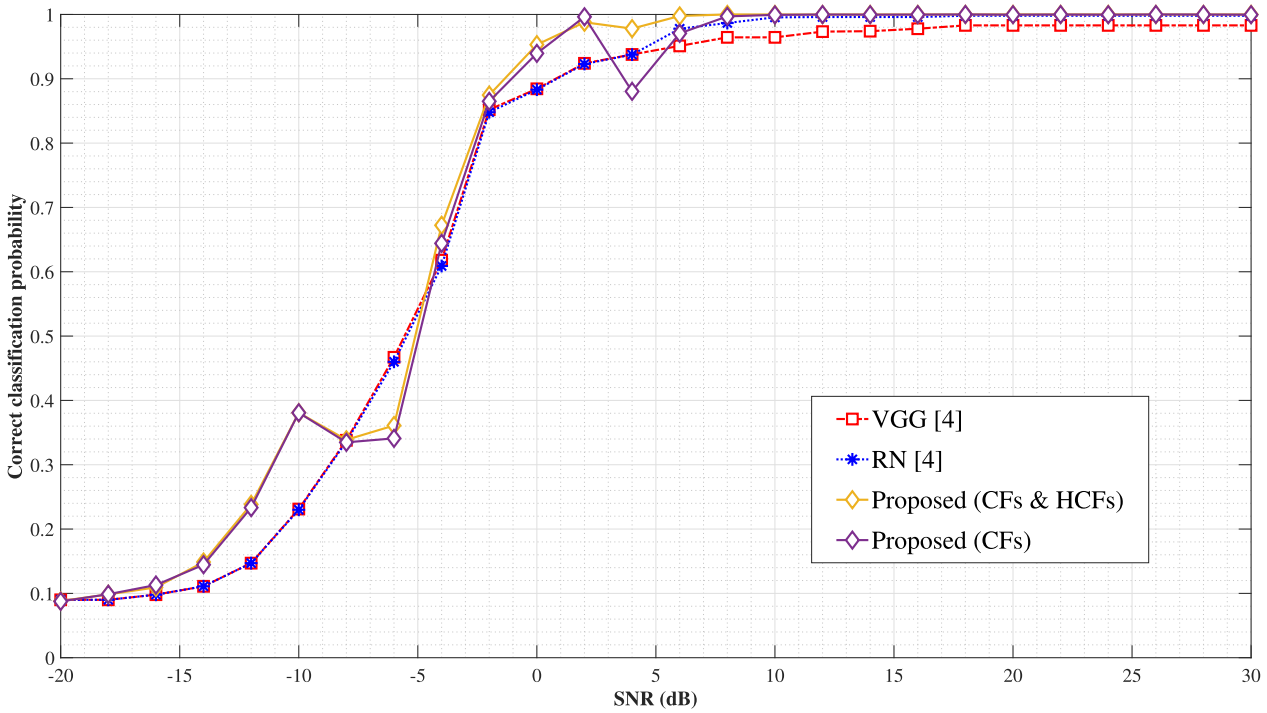

FIGURE 6. Performance comparison of the proposed AMR method.

The classification accuracies produced by the proposed DL-based AMR method and other benchmark methods are presented in Fig. 6. It is proven that the presence of HCFs can indeed contribute to the performance boosting because the networks that consider the concatenation of HCFs and CFs are able to achieve higher classification accuracy than those which only consider the CFs. The proposed method is also able to produce better accuracy than other benchmark methods regardless of the presence of HCFs. Particularly, the classification accuracy rate of proposed method is more than 87% and 95% when $SNR \geq -2\ dB$ and $SNR \geq 0\ dB$, respectively. It is also notable that the proposed method is able to achieve the maximum classification accuracy rate of 100% for $SNR \geq 6\ dB$.

The classification performances of proposed DL-based AMR method in handling each individual modulation scheme is further illustrated in Fig. 7. Specifically, the modulation schemes with lower information rates and unique structures such as AM and FM can be classified by the proposed method with high accuracy at the low SNR regions. For instance, both of the AM and FM schemes are identified by the proposed method with accuracy levels greater than 88% and 99.8% for $SNR = -4\ dB$ and $SNR \geq -2\ dB$, respectively. Meanwhile, the proposed method is able to identify the low order schemes such as GMSK, 4ASK and BPSK with classification accuracy rate around 99% when $SNR = -2\ dB$. The modulation schemes with more sophisticated structures and higher orders require higher SNR for robust performance. From Fig. 7, it is notable that the proposed method is able to classify all modulation schemes with the accuracy rate of 100% when $SNR \geq 8\ dB$.

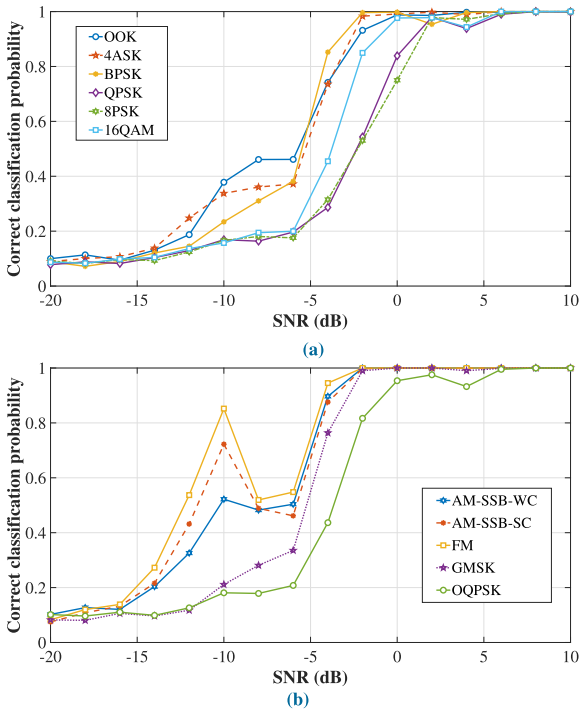

FIGURE 7. Classification performance for individual modulation schemes (a) AM, FM, GMSK, and OQPSK schemes; and (b) OOK, 4ASK, BPSK, QPSK, 8PSK, and 16QAM modulation schemes.

Six confusion matrices produced by the proposed DL-based AMR method across all 11 classes for multiple SNR values are presented in Fig. 8 in order to further analyze the classification performance of proposed method. On one

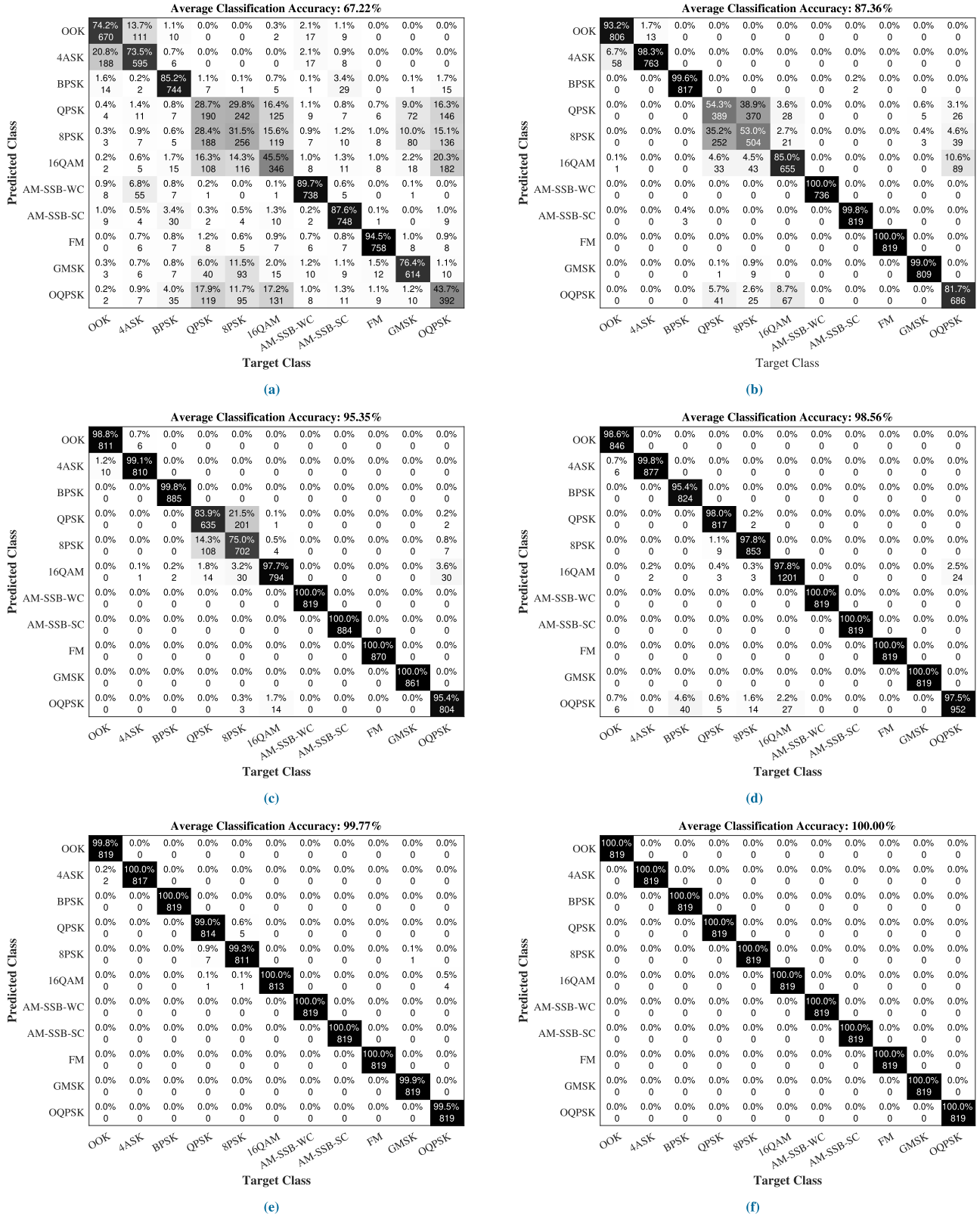

FIGURE 8. 11-modulation confusion matrix of the proposed DL-based AMR method for (a) SNR = $-4dB$, (b) SNR = $-2dB$, (c) SNR = $0dB$, (d) SNR = $2dB$, (e) SNR = $6dB$, and (f) SNR = $8dB$.

hand, it can be observed that the majority of classification errors can be found between phase-shift keying (QPSK, 8PSK, OQPSK), especially with substantial noise. This may be due to lack of information at low SNR values [31]. Besides, they have similar symbol structure and statistical information when using features computed by using the IQ samples. On the other hand, some signals with lower information rates and unique structure such as AM and FM are more readily classified at low SNR.

VII. CONCLUSION

In this paper, a novel DL-based AMR method that is able to perform robustly under varying noise regimes is proposed by addressing the key issues of feature extraction and selection criteria of features. Three groups of HCFs are first extracted from the received signals by using the proposed algorithms. The first group of HCFs is used to identify the SNR range of received signals, whereas the remaining two groups are concatenated with the CFs extracted by two corresponding CNNs in order to solve the modulation classification tasks. A novel algorithm is also introduced to select the best SNR threshold and HCFs that can split the total SNR range into two smaller ranges. Furthermore, an HCFs selection algorithm is proposed to select the most relevant features in order to reduce the dimension of feature vector.

Extensive simulation studies have verified the reliability and effectiveness of proposed HCFs selection algorithm in reducing the redundant features without compromising the classification performances. These desirable characteristics enables the proposed method to be implemented through simpler CNNs and verified using 11 modulation schemes. The competitive classification performance demonstrated by proposed method have proven that it is more efficient to split the total SNR range with positive and negative values into two smaller ranges. When training with the total SNR range, the hyperplanes of classifiers tend to be distorted and hence compromising their classification accuracies. Furthermore, the presence of HCFs is also proven useful to boost the classification performance of the proposed DL-based AMR method. The proposed DL-based AMR method is expected to make significant contribution in wireless communication because it is able to solve the AMR tasks with higher classification accuracy while incurring lower computation complexity due to the adoption of simpler CNN architecture. Current research work can be extended by incorporating more optimized CNN architectures into the proposed DL-based AMR method to solve larger sets of modulation schemes.

REFERENCES

[1] 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, doi: 10.1109/ACCESS.2021.3071801.

[2] T. O'Shea and N. West, "Radio machine learning dataset generation with GNU radio," in *Proc. GNU Radio Conf.*, vol. 1, no. 1, 2016. [Online]. Available: https://pubs.gnuradio.org/index.php/grcon/article/view/11

[3] T. J O'Shea, J. Corgan, and T. Charles Clancy, "Convolutional radio modulation recognition networks," 2016, *arXiv:1602.04105*. [Online]. Available: http://arxiv.org/abs/1602.04105

[4] 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, doi: 10.1109/JSTSP.2018.2797022.

[5] N. E. West and T. J. O'Shea, "Deep architectures for modulation recognition," 2017, *arXiv:1703.09197*. [Online]. Available: https://arxiv.org/abs/1703.09197

[6] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017, doi: 10.1109/TCCN.2017.2758370.

[7] Q. Zhang, Z. Xu, and P. Zhang, "Modulation scheme recognition using convolutional neural network," *J. Eng.*, vol. 2019, no. 23, pp. 9075–9078, Dec. 2019, doi: 10.1049/joe.2018.9188.

[8] Y. Sun, J. Li, F. Lin, and G. Pan, "Automatic signal modulation recognition based on deep convolutional neural network," in *Proc. 3rd Int. Conf. Comput. Eng., Inf. Sci. Appl. Technol. (ICCIA)*, 2019, pp. 550–554, doi: 10.2991/iccia-19.2019.86.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[10] Z. Zhang, C. Wang, C. Gan, S. Sun, and M. Wang, "Automatic modulation classification using convolutional neural network with features fusion of SPWVD and BJD," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 3, pp. 469–478, Sep. 2019, doi: 10.1109/TSIPN.2019.2900201.

[11] S. Peng, H. Jiang, H. Wang, H. Alwageed, and Y.-D. Yao, "Modulation classification using convolutional neural network based deep learning model," in *Proc. 26th Wireless Opt. Commun. Conf. (WOCC)*, Apr. 2017, pp. 1–5, doi: 10.1109/WOCC.2017.7929000.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[13] K. Jiang, J. Zhang, H. Wu, A. Wang, and Y. Iwahori, "A novel digital modulation recognition algorithm based on deep convolutional neural network," *Appl. Sci.*, vol. 10, no. 3, p. 1166, Feb. 2020, doi: 10.3390/app10031166.

[14] B. Tang, Y. Tu, Z. Zhang, and Y. Lin, "Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks," *IEEE Access*, vol. 6, pp. 15713–15722, 2018, doi: 10.1109/ACCESS.2018.2815741.

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014, *arXiv:1710.07035*. [Online]. Available: https://arxiv.org/abs/1710.07035

[16] K. Hassan, I. Dayoub, W. Hamouda, C. N. Nzeza, and M. Berbineau, "Blind digital modulation identification for spatially-correlated MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 683–693, Feb. 2012, doi: 10.1109/TWC.2011.122211.110236.

[17] 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, doi: 10.1109/ACCESS.2019.2928463.

[18] I. Vernikos, E. Mathe, E. Spyrou, A. Mitsou, T. Giannakopoulos, and P. Mylonas, "Fusing handcrafted and contextual features for human activity recognition," in *Proc. 14th Int. Workshop Semantic Social Media Adaptation Personalization (SMAP)*, Jun. 2019, pp. 1–6, doi: 10.1109/SMAP.2019.8864848.

[19] D. T. Nguyen, T. D. Pham, N. R. Baek, and K. R. Park, "Combining deep and handcrafted image features for presentation attack detection in face recognition systems using visible-light camera sensors," *Sensors*, vol. 18, no. 3, p. 699, 2018, doi: 10.3390/s18030699.

[20] H. Wang, A. Cruz-Roa, A. Basavanhally, H. Gilmore, N. Shih, M. Feldman, J. Tomaszewski, F. González, and A. Madabhushi, "Cascaded ensemble of convolutional neural networks and handcrafted features for mitosis detection," *Proc. SPIE*, vol. 9041, Feb. 2014, Art. no. 90410B, doi: 10.1117/12.2043902.

[21] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 7, pp. 710–732, Jul. 1992, doi: 10.1109/34.142909.

[22] T. V. R. O. Camara, A. D. L. Lima, B. M. M. Lima, A. I. R. Fontes, A. D. M. Martins, and L. F. Q. Silveira, "Automatic modulation classification architectures based on cyclostationary features in impulsive environments," *IEEE Access*, vol. 7, pp. 138512–138527, 2019, doi: 10.1109/ACCESS.2019.2943300.

[23] J. Zhang, Y. Li, and J. Yin, "A novel modulation classification method for fm signals based on the time-frequency distribution and CNN," *IET Radar, Sonar Navigat.*, vol. 12, no. 2, pp. 244–249, Oct. 2017, doi: 10.1049/iet-rsn.2017.0265.

[24] N. Daldal, Z. Cömert, and K. Polat, "Automatic determination of digital modulation types with different noises using convolutional neural network based on time–frequency information," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105834, doi: 10.1016/j.asoc.2019.105834.
[25] D. Figueiredo, A. Furtado, and R. Oliveiray, "Modulation classification using joint time and frequency-domain data," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5, doi: 10.1109/VTC2020-Spring48590.2020.9128493.
[26] E. E. Azzouz and A. K. Nandi, "Automatic identification of digital modulation types," *Signal Process.*, vol. 47, no. 1, pp. 55–69, Nov. 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0165168495000992, doi: 10.1016/0165-1684(95)00099-2.
[27] 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, doi: 10.1109/ACCESS.2019.2916833.
[28] Z. Zhang, Z. Hua, and Y. Liu, "Modulation classification in multipath fading channels using sixth-order cumulants and stacked convolutional auto-encoders," *IET Commun.*, vol. 11, no. 6, pp. 910–915, Apr. 2017, doi: 10.1049/iet-com.2016.0533.
[29] C. E. Brown, *Coefficient Variation*. Berlin, Germany: Springer, 1998, pp. 155–157, doi: 10.1007/978-3-642-80328-4_13.
[30] Z. Wu, S. Zhou, Z. Yin, B. Ma, and Z. Yang, "Robust automatic modulation classification under varying noise conditions," *IEEE Access*, vol. 5, pp. 19733–19741, Aug. 2017, doi: 10.1109/ACCESS.2017.2746140.
[31] S. Kharbech, I. Dayoub, M. Zwingelstein-Colin, and E. P. Simon, "Blind digital modulation identification for MIMO systems in railway environments with high-speed channels and impulsive noise," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7370–7379, Aug. 2018, doi: 10.1109/TVT.2018.2834869.

BACHIR JDID received the B.Eng.Sc. degree in communications engineering from the Higher Institute for Applied Sciences and Technology, Damascus, Syria, in 2013. He is currently pursuing the M.Phil. degree with the Faculty of Engineering, Technology and Built Environment, UCSI University, Kuala Lumpur, Malaysia. His research interests include wireless communications, signal processing, image processing, embedded systems, artificial intelligence, and its application in wireless communications.

WEI HONG LIM (Senior Member, IEEE) received the B.Eng. degree (Hons.) in mechatronic engineering and the Ph.D. degree in computational intelligence from Universiti Sains Malaysia, Penang, Malaysia, in 2011 and 2014, respectively. From 2015 to 2017, he was with the Intelligent Control Laboratory, National Taipei University of Technology, Taiwan, as a Postdoctoral Researcher, where he was a Visiting Researcher, in 2019. He is currently an Assistant Professor with the Faculty of Engineering, Technology and Built Environment, UCSI University. He is working with three national research grants awarded by the Ministry of Education, Malaysia, and five internal grant projects supported by UCSI University. He has published more than 50 research articles in research areas related to computational intelligence, optimization algorithms, energy management, and digital image processing. To date, he has been actively involved in various professional bodies and registered as the European Engineer (EUR ING) with the Fédération Européenne d'Associations Nationales d'Ingénieurs (FEANI), a Chartered Engineer (C.Eng.) with the Engineering Council UK (ECUK), and a Professional Technologist (P.Tech.) with the Malaysia Board of Technologist (MBOT). He is also an Active Reviewer for various reputable journals, such as IEEE Access, *Complexity*, *Mathematical Problems in Engineering*, *Computational Intelligence*, and *Neuroscience*.

IYAD DAYOUB (Senior Member, IEEE) received the B.Eng. degree in telecommunications and electronics, Syria, in 1993, the M.A.Sc. degree in electrical engineering from the National Polytechnic Institute of Lorraine (INPL), and the Ph.D. degree from the University of Valenciennes/Institute of Electronics, Microelectronics and Nanotechnology (IEMN), in 2001. He has worked as a System Engineer with Siemens, Middle East. He has worked as a Researcher with Alcatel Business Systems Alcatel, Colombes, Paris. From 2007 to 2014, he was a member in the area of electrical engineering, electronics, photonics, and systems of the National Council of Universities (CNU), France. From 2010 to 2014, he was an Adjunct Professor with Concordia University, Montreal. He is currently Professor of communications engineering. His current research activities at the IEMN of Université Polytechnique Hauts de France (UPHF) and INSA des Hauts de France are focused on wireless communications, high-speed communications, cognitive radio, and hybrid radio-optic technologies. He is a member of several International Conference Advisory Committees, Technical Program Committees, and Organization Committees, such as VTC, GLOBECOM, ICC, PIMRC, and WWC.

KAIS HASSAN was born in 1975. He graduated in telecommunications engineering from the Higher Institute for Applied Science and Technology (HIAST), Syria, in 2000. He received the M.S. degree in telecommunications systems and the Ph.D. degree in telecommunications from the University of Valenciennes, France, in 2009 and 2012, respectively. From 2000 to 2008, he was a Research Engineer at HIAST. From 2013 to 2014, he was an Assistant Professor with the University of Western Brittany, France. Since September 2014, he has been an Associate Professor with Le Mans University (LAUM), France. He is with the Acoustics Laboratory, LAUM. His research interests include wireless communications, signal processing, and cognitive radio.

MOHD RIZON BIN MOHAMED JUHARI received the bachelor's degree in engineering and the master's degree in electrical and electronics engineering from the University of Tokushima, Japan, in 1993 and 1995, respectively, and the Ph.D. degree in engineering from the Department of Computer Science and Intelligent System, Oita University, Japan, in 2002. From 1995 to 1997, he was a Software Engineer with System LSI Laboratory, Mitsubishi Electric Corporation, Itami, Japan. In 1997, he moved to the University of Malaya, Malaysia. He was the Head of the Biomedical Engineering Department, University Malaya. He was a Professor of biomedical technology with King Saud University, Saudi Arabia, in 2010. He served at MIEC, Fuzhou University, China, in 2019. He is currently a Professor with the Department of Electrical and Electronics, UCSI University, Malaysia.