Received: 20 June 2023 | Revised: 2 August 2023 | Accepted: 7 August 2023 | Published online: 8 August 2023

Journal of Data Science and Intelligent Systems
2025, Vol. 3(1) 18–26
DOI: 10.47852/bonviewJDSIS32021244
BON VIEW PUBLISHING

RESEARCH ARTICLE

Automatic Modulation Recognition Based on a New Deep K-SVD Denoising Algorithm

Yanhe Li[1], Xingxing He[1,*] and Chun Zhou[1]

[1]School of Mathematics, Southwest Jiaotong University, China

Abstract: Automatic modulation recognition (AMR) has a wide range of applications in wireless communication. To solve the problem that the previous methods convert signal modulation recognition into image recognition, leading to increased time costs and information loss, an AMR approach consisting of the improved deep singular value decomposition (K-SVD) denoising algorithm is suggested. First, the effectiveness of the model for random sine wave denoising is demonstrated by simulation. Second, the original I/Q signals are fed into the modified deep K-SVD model for denoising, skipping the complicated image processing steps. Finally, the noise-reduced signals are input into a multi-channel convolutional long short-term neural network to complete the modulation recognition. To solve the slow convergence problem of iterative shrinkage thresholding algorithms in sparse coding, the fast iterative shrinkage thresholding algorithm is adopted to improve the computational efficiency and obtain a better denoising effect. The experiments show that the improved model has an average recognition accuracy of 91.26% at different signal-to-noise ratio (SNRs) from −2 dB to 18 dB, which is better than the state-of-the-art modulation recognition models.

Keywords: deep learning, automatic modulation recognition, sparse coding, deep K-SVD

1. Introduction

Automatic modulation recognition (AMR) is a technique for determining the kind of modulation of the received signal at the receiver end of a communication system in the presence of unknown signal parameters and noisy interference with the modulation information. AMR has recently become a key step between signal detection and signal demodulation due to the advancement of communication technology, the increasing diversity and complexity of electromagnetic spaces, and signal modulation types. It is also one of the main research topics in the area of wireless communication [1, 2].

AMR can be generally grouped into two categories: likelihood-based (LB) methods and feature-based (FB) methods [3–5]. In the LB method, AMR first formulates several hypotheses for various modulation types and calculates the likelihood function of the received signal with various assumptions. Next, it compares the value of the likelihood function with a pre-set threshold to obtain the potential classes of the signals. The FB approach recognizes patterns by extracting artificial features and a trained classification model. However, the former limits its application to practical problems due to its high computational complexity and reliance on a priori knowledge, and the latter suffers from the difficulty and relatively time-consuming nature of some feature extraction.

Recently, deep learning (DL) has attracted much attention as a result of its notable success in different applications, including speech recognition, computer vision, and natural language processing. These fields once relied heavily on feature extraction, and DL has significantly improved the capacity to do so. This trend also widely applied in radio signal and communication in recent years. For instance, in previous studies like Li et al. [6], Peng et al. [7], and Zhang et al. [8], the raw radio signal was first converted into images, and then used as inputs to the deep neural network by some data preprocess. However, these image-based approaches also have some weaknesses. For example, it takes time to convert all input signals into time-frequency distribution images, it may result in the loss of amplitude information compared to the raw data, and it does not fully utilize the complementarity and synergy between the various signal channels.

In this article, a signal modulation recognition method based on deep singular value decomposition (K-SVD) is proposed to address the above problem. Firstly, the deep SVD (K-SVD) model is used to complete the noise reduction of the original signal, skipping the complex image processing steps. The noise-reduced signal is then fed into multi-channel convolutional long short-term deep neural network (MCLDNN), which integrates the different features of the noise-reduced signal to a large extent. The classification results are finally obtained. In addition, an accelerated version of the iterative shrinkage thresholding algorithm (ISTA) called the fast iterative shrinkage thresholding algorithm (FISTA) is used to improve computational efficiency while obtaining a better sparse representation to address the issue of slow convergence of the ISTA in solving the sparse coding process.

This article is structured as follows: Section 2 recalls the related work of DL applications to AMR. Section 3 describes the deep K-SVD algorithm and the MCLDNN. The architecture of the

*Corresponding author: Xingxing He, School of Mathematics, Southwest Jiaotong University, China. Email: x.he@swjtu.edu.cn.

proposed model and loss function are discussed in Section 4. Dataset and experimental results are discussed in Section 5. Finally, Section 6 concludes the article.

2. Literature Review

DL applications for AMR can be grouped into two categories, depending on the type of input data used [1, 9]: one is based on the one-dimensional features of the signal (I/Q sequences, phase spectrum, etc.), which is mainly modulated by extracting features directly from one-dimensional signals using convolutional neural network (CNN) and other recognition [10–12]; the second is based on two-dimensional feature maps of signals (constellation maps, time-frequency maps, etc.), which focus on converting signal recognition into the studied image recognition problem [6–8].

For the 1D signal FB approach, Rajendran et al. [10] attempted to train a long short-term memory network (LSTM) by applying the time-domain amplitude and phase information of the received data. West and O'Shea [12] demonstrated that LSTM excels at modeling time-series data, while CNN is skilled at extracting spatial features. In AMR, combining the two can produce outstanding results, but it neglects the correlation between different features. Xu et al. [11] developed another multi-channel learning framework based on CNN-LSTM networks, which exploits the complementarity and synergy of signals for feature extraction and obtains better recognition accuracy.

For the signal-based 2D image conversion, Zhang et al. [8] suggested an upgraded CNN-AMC network by combining images and handcrafted features to extract more complex characteristics, which can enhance performance in comparison to prior research. Li et al. [6] built a CNN-based radio signal recognition model to obtain the spectrum image features of the received data and introduced a sparse filtering criterion to enhance the recognition efficiency. When representing complicated signals as images with grid-like topology, Peng et al. [7] looked into a number of representation techniques and used two CNN networks (AlexNet and GoogLeNet) to extract features from the generated images for AMR.

In this article, to solve the problem that the image-based conversion methods convert signal modulation recognition into image recognition, leading to increased time costs and information loss, an end-to-end AMR model consisting of the improved deep K-SVD denoising algorithm is suggested.

3. Basic Theory

3.1. K-SVD algorithm

K-SVD is an overcomplete dictionary design algorithm based on sparse representation: given a set of training data, find the best representation dictionary for each sample under the sparsity requirement [13, 14]. There are two basic steps: sparse coding and dictionary updating [14].

3.1.1. Sparse coding

The modulation signal x is represented as a small block of data of size $\sqrt{p} \times \sqrt{p}$, arranged in dictionary order into a vector of columns with the length p. The sparse representation model is based on the assumption that x is a linear combination of $s \ll p$ columns drawn from a chosen dictionary $\boldsymbol{D} \in R^{p\times m}$, i.e., $x = \boldsymbol{D\alpha}$, where $\alpha \in R^m$ is a sparse vector containing s non-zero elements. The noisy result y is obtained by adding to x a zero-mean Gaussian White Noise with a standard deviation of σ. The sparse code $\hat{\alpha}$ for y can be yielded by resolving the optimized function:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\alpha} \lambda||\boldsymbol{\alpha}||_1 + \frac{1}{2}||\boldsymbol{D\alpha} - y||_2^2 \tag{1}$$

where λ is the regularization factor.

Consider the data $\boldsymbol{X}$ of size $\sqrt{N} \times \sqrt{N}$ and its noisy data $\boldsymbol{Y}$ with the objective equation:

$$\min_{\alpha_k, X} \frac{\boldsymbol{\mu}}{2}||\boldsymbol{X} - \boldsymbol{Y}||_2^2 + \sum_k (\lambda_k||\boldsymbol{\alpha}_k||_0 + \frac{1}{2}||\boldsymbol{D\alpha}_k - F_k\boldsymbol{X}||_2^2) \tag{2}$$

where μ and λ are the regularization parameters, $x_k = F_k\boldsymbol{X}$, and $F_k \in R^{p\times N}$ denotes the process that splits $\boldsymbol{X}$ into small patches $\boldsymbol{x}_k$.

Assuming that the dictionary $\boldsymbol{D}$ has been defined, Equation (2) involves two types of unknowns: the output $\boldsymbol{X}$ and the sparse representation $\boldsymbol{\alpha}_k$ for each position.

Elad and Aharon [14] proposed the block coordinate descent that begins by initializing $\boldsymbol{X} = \boldsymbol{Y}$ and then searches the optimal $\hat{\alpha}_k$ for every position k, rather than solving both problems simultaneously. As a result, decoupling the task into numerous smaller pursuit forms becomes:

$$\hat{\boldsymbol{\alpha}}_k = \arg\min_{\alpha_k} \lambda_k||\boldsymbol{\alpha}_k||_0 + \frac{1}{2}||\boldsymbol{D\alpha}_k - \boldsymbol{x}_k||_2^2 \tag{3}$$

3.1.2. Dictionary updating

Assume that the sparse representation of all patches is $\{\hat{\boldsymbol{\alpha}}_k\}_k$. Then fix those and turn to update $\boldsymbol{X}$ by substituting $\{\hat{\boldsymbol{\alpha}}_k\}_k$ for all instances of α_k in Equation (2) and ignoring the constant term to yield:

$$\hat{\boldsymbol{X}} = \arg\min_{X} \frac{\mu}{2}||\boldsymbol{X} - \boldsymbol{Y}||_2^2 + \frac{1}{2}\sum_k ||\boldsymbol{D}\hat{\alpha}_k - F_k\boldsymbol{X}||_2^2 \tag{4}$$

This is a quadratic term that has a closed-form solution:

$$\hat{\boldsymbol{X}} = (\sum_k F_k^T F_k + \mu\boldsymbol{I})^{-1}(\mu\boldsymbol{Y} + \sum_k F_k^T\boldsymbol{D}\hat{\alpha}_k) \tag{5}$$

This yields the result after the first round of denoising. In fact, these expressions return the blocks to their initial places and average them using a weighted form of the noisy data.

In the case where the dictionary $\boldsymbol{D}$ is unknown, the ultimate function becomes:

$$\hat{\boldsymbol{X}} = \min_{\alpha_k, \boldsymbol{X}, \boldsymbol{D}} \frac{\mu}{2}||\boldsymbol{X} - \boldsymbol{Y}||_2^2 + \sum_k (\lambda_k||\boldsymbol{\alpha}_k||_0 + \frac{1}{2}||\boldsymbol{D\alpha}_k - F_k\boldsymbol{X}||_2^2) \tag{6}$$

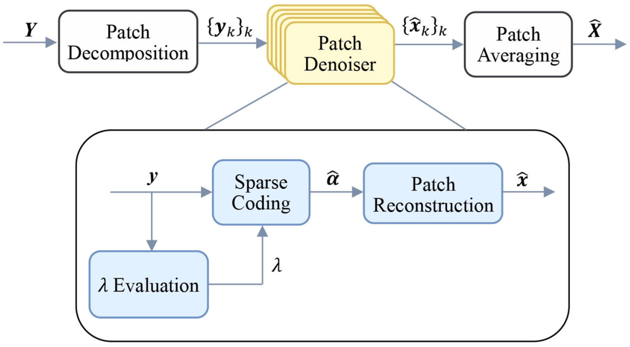
Block Coordinate Descent is utilized in this situation to determine the dictionary $\boldsymbol{D}$: Let $\boldsymbol{X} = \boldsymbol{Y}$ and initialize $\boldsymbol{D}$ to be the overcomplete discrete cosine transform (DCT) matrix. Then, cycle among an update of $\boldsymbol{D}$ using the K-SVD and the orthogonal matching pursuit (OMP) [15] for all the blocks. Both procedures of sparse coding and dictionary updating are cyclical until a predetermined error criterion or a number of iterations is reached.

3.2. Deep K-SVD denoising algorithm

The deep K-SVD algorithm combines the ideas of the K-SVD denoising algorithm and DL by unfolding the K-SVD into a deep network, which retains the essence of the original K-SVD while having only a small number of learnable parameters [14]. In the pursuit stage, since the OMP algorithm is not differentiable, an equivalent learnable algorithm such as ISTA [16, 17] is considered to replace the OMP. However, the OMP can easily be adapted to process each block using a noise level-based stopping criterion; the equivalence of ISTA requires the selection of the appropriate regularization parameter λ_k for every block, and therefore a neural network with a better fitting capability is considered for this task.

The complete process involves (i) splitting the data into tiny overlapped blocks, (ii) getting the sparse representation in a gained dictionary, and (iii) averaging the blocks to recover the clean data by adhering to the K-SVD calculation method. The following three parts constitute the end-to-end architecture, which is depicted in Figure 1.

Figure 1
The architecture of deep K-SVD



3.2.1. Patch decomposition

For a noisy signal $\boldsymbol{Y} \in R^{\sqrt{N}\times\sqrt{N}}$ with a standard deviation σ and additive zero-mean white Gaussian noise, decompose it into many overlapping patches $\{\boldsymbol{y}_k\}_k$, where $\boldsymbol{y}_k \in R^{\sqrt{p}\times\sqrt{p}}$.

3.2.2. Patch denoising

According to the computational steps of the deep K-SVD denoising algorithm in Elad and Aharon [14], for each patch y, a three-layer perceptron network is utilized to learn its regularization parameter λ.Then the sparse representation $\hat{\alpha}$ of each patch is given by ISTA, and finally the denoising result $\hat{x}$ is obtained by multiplying $\boldsymbol{D}$ with $\hat{\alpha}$, where $\boldsymbol{D}$ is a randomly initialized DCT matrix, and the step size c of the soft threshold function is the squared spectral parametrization of $\boldsymbol{D}$. The dictionary $\boldsymbol{D}$, the step size c, and the regularization parameter λ are the optimizable parameters of the network.

1) Sparse coding

The aim is to determine the sparse code of a patch $y \in R^{\sqrt{p}\times\sqrt{p}}$ using a known dictionary $\boldsymbol{D} \in R^{p\times m}$, where y is represented as a column vector of length p. Equation (7) is the formulation for this objective.

$$\hat{\alpha} = \arg\min_{\boldsymbol{\alpha}} \lambda||\boldsymbol{\alpha}||_1 + \frac{1}{2}||\boldsymbol{D\alpha} - \boldsymbol{y}||_2^2 \tag{7}$$

ISTA is an efficient method for resolving the issue and is assured to reach the global optimum [16].

$$\hat{\boldsymbol{\alpha}}_{t+1} = S_{\lambda/c}(\hat{\boldsymbol{\alpha}}_t - \frac{1}{c}\boldsymbol{D}^T(\boldsymbol{D}\hat{\boldsymbol{\alpha}}_t - y)); \hat{\boldsymbol{\alpha}}_0 = 0 \tag{8}$$

where $S_{\lambda/c}$ is the soft-thresholding operator and c is the square spectral norm of $\boldsymbol{D}$:

$$[S_\theta(\boldsymbol{V})]_i = sign(v_i)(|v_i| - \theta)_+ \tag{9}$$

2) λ evaluation

Along with σ, the patch y itself affects the regularization parameter λ. Therefore, the regularization factor λ needs to be set for each y to obtain sparse representation with a controllable degree of error, i.e., $||\boldsymbol{D\alpha} - \boldsymbol{y}||_2^2 \leq p\sigma^2$. This function, $\lambda = f_\theta(y)$, is represented by a multi-layer perceptron (MLP) network, where θ is the parameters of the MLP. Three hidden layers make up the MLP, each of which is made up of a fully connected linear mapping and a ReLU.

3) Patch reconstruction

Reconstruct the denoised result $\hat{x}$ by multiplying the dictionary $\boldsymbol{D}$ with the sparse encoding $\hat{\boldsymbol{\alpha}}$:

$$\hat{x} = \boldsymbol{D}\hat{\alpha} \tag{10}$$

3.2.3. Patch averaging

In the last stage, the denoised patches $\{\hat{\boldsymbol{x}}_k\}_k$ are multiplied with the corresponding weights $\boldsymbol{w}$, obtaining the overall reconstructed data $\hat{\boldsymbol{X}}$:

$$\hat{\boldsymbol{X}} = \frac{\sum_k F_k^T(\boldsymbol{w} \odot \hat{\boldsymbol{x}}_k)}{\sum_k F_k^T\boldsymbol{w}} \tag{11}$$

where $\boldsymbol{w} \in R^{\sqrt{p}\times\sqrt{p}}$ denotes the weights of a patch, and $\odot$ is the Schur product.

To conclude, the deep K-SVD network $H(\cdot)$ is a parametrized function of the MLP parameters θ for computing λ, the step size c in the ISTA, the dictionary $\boldsymbol{D}$, and the weights $\boldsymbol{w}$ for the patch averaging. The training parameters of the model are relatively small. For instance, the total number of parameters is $p(4p + m + 3/2) + 1 = 32865$ for $p = 64$ and $m = 256$.

3.3. The MCLDNN

MCLDNN incorporates CNNs, LSTMs, and fully connected networks in a single structure to use complementary and synergistic properties for feature extraction and classification of modulated signals [11]. The three separate components are multi-channel and spatial characteristics mapping, temporal characteristics extraction, and fully connected network. Figure 2 shows the framework of the MCLDNN.

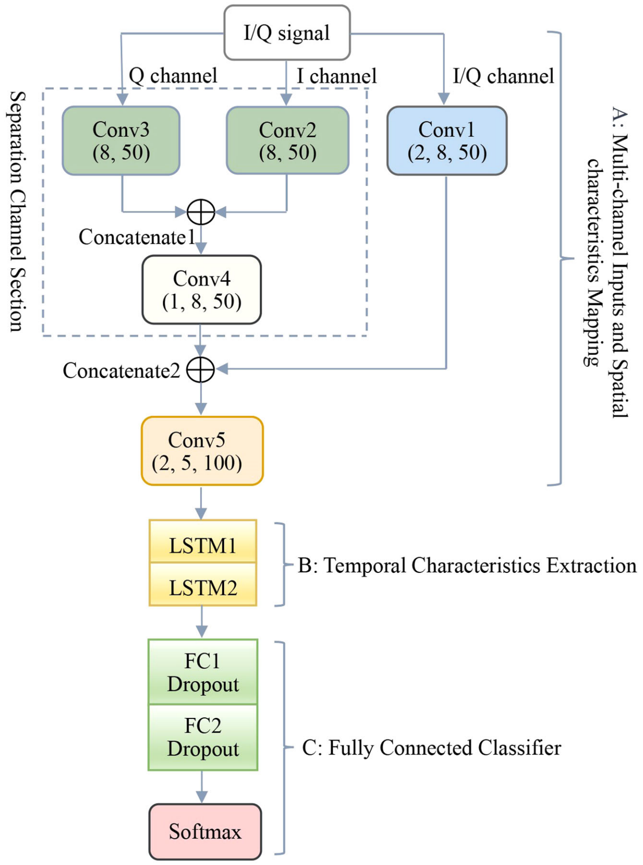
3.3.1. Feature extraction of I/Q signals

Three 2D convolutional layers (Conv1, Conv4, and Conv5) and two 1D convolutional layers (Conv2, Conv3) make up part A. The first step is to split the received I/Q signals into individual I and Q channel data. To learn the multi-channel and individual channel characteristics of signals, the three I/Q multi-channel, I channel, and Q channel data are input to Conv1, Conv2, and Conv3, respectively. To obtain the spatial correlations, they eventually merge in Concatenate2. The multi-channel input structure efficiently utilizes the complementary information of I, Q channel, and I/Q multi-channel signals by capturing details about the input representation at various scales.

3.3.2. Temporal characteristics extraction

Inspired by the network structure in Rajendran et al. [10] for wireless signal recognition based on distributed low-cost spectrum

Figure 2
The structure of MCLDNN



sensors, part B consists of two 128-cell LSTM networks that can efficiently process serial input and extract the temporal correlation.

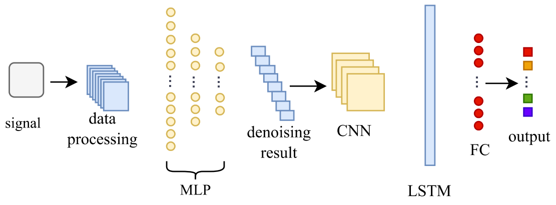
3.3.3. Fully connected classification

To map the characteristics to a more separable space, two fully connected networks with 128 neurons and scaled exponential linear units as activation functions are added in part C to expand the model. The dropout is also used to avoid the network overfitting. Each of the 11 neurons in the Softmax output layer corresponds to a different modulation scheme.

4. AMR Based on Deep K-SVD

The proposed architecture is shown in Figure 3, and the network consists of a denoising model and a classification model. The denoiser is the deep K-SVD network, which was initially used to denoise images with Gaussian white noise. Given that the modulated signal in the AMR problem also carries Gaussian white noise, the denoising process is completed using the modified deep K-SVD. In addition, the effectiveness of the denoiser for random sine wave denoising is demonstrated by simulation.

Figure 3
Automatic modulation recognition framework based on deep K-SVD



A deep K-SVD denoising model with modified internal details is first used to complete the denoising process of the original signal, skipping the complex image processing steps. The denoised signals are then fed into the MCLDNN, which integrates the different features of the data. Finally, classification results are obtained. In addition, to address the problem of slow convergence of ISTA during solving sparse coding, its accelerated version FISTA is used to improve the computational efficiency while obtaining a better sparse representation.

4.1. Radio signal denoising based on deep K-SVD

Algorithm 1: Fast Deep K-SVD

Input: The pairs of clean signals $\boldsymbol{X} \in R^{2\times128}$ and noisy signals $\boldsymbol{Y} \in R^{2\times128}$.
Output: The denoised signals $\hat{\boldsymbol{X}}$.
Parameters: The MLP parameters θ for computing λ, the step size c in the FISTA, the dictionary $\boldsymbol{D}$ and the weights $\boldsymbol{w}$ for the patch averaging.

$$\min_{\alpha_k,\boldsymbol{X}} \frac{\mu}{2}||\boldsymbol{X}-\boldsymbol{Y}||_2^2+\sum_k(\lambda_k||\boldsymbol{\alpha}_k||_0+\frac{1}{2}||\boldsymbol{D}\alpha_k-F_k\boldsymbol{X}||_2^2)$$

```
1 Initialization: Set $\boldsymbol{X}=\boldsymbol{Y}$, $\boldsymbol{D}=$ DCT matrix.
2 while the model does not converge do
3 Split the noisy data $\boldsymbol{Y} \in R^{2\times128}$ into tiny overlapped patches $\{\boldsymbol{y}_k\}_k \in R^{1\times64}$.
      $\boldsymbol{y}_k=F_k\boldsymbol{Y}$
4 Use the MLP to learn the regularization parameter $\lambda$, and FISTA to compute the sparse representation $\hat{\alpha}_k$ for each patch $\boldsymbol{y}_k$.
      $\hat{\boldsymbol{\alpha}}_k=\arg\min_{\boldsymbol{\alpha}_k}\lambda_k||\boldsymbol{\alpha}_k||_0+\frac{1}{2}||\boldsymbol{D}\boldsymbol{\alpha}_k-\boldsymbol{x}_k||_2^2$
5 Reconstruct the denoised patch $\hat{x}_k$ by multiplying the dictionary $\boldsymbol{D}$ with the sparse encoding $\hat{\boldsymbol{\alpha}}_k$:
      $\hat{x}_{\boldsymbol{k}}=\boldsymbol{D}\hat{\alpha}_k$
6 Average the patches to obtain the overall reconstructed data $\hat{\boldsymbol{X}}$:
      $\hat{\boldsymbol{X}}=\frac{\sum_k F_k^T(\boldsymbol{w}\odot\hat{\boldsymbol{x}}_k)}{\sum_k F_k^T\boldsymbol{w}}$
7 Back propagation and update parameters.
8 end
```

To address the problem that existing methods convert AMR to image recognition, leading to increased time cost and information loss, this article proposes a deep K-SVD denoising algorithm to denoise radio signals without introducing complex image processing steps. In addition, for the problem of slow convergence of ISTA for computing the sparse coding in the denoiser, FISTA is used to improve computational efficiency. The specific computational procedure is as follows. And Algorithm 1 describes how the denoiser's outputs are computed.

A certain amount of noise is added to the clean data to generate the corresponding noisy data. For the noisy signal $\boldsymbol{Y} \in R^{2\times 128}$, it is split into patches $\{\boldsymbol{y}_k\}_k$ of size 1×64, and its size is changed to 8×8, i.e., $\boldsymbol{y}_k \in R^{8\times 8}$. This process is similar to extracting small patches of Y using a sliding window of size 1×64, where the step size is 1.

Next, for each noisy patch $\boldsymbol{y}$, its sparse representation is given with ISTA. When solving the sparse encoding, the momentum term can be added to each iteration of Equation (8) to accelerate the convergence, i.e., the FISTA [16]:

$$\boldsymbol{\beta}_{t+1} = \hat{\boldsymbol{\alpha}}_t + \frac{T_t - 1}{T_{t+1}}(\hat{\boldsymbol{\alpha}}_t - \hat{\boldsymbol{\alpha}}_{t-1}),\ \boldsymbol{\beta}_1 = \hat{\boldsymbol{\alpha}}_0 = 0 \tag{12}$$

$$\hat{\boldsymbol{\alpha}}_{t+1} = S_{\lambda/c}(\boldsymbol{\beta}_{t+1} - \frac{1}{c}\boldsymbol{D}^T(\boldsymbol{D}\boldsymbol{\beta}_{t+1} - y)). \tag{13}$$

where $T_{t+1} = \frac{1+\sqrt{1+4T_t^2}}{2}$ and $t_1 = 1$.

The equation to be optimized is Equation (7), for which the regularization coefficient λ is learned with a triple-layer perceptron to generate a sparse representation of $\boldsymbol{y}$ under the control of error. After obtaining the sparse representation $\hat{\alpha}$ of the patch $\boldsymbol{y}$, it is multiplied with a predefined $\boldsymbol{D}$ to yield the denoising results $\hat{x}$ for each patch.

For each denoised $\{\hat{\boldsymbol{x}}_k\}_k$, the original signal size is restored by patch averaging. Specifically, the denoised patches are multiplied with the corresponding coefficients $\boldsymbol{w}$ and then placed in positions corresponding to the original signal. Later, the denoised version $\hat{\boldsymbol{X}}$ of the noisy signal $\boldsymbol{Y}$ is obtained.

4.2. Identification of denoised signals based on MCLDNN

To better classify the modulated signals, the multi-channel convolutional LSTM network is taken as the classifier. In previous work, only CNN or LSTM is employed for feature extraction. However, CNN is not skilled in extracting temporal information in serial data, and LSTM ignores the spatial features of the signal. Although O'Shea and West [18] built a CNN and LSTM model (convolutional long short-term neural network, CLDNN), it only utilized the data representation of I/Q channels and ignored the interactions between different features.

MCLDNN integrates CNN, LSTM in a single structure, and exploits the synergy and complementarity among them to extract and classify temporal and spatial features. MCLDNN first separates the I/Q multi-channel signals into individual I and Q channel, and then inputs them to CNN separately to learn the multi and single-channel features. The multi-channel structure extracts the input presentation features at various scales, exploiting the complementary data of I, Q channels and I/Q multi-channel. The outputs are then concatenated and fed into LSTMs with 128 cells, which is designed to derive the temporal correlation of the signals while efficiently processing the sequential data. Finally, the results of LSTMs are fed into a fully connected network for classification.

Cross entropy is taken as the loss function to calculate the difference between the true and predicted labels of the classifier, and the overall loss is the sum of the losses of the denoiser and classifier.

$$L = L_r + L_c \tag{14}$$

$$L_r = \frac{1}{2N}\sum\nolimits_{i=1}^{2}\sum\nolimits_{j=1}^{N}\left[H(\hat{\boldsymbol{x}})_{i,j} - \boldsymbol{x}_{i,j}\right]^2 \tag{15}$$

$$L_c = -\sum\nolimits_{k=1}^{K} p_k \ln \hat{p}_k \tag{16}$$

where $H(\cdot)$ represents the denoiser, $\boldsymbol{x}$ is the original data,$H(\hat{\boldsymbol{x}})$ is the denoised result, and the reconstruction loss L_r is the mean square error of the true value and the denoised data. L_c represents the cross-entropy loss, $p_k = 1$ when the label y belongs to the k-th modulation mode, otherwise $p_k = 0$.

5. Experiments and Results

5.1. Data and implementation details

The RadioML.2016.10a adopted for the experiments is a dataset generated by combining GNU Radio and commercially used modulation parameters. A total of 220,000 modulated signals are included, with SNR ranging from −20 to +18 dB (in 2 dB steps), and 11 commonly used modulations are considered: eight digital and three analog modulations. This includes BPSK, QPSK, 8PSK, 16QAM, 64QAM, BFSK, CPFSK, and PAM4 for digital modulation and WB-FM, AM-SSB, and AM-DSB for analog modulation. Each signal in the dataset has 128 complex floating-point times I/Q data, and the dataset also includes many realistic channel defects. There are more information about the specifications and generation details of the dataset in O'Shea et al. [19].

Split the dataset RadioML.2016.10a by a ratio of 6:2:2. Training the deep K-SVD model by taking the clean signal and its noisy signal with the addition of Gaussian white noise with certain SNR. For data of various modulation types with $-2 \leq \text{SNR} \leq 16$, the denoiser and classifier are trained simultaneously on the basis of using the pre-trained denoising network.

The inputs are pairs of clean and noisy signals, $-2 \leq \text{SNR} \leq 16$ noisy signals and corresponding labels. The outputs are the accuracy of signals with SNR in [−2, 16]. Among them, the signals with SNR between [−2, 16] are used to train the classifier after denoising, and the training is stopped when the training and validation loss tend to be smooth.

The model implementation is trained based on the PyTorch framework and NVIDIA Quadro P2000 GPU. In particular, the Adam optimization algorithm is utilized to optimize the model, with the learning rate set to 0.003 and the batch size of 64. In addition, the values reported in the experiments are the average of the results of 10 calculations of the model under the optimal parameter configuration.

5.2. Evaluation indicators and benchmark models

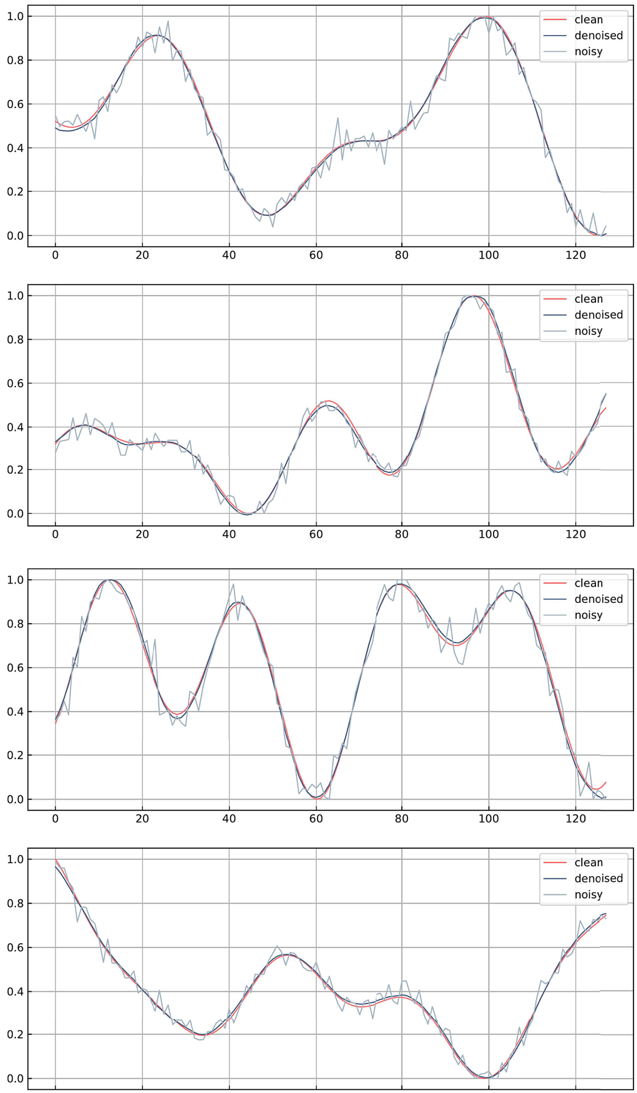
The metric uses the top 1 accuracy to determine the performance of the suggested framework. In addition, to further investigate what limits the recognition accuracy of radio signals with high SNRs, the confusion matrixes are utilized to show the intuitive classification effects of various modulation methods.

The denoiser using FISTA and the original denoiser are denoted as FDK-SVD (fast deep K-SVD) and DK-SVD (deep K-SVD), respectively. FDK-SVD and DK-SVD are compared with LSTM [10], CLDNN [12], CNN [19], and denoising auto-encoder (DAE) [20]. Among them, in LSTM, Rajendran et al. [10] proposed a new LSTM-based DL method combining time-domain amplitude and phase samples, and it achieved the best classification results on the benchmark dataset RadioML.2016.10a. CLDNN is a technique that applies deep neural networks studied for speech processing to wireless communication signals. The effectiveness of CLDNN for the AMR is verified by reviewing deep neural networks that may be useful for wireless communication applications and several simulation experiments [12]. The CNN classification model is trained on time-domain I and Q data and learns different matching filters for different SNRs [19]. To gather stable and robust features from noisy radio signals and apply the learned features for AMR, DAE is an automatic learning framework based on LSTM denoising self-encoders that uses amplitude and phase data for modulation recognition [20].

5.3. Results and discussions

To demonstrate the effectiveness of the deep K-SVD model for radio signal denoising, randomly generated sine waves are used for simulation experiments. Specifically, 1000 sine wave data are added with a certain SNR of Gaussian white noise, and then the paired clean and corresponding noisy data are fed into the denoiser. The mean square error of the clean and the denoised data is used as the loss function to train the deep K-SVD network. Figure 4 displays a few results from the simulation.

Figure 4
The denoised results of random sine wave



The results show that the denoiser can remove the Gaussian white noise from the noisy signals and recover the basic features of the signals. Given its good noise removal ability, it is applied to the denoised process in AMR.

The FDK-SVD and DK-SVD are trained in the SNR of −2 dB to 18 dB of the signal, and since radio signals with SNR less than −2 dB are more seriously contaminated by noise, these data are not available for training. The recognition accuracy of the model is illustrated in Figure 5, which compares the proposed method with LSTM [10], CLDNN [12], CNN [19], and DAE [20]. The performance of the proposed model is higher than the other four algorithms when SNR ≥ −2 dB. In the range of SNR from 0 dB to 18 dB, the average accuracy of FDK-SVD and DK-SVD reached 92.4% and 92.15%, respectively. In addition, Table 1 reports the classification accuracy of FDK-SVD and other comparison models.

Figure 6 shows the confusion matrixes of FDK-SVD at SNR of 14 dB and 16 dB. The vertical and horizontal coordinates of the confusion matrix represent the true and predicted labels, respectively, and the element c_{ij} of the confusion matrix indicates the division of the i-th into the j-th modulation mode. The confusion matrixes have two main confusion regions, one between analog modulation and the other between higher-order QAM. With SNR = 14 dB and SNR = 16 dB, the diagonals of the modes used for digital modulation (the first eight modulation modes) are relatively clear, even though it is challenging to distinguish the AM-DSB and WBFM signals. The main reason is the silent period of the audio since the modulated signal is generated from a real audio stream, and the last three modulation modes are analog modulation without much attention. In addition, a degree of confusion between QAM16 and QAM64 is noted at SNRs of 14 dB and 16 dB, as the former is a subset of the latter, and separation difficulties can arise.

Figure 5
Performance of the FDK-SVD and others based on RadioML.2016.10a

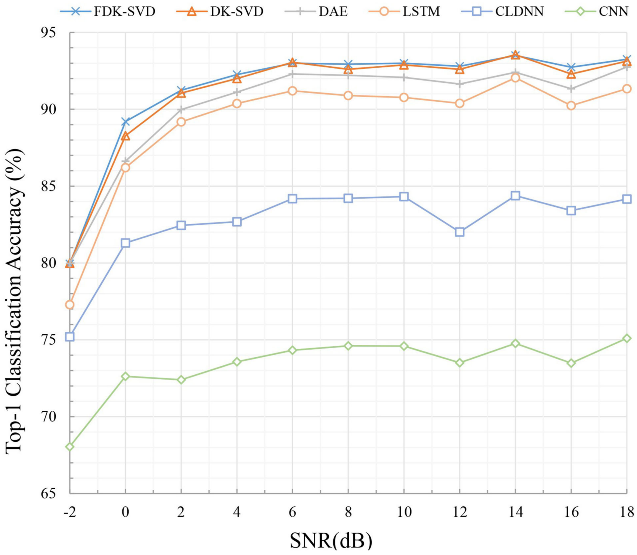


Table 1
Classification accuracy comparison of FDK-SVD vs. existing models on RadioML.2016.10a

Model	−2dB	0 dB	2 dB	4 dB
FDK-SVD	79.95	**89.2**	**91.23**	**92.26**
DK-SVD	**79.99**	88.28	91.05	92
DAE	79.97	86.62	89.98	91.11
LSTM	77.29	86.2	89.18	90.37
CLDNN	75.2	81.3	82.45	82.68
CNN	68.05	72.62	72.4	73.57
Model	6 dB	8 dB	10 dB	12 dB
FDK-SVD	93	**92.93**	**92.99**	**92.8**
DK-SVD	**93.06**	92.6	92.88	92.6
DAE	92.3	92.21	92.07	91.64
LSTM	91.2	90.89	90.77	90.39
CLDNN	84.18	84.21	84.32	82.02
CNN	74.32	74.61	74.59	73.51
Model	14 dB	16 dB	18 dB	Overall
FDK-SVD	93.5	**92.74**	**93.25**	**91.26**
DK-SVD	**93.55**	92.3	93.13	91.04
DAE	92.4	91.34	92.75	90.21
LSTM	92.05	90.24	91.33	89.08
CLDNN	84.38	83.41	84.16	82.57
CNN	74.76	73.49	75.10	73.37

Table 2
Computation time analysis of DK-SVD and FDK-SVD

Model	Objective SNR (dB)	Time (s)
DK-SVD	16.8	13
	17.6	21.5
FDK-SVD	16.8	10
	17.6	15

Table 2 displays the time comparison of DK-SVD and FDK-SVD with the same denoising effect. The experiment uses the paired clean and noisy signals for testing and records the computation time required for ISTA and FISTA to achieve the same denoising effect. When training the denoiser, the deep K-SVD combined with FISTA converges faster compared to ISTA, and the iteration time for calculating the sparse coding accounts for a more proportion of the overall denoising time, so the deep K-SVD model combined with FISTA has a shorter computation time under the condition of achieving the same SNR.

Further analysis of the convergence speed of ISTA and FISTA, Equation (1) for solving the sparse coding is denoted as:

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}_k} F(\boldsymbol{\alpha}) = \arg\min_{\boldsymbol{\alpha}_k} \lambda||\boldsymbol{\alpha}||_1 + \frac{1}{2}||\boldsymbol{D}\alpha - \boldsymbol{y}||_2^2 \tag{17}$$

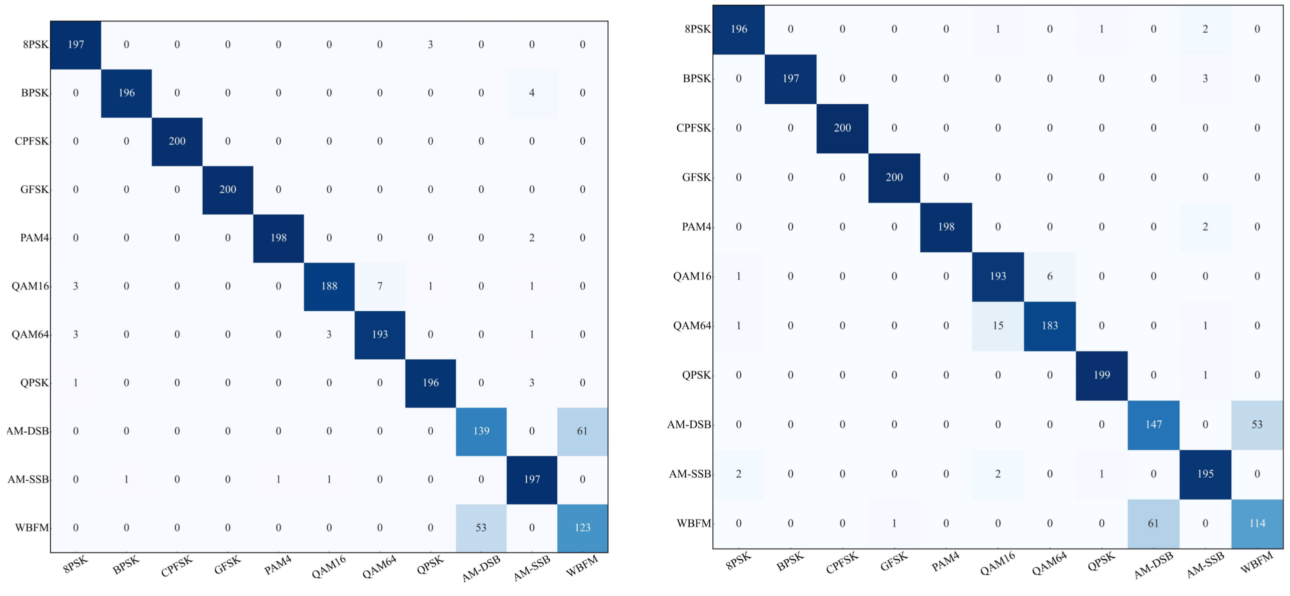
Then for any $k1$, the convergence rate of FISTA is [16]:

$$F(\boldsymbol{\alpha}_k) - F(\boldsymbol{\alpha}^*) \cong O(\frac{1}{k^2}) \tag{18}$$

where $\{\boldsymbol{\alpha}_k\}_k$ is the value of the sequence obtained by the FISTA. Compared to the convergence rate $O(\frac{1}{k})$ of ISTA, FISTA has a faster convergence rate, so it is more computationally efficient [16].

In conclusion, compared with other AMR methods, the performance of the AMR method based on deep K-SVD has been improved to some extent and has a better denoising ability for noisy radio signals.

Figure 6
Confusion matrix of FDK-SVD based on RML.2016.10a when (a) SNR = 14 dB and (b) SNR = 16 dB



	8PSK	BPSK	CPFSK	GFSK	PAM4	QAM16	QAM64	QPSK	AM-DSB	AM-SSB	WBFM
8PSK	197	0	0	0	0	0	0	3	0	0	0
BPSK	0	196	0	0	0	0	0	0	0	4	0
CPFSK	0	0	200	0	0	0	0	0	0	0	0
GFSK	0	0	0	200	0	0	0	0	0	0	0
PAM4	0	0	0	0	198	0	0	0	0	2	0
QAM16	3	0	0	0	0	188	7	1	0	1	0
QAM64	3	0	0	0	0	3	193	0	0	1	0
QPSK	1	0	0	0	0	0	0	196	0	3	0
AM-DSB	0	0	0	0	0	0	0	0	139	0	61
AM-SSB	0	1	0	0	1	1	0	0	0	197	0
WBFM	0	0	0	0	0	0	0	0	53	0	123

(a)

	8PSK	BPSK	CPFSK	GFSK	PAM4	QAM16	QAM64	QPSK	AM-DSB	AM-SSB	WBFM
8PSK	196	0	0	0	0	1	0	1	0	2	0
BPSK	0	197	0	0	0	0	0	0	0	3	0
CPFSK	0	0	200	0	0	0	0	0	0	0	0
GFSK	0	0	0	200	0	0	0	0	0	0	0
PAM4	0	0	0	0	198	0	0	0	0	2	0
QAM16	1	0	0	0	0	193	6	0	0	0	0
QAM64	1	0	0	0	0	15	183	0	0	1	0
QPSK	0	0	0	0	0	0	0	199	0	1	0
AM-DSB	0	0	0	0	0	0	0	0	147	0	53
AM-SSB	2	0	0	0	0	2	0	1	0	195	0
WBFM	0	0	0	1	0	0	0	0	61	0	114

(b)

6. Conclusion

In this article, an AMR method based on the deep K-SVD denoising algorithm is proposed. To address the problem that existing methods convert AMR into studied image recognition but introduce complicated image processing and increase the computation time, a deep K-SVD model without image conversion is proposed to denoise the radio signal. In addition, the FISTA is used in the denoiser to improve computational efficiency. Experiments show that the proposed model FDK-SVD has an average classification accuracy of 91.26% at different SNRs from −2 dB to 18 dB, which is 1.05 to 8.69 percentage points better than the mainstream modulation recognition models such as LSTM and CLDNN. Finally, considering the increasingly complex electromagnetic environment, network design under signal mixing scenarios will be the future research direction.

Funding Support

This work was sponsored by the Fundamental Research Funds for the Central Universities (2682020ZT107); National Natural Science Foundation of China (62106206); Humanities and Social Sciences Project of the Ministry of Education (19YJCZH048, 20XJCZH016); and Sichuan Provincial Science and Technology Program (2023YFH0066).

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Author Contribution Statement

Yanhe Li: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Xingxing He:** Resources, Writing – review & editing, Supervision, Funding acquisition. **Chun Zhou:** Writing – review & editing, Project administration.

References

[1] He, C., Chen, J., Jin, Z., & Lei, Y. (2023). Automatic modulation recognition based on multimodal time-frequency feature fusion. *Computer Science*, *50*(4), 226–232. https://doi.org/10.11896/jsjkx.220600242

[2] Jiao, X., Wei, X. L., Xue, Y., Wang, C., & Duan, Q. (2022). Automatic modulation recognition based on deep learning. *Computer Science*, *49*(5), 266–278. https://doi.org/10.11896/jsjkx.211000085

[3] Han, G., Li, J. D., & Lu, D. H. (2004). Study of modulation recognition based on HOCs and SVM. In *2004 IEEE 59th Vehicular Technology Conference, 2*, 898–902. https://doi.org/10.1109/VETECS.2004.1388960

[4] Hong, L., & Ho, K. C. (2000). BPSK and QPSK modulation classification with unknown signal level. In *MILCOM 2000 Proceedings of 21st Century Military Communications. Architectures and Technologies for Information Superiority, 2*, 976–980. https://doi.org/10.1109/MILCOM.2000.904076

[5] Lunden, J., & Koivunen, V. (2007). Automatic radar waveform recognition. *IEEE Journal of Selected Topics in Signal Processing*, *1*(1), 124–136. https://doi.org/10.1109/JSTSP.2007.897055

[6] Li, R., Li, L., Yang, S., & Li, S. (2018). Robust automated VHF modulation recognition based on deep convolutional neural networks. *IEEE Communications Letters*, *22*(5), 946–949. https://doi.org/10.1109/LCOMM.2018.2809732

[7] Peng, S., Jiang, H., Wang, H., Alwageed, H., Zhou, Y., Sebdani, M. M., & Yao, Y. D. (2019). Modulation classification based on signal constellation diagrams and deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(3), 718–727. https://doi.org/10.1109/TNNLS.2018.2850703

[8] Zhang, Z., Wang, C., Gan, C., Sun, S., & Wang, M. (2019). Automatic modulation classification using convolutional neural network with features fusion of SPWVD and BJD. *IEEE Transactions on Signal and Information Processing over Networks*, *5*(3), 469–478. https://doi.org/10.1109/TSIPN.2019.2900201

[9] Zhang, J., Hou, J., & Chen, G. (2022). jī yú lián hé juàn jī yǔ jì yì shén jīng wǎng luò de xìn hào zì dòng tiáo zhì shí bié [Automatic modulation recognition for signals based on convolutional and memory neural network]. *Computer Applications and Software*, *39*(08), 226–233.

[10] Rajendran, S., Meert, W., Giustiniano, D., Lenders, V., & Pollin, S. (2018). Deep learning models for wireless signal classification with distributed low-cost spectrum sensors. *IEEE Transactions on Cognitive Communications and Networking*, *4*(3), 433–445. https://doi.org/10.1109/TCCN.2018.2835460

[11] Xu, J., Luo, C., Parr, G., & Luo, Y. (2020). A spatiotemporal multi-channel learning framework for automatic modulation recognition. *IEEE Wireless Communications Letters*, *9*(10), 1629–1632. https://doi.org/10.1109/LWC.2020.2999453

[12] West, N. E., & O'Shea, T. (2017). Deep architectures for modulation recognition. In *2017 IEEE International Symposium on Dynamic Spectrum Access Networks*, 1–6. https://doi.org/10.1109/DySPAN.2017.7920754

[13] Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, *54*(11), 4311–4322. https://doi.org/10.1109/TSP.2006.881199

[14] Elad, M., & Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, *15*(12), 3736–3745. https://doi.org/10.1109/TIP.2006.881969

[15] Pati, Y. C., Rezaiifar, R., & Krishnaprasad, P. S. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers, 1*, 40–44. https://doi.org/10.1109/acssc.1993.342465

[16] Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, *2*(1), 183–202. https://doi.org/10.1137/080716542

[17] Gregor, K., & LeCun, Y. (2010). Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning*, 399–406.

[18] O'Shea, T., & West, N. (2016). Radio machine learning dataset generation with GNU radio. *Proceedings of the GNU Radio Conference, 1*(1).

[19] O'Shea, T. J., Corgan, J., & Clancy, T. C. (2016). Convolutional radio modulation recognition networks. In *Engineering Applications of Neural Networks: 17th International Conference*, 213–226. https://doi.org/10.1007/978-3-319-44188-7_16

[20] Ke, Z., & Vikalo, H. (2022). Real-time radio technology and modulation classification via an LSTM auto-encoder. *IEEE Transactions on Wireless Communications*, *21*(1), 370–382. https://doi.org/10.1109/TWC.2021.3095855

How to Cite: Li, Y., He, X., & Zhou, C. (2025). Automatic Modulation Recognition Based on a New Deep K-SVD Denoising Algorithm. *Journal of Data Science and Intelligent Systems*, *3*(1), 18–26. https://doi.org/10.47852/bonviewJDSIS32021244