Hindawi
Mobile Information Systems
Volume 2021, Article ID 6870938, 6 pages
https://doi.org/10.1155/2021/6870938

*Research Article*

# Deep Feature Autoextraction Method for Intrapulse Data of Radar Emitter Signal

**Shiqiang Wang**,[1] **Caiyun Gao**,[2] **Chang Luo**,[3,4] **Huiyong Zeng**,[1] **Guimei Zheng**,[1] **Qin Zhang**,[1] **Juan Bai**,[1] **and Binfeng Zong**[1]

[1]*Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China*
[2]*Department of Basic Science, Air Force Engineering University, Xi'an 710051, China*
[3]*Troops of 78092, Chengdu 610000, China*
[4]*National Defence University, Joint Operations College, Beijing 100080, China*

Correspondence should be addressed to Caiyun Gao; jianjiangchen@126.com

Received 21 June 2021; Revised 9 August 2021; Accepted 16 August 2021; Published 27 August 2021

Academic Editor: Sang-Bing Tsai

Concerned with the problems that the extracted features are the absence of objectivity for radar emitter signal intrapulse data because of relying on priori knowledge, a novel method is proposed. First, this method gets the sparse autoencoder by adding certain restrain to the autoencoder. Second, by optimizing the sparse autoencoder and confirming the training scheme, intrapulse deep features are autoextracted with encoder layer parameters. The method extracts the eigenvectors of six typical radar emitter signals and uses them as inputs to a support vector machine classifier. The experimental results show that the method has higher accuracy in the case of large signal-to-noise ratio. The simulation verifies that the extracted features are feasible.

## 1. Introduction

The key to sorting and identification of radar signals effectively is to extract the features that can reflect the nature of the signal. In recent years, scholars have constructed features or proposed feature extraction methods for sorting and identification signal, such as features extracted via fractional Fourier transform (FRT) [1], scale-invariant feature transform (SIFT) features [2], low probability of sorting (LPS) features [3], fusion image feature [4], compressed sensing mask (CS-mask) feature [5], feature extraction algorithm using probability moment and ApEn [6], Manhattan distance-based features [7], and other features. Design features are a good way to sort or identify radar signals with exploit human wisdom and prior knowledge.

The autoencoder (AE) under the deep learning theory aims to reconstruct the original input signal at the output layer. When extracting the distributed features of the data, the AE does not require additional supervisory information, and also, it can avoid the subjectivity when designing features. The research of AE has been a hot topic of concern in recent years, and many variation and application of the autoencoder has been developed [8–12]. Hinton improved the structure of the AE prototype and obtained the deep autoencoder (DAE) [13].

Also, the different deep structure and the optimization of the cost function are constructed. In [14], Bengio develops the AE depth and proposes the sparse autoencoder (SAE), which divides the hidden structure by adding sparseness constraints to the hidden layer nodes.

With different sparse punitive functions [15], different numbers of nodes in hidden layers [16], and preprocessing methods [17], SAE has different performance. The use of sparse encoders can not only carry out deep feature extraction [18] but also can complete the defect detection, classification, and blind source separation work [17].

With development of modern radar toward multifunctional, multipurpose, and multisystem, the design of signal waveform is increasingly complex, and the signal regularity has also been damaged severely. So, relying on experience to design features is not enough to be competent for the task of radar signal in pulse feature extraction in the current electromagnetic environment. Therefore, if we use SAE to complete this task, it is expected to break through the shackles of the conventional method by extracting the inherent features of pulse signals.

By studying the SAE framework, this study proposed a method of extracting the deep pulse features of the radar signal using SAE. First, the SAE frame is analyzed and obtained by applying the specific sparse constraint. Second, by optimizing the SAE and determining the training program, the radar signal deep pulse features are extracted automatically from coding layer parameters. Finally, the validity of the extracted features is proved by experiments.

## 2. Sparse Autoencoder Mechanism

The AE is a deep learning architecture that includes coding and decoding (the architecture as shown in Figure 1).

The coding refers to get the representation of the middle layer with the original data as the network input and to get the code of the hidden layer. The decoding means that the intermediate layer features are decoded by the hidden layer, and the original input is rebuilt via the output layer. Through the encoding and decoding mechanism, the encoder makes the reconstructed error of the reconstructed signal small. Since it reconstructs the original input at the output layer as the target and no additional supervisory information is needed, the data feature can be learned from the original data automatically.

*2.1. Autoencoder (AE) Theory.* The basic theory of the AE can be summarized as follows: supposing an unlabelled training is set to $\boldsymbol{x} = \{x^{(1)}, x^{(2)}, x^{(3)}, \ldots\}$, where$x^{(i)} \in \mathfrak{R}^n$. As a neural network, the AE uses back propagation for unsupervised learning. The learning task is to make the output value equal to the original input, that means $y^{(i)} = x^{(i)}$. Prototype of AE is shown in Figure 2:

According to the meaning of an autoencoder, by learning knowledge, the function $h_{W,b}(x) \approx x$ would be captured. If the training set contains $m$ samples, training the special neural network AE shown in Figure 2 using the gradient descent method, we get the loss function for a single training sample $(x, y)$:

$$J(W, b; x, y) = \frac{1}{2}\left\| h_{W,b}(x) - y \right\|^2. \tag{1}$$

The loss function of the entire network (training set) is expressed as follows:

$$\begin{aligned} J(W, b) &= \left[\frac{1}{m}\sum_{i=1}^{m} J\left(W, b; x^{(i)}, y^{(i)}\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{l=1}^{s_l+1}\left(W_{ji}^{(l)}\right)^2 \\ &= \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\| h_{W,b}\left(x^{(i)}\right) - y^{(i)} \right\|^2\right)\right] \\ &\quad + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{l=1}^{s_l+1}\left(W_{ji}^{(l)}\right)^2. \end{aligned} \tag{2}$$

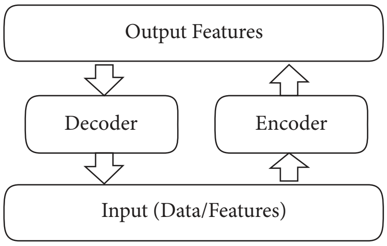

Figure 1: Autoencoder frame.

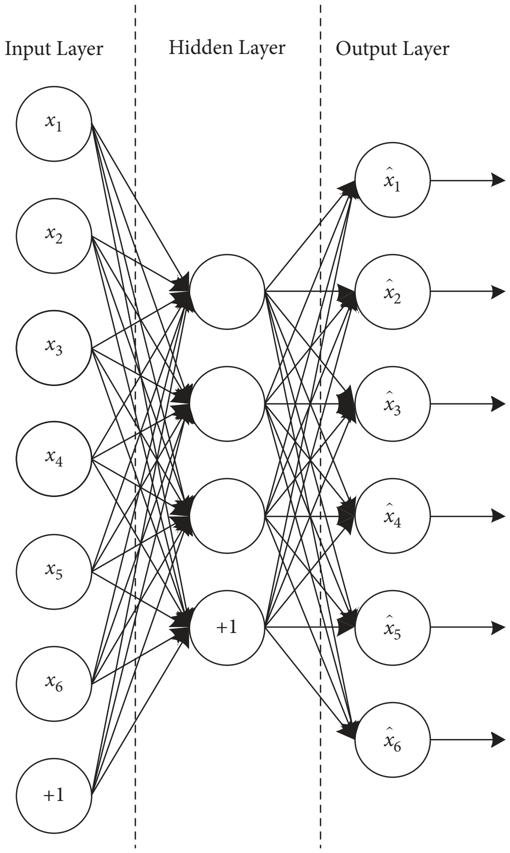

Figure 2: Prototype autoencoder.

In formula (2), the first item is the sum squared error term and the second item is the regularization term (also known as the weight decay term), which tends to reduce the weight and help prevent overfitting. The weight decay parameter $\lambda$ dominates the relative importance of these two items. Also, the slightly overloaded symbols should be noted: formula (1) is the error cost squared for a single example, and formula (2) is the total cost function, including the weight decay item. Also, in formula (2), the symbol $W_{ji}^{(l)}$ represents the relevance of parameters between unit $j$ in layer $l$ and unit $i$ in layer $l+1$. The symbol $b$ denotes the bias, $h_{W,b}(x)$ is the encoder output and is the function of the activation value, connection parameter $W$, and bias term. Via the encoder formula (2), we could achieve the minimum value using $W$ and $b$ as the parameters. If the activation value could reconstruct its original input properly, it is assumed that it retains most of the information contained in the original data.

If we simply keep the radar pulse modulation information, it is not enough to get useful features from the encoder. That is, an autoencoder (AE) with the same input and output dimensions only needs to learn a simple constant function for achieving the perfect reconstruction of the data. But, in fact, it is expected that an AE can learn a more complex nonlinear function. So, it is necessary to give the AE a certain constraint to learn a better feature representation.

*2.2. Analysis of Sparse Autoencoder.* Consider two cases. One is that if the nodes of the input layer are more than the hidden ones, then the network must learn the compressed representation of input, which means to give the hidden layer node activation value as a vector element. It needs to reconstruct the larger dimension input with the vector. The other one is that if the nodes of the hidden layer are more, even more than the input ones, then it is necessary to impose some constraints on the network so as to find the structure of the original input. Here, the hidden nodes work with sparseness constraints.

Sparseness constraint is an important type of constraint that makes the learned expression more meaningful. Such an AE is called deep sparse autoencoders (DSAE), SAE in short.

The implementation of SAE mainly includes three important aspects, namely, the application of specific sparseness constraints, optimization of SAE structure, and determination of the DAE training schemes. Therefore, when optimizing the deep SAE for extracting the features of the pulse data, it is necessary to add the sparse constraint, and then, the hidden layers and neurons are increased, the distribution of hidden layer nodes is adjusted, and the sharing method of weight values is changed. Through these operations, the basic framework of DAE is optimized. Finally, according to the different requirements, the appropriate cost function, optimization strategy [19], hidden layer quality factor, and parameter optimization performance index are selected to determine the training scheme of DAE.

Assuming that the activation value of the hidden layer node $j$ is denoted as $a_j^{(2)}(x)$ when the input is $x$, formula $\hat{\rho}_j = (1/m)\sum_{i=1}^{m}[a_j^{(2)}(x^{(i)})]$ indicates the average activation of the hidden layer node $j$ [20]. The specific sparse constraint $\hat{\rho}_j = \rho$ is added, where $\rho$ denotes the sparse parameter and usually is close to 0 (e.g., $\rho = 0.05$). That means, if this constraint is satisfied, the activation of the hidden layer node must be approximate to zero [21]. To reach this purpose, when optimizing the objective function, $\hat{\rho}_j$ with a large deviation from the sparse parameters $\rho$ is needed to be punished. Usually, we use KL distances as a penalty item:

$$\mathrm{KL}\left(\rho \middle\| \hat{\rho}_j\right) = \rho \log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}. \tag{3}$$

According to the loss function of the autoencoder and the sparseness requirement, the SAE loss function expression is

$$J_{\text{sparse}}(W,b) = J(W,b) + \beta\sum_{j=1}^{s_2}\mathrm{KL}\left(\rho \middle\| \hat{\rho}_j\right). \tag{4}$$

In formula (4), $s_2$ denotes the hidden neurons numbers, and $\beta$ is used as the parameter to control the proportion of the sparse penalty term [21].

Next, the function $J(W,b)$ would be solved, and the minimum value would be seeked out under different $W$ and $b$.

For the purpose of solving the neural network, the parameters of $W_{ji}^{(l)}$ and $b_i^{(l)}$ should be initialized to an initial random number close to zero. Then, the optimization algorithm similar to the bulk gradient descent method is used for the objective function, and finally, the parameter matrix of the whole network is obtained.

In order to restrict all connection parameters $W$ and bias items to a specific data space, the SAE should be pretrained.

Through this step, the reduction of the quality factor randomly initialized for the hidden layer is prevented and for the purpose of facilitating to optimize these parameters of the whole neural network systematically. When pretraining the SAE, it is very important to initialize the SAE input and hidden layers in an unsupervised way and then to train each hidden layer as an autocorrelator with the greedy layer-wise pretraining algorithm to reconstruct the input data [22].

## 3. Deep Feature Extraction of Radar Signal with SAE

Different radar emitter signals have different intrapulse features. The sparse autoencoder (SAE) is used for feature extraction, and there is no need to define features in advance. When the intrapulse data are input into the network, the network will automatically learn to obtain the various levels of the input radar emitter signal, that means the feature expression.

As a deep learning framework, SAE builds a multilayer network layer by layer, so that the machine can learn the relationship reflected in the data automatically. SAE learns features with better generalization and expressiveness ability in this way. In other words, the SAE combines low-level features to achieve implicit data feature expression, thus forming more abstract high-level features or expressions.

*3.1. Deep Intrapulse Feature Extraction Framework.* For the radar emitter signal pulse sequence entering the reconnaissance and receiving system, because of their approximate short-term stability, it is considered that the adjacent continuous multiframe short-time samples are concatenated to obtain long-term samples, which constitute the original input of the network. Considering that it is necessary to describe the complex data of the radar signal and the training requirement of the subsequent sorting model for the extracted deep intrapulse feature, the intermediate coding layer uses the Gaussian-type node. The remaining hidden layer adopts the Bernoulli-type node. Based on optimization of SAE, the radar signal deep intrapulse feature extraction framework is shown in Figure 3.

For Gaussian-type nodes, the output is a linear combination of inputs that satisfies

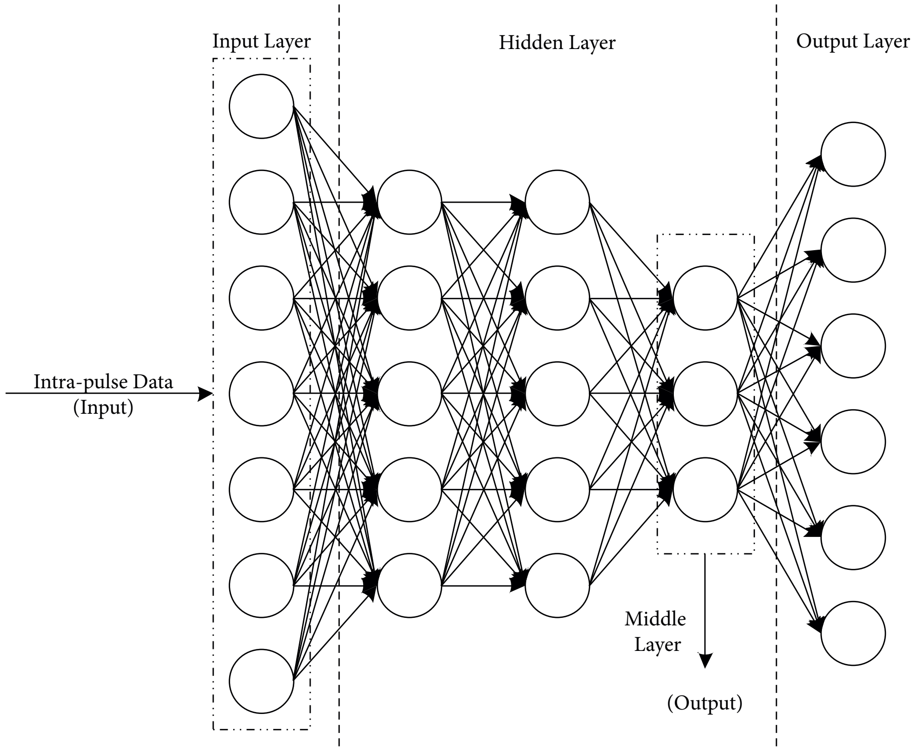

Figure 3: Deep intrapulse feature extraction frame.

$$y_i^l = f\left(x_i^l\right) = \sum_j w_{ij} y_i^{l-1} + b_i^l. \tag{5}$$

For Bernoulli-type nodes, the output is the sigmoid map for the input that satisfies

$$y_i^l = \sigma\left(x_i^l\right) = \sigma\left(\sum_j w_{ij} y_i^{l-1} + b_i^l\right) = \frac{1}{1 + \exp\left(-\sum_j w_{ij} y_i^{l-1} - b_i^l\right)}, \tag{6}$$

where $x_i^l$ and $y_i^l$ represent the node $i$ of the first layer for source input and output, respectively, $b_i^l$ represents the offset of the node, $w_{ij}$ represents the connection weight between $i$ and the next node, and $y_i^{l-1}$ represents the output value of the next node.

SAE uses the minimum error between the reconstructed input and the original input as the objective function and then adjusts the network parameters using the back-propogation (BP) algorithm. The objective function is written as

$$J(\theta) = \frac{1}{m}\left\|\mathbf{x} - f_{\text{dec}}\left(f_{\text{enc}}(\mathbf{x})\right)\right\|, \tag{7}$$

where $\theta$ denotes the network parameter, $m$ denotes the training sample amount, $\boldsymbol{x}$ denotes the original input of the network, $f_{\text{enc}}(\boldsymbol{x})$ is the middle layer coding output of the network, and $f_{\text{dec}}(f_{\text{enc}}(\mathbf{x}))$ is the middle layer input coding result reconstructed by the decoding network.

*3.2. Deep Intrapulse Features Automatic Extraction.* After training the deep SAE, fine-tuning is needed for the network, and it is a necessary step to optimize the deep SAE, where the BP algorithm is used to complete this task. When fine-tuning, the input layer, output layer, and all hidden layers are taken as a whole, and then, the supervised learning method is used to adjust the pretrained neural network [23]. Then, the ownership and bias are optimized after several iterations at last. Through this process, we can complete the radar emitter signal hierarchical feature extraction. The steps of feature automatic extraction are as follows:

Step 1: initialize the network by assigning weight values and thresholds

Step 2: select the class data samples randomly with the algorithm for training the SAE and then calculate the output of each layer

Step 3: find the reconstruction error of each layer which is used for adjusting weight values and deviations

Step 4: determine whether the reconstruction error satisfies designing demands and repeat steps 2 and 3 if the requirements are not met until the entire SAE output meets designing demands

Step 5: use the encoding layer parameters to map the original input to get new features, that means$\mathbf{y} = f(\mathbf{x}; \theta_{\text{encode}})$

In Step 5, x denotes the original radar signal data input, $\theta_{\text{encode}}$ denotes the network parameter of the encoding part, and $\boldsymbol{y}$ denotes the intermediate layer feature vector based on deep intrapulse features extraction.

By using SAE to extract the pulse features automatically, we can extract the deep interpretation factor of the dense radar signal sample. Also, we can keep the original input nonzero feature, increase the algorithm robustness, and enhance the linear separability of the pulse signal, so that the classification boundary becomes clearer, the scale of the variable can be control to a certain extent, the structure of the given input data can be changed, and the original information can be enriched. Finally, the comprehensiveness and accuracy of the information expression can be improved.

## 4. Experimentation and Analysis

In this study, 6 typical radar emitter signals are selected for simulation experiments. The 6 signals are conventional waveform (CW), linear frequency modulation (LFM), nonlinear frequency modulation (NLFM), binary phase-shift keying (BPSK), quaternary phase-shift keying (QPSK), and frequency-shift keying (FSK) signal. For these signals, the pulse width adopts 10.8 μs, the carrier frequency adopts 0.85 GHz, the sampling frequency adopts 2400 MHz, the modulation bandwidth of LFM adopts 0.045 GHz, the NLFM adopts sinusoidal frequency modulation, BPSK adopts 31 bit pseudorandom code, QPSK adopts Huffman code, and FSK adopts the Barker code. For each radar signal, there are 120 samples for every 5 dB in the range of 0–20 dB SNR, total samples number is 600, of which 200 samples are used as the training data of the classifier, and the other 400 data are used as test data for signal identification. Before training the classifier and testing signal classification recognition effect, all samples are extracted with deep intrapulse feature extraction. In order to reflect the feature distribution of each radiation source signal, this study selects 60 datasets of feature samples with typical SNR (15 dB) of each signal from these extracted eigenvectors. A total of 300 feature samples distribution are shown Figure 4.

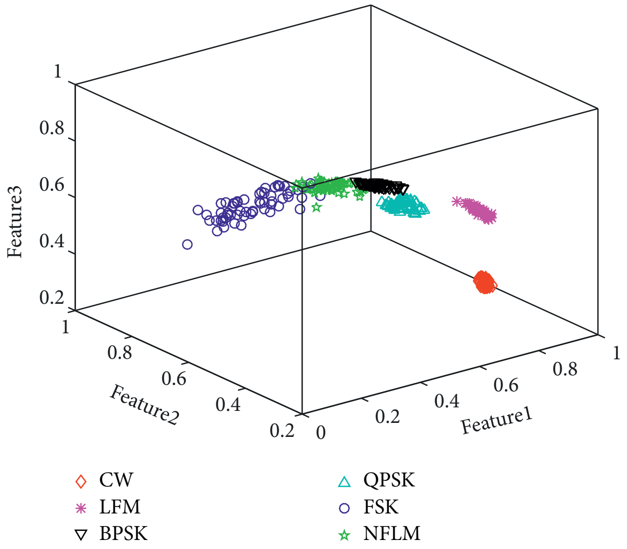

Figure 4: Deep feature scattering of emitter signal.

The conclusion can be drawn via Figure 4 that these three-dimensional deep features of the two signal of CW and LFM are better. The deep features of NLFM, BPSK, and QPSK are clustering well, but between different signals, the features overlap partially. As to the FSK signal, it has poor intraclass feature clustering and overlap with NLFM. Figure 4 shows that with optimized SAE, we can extract different radar signal deep features of interclass separation and intraclass aggregation. For the purpose of verifying the validity of the extracted deep feature further, the SVM is used to classify and identify the radar emitter signals denoted by the deep feature vector.

The correct classification recognition rate obtained by SVM with different SNRs is given Table 1. The result adopts the mean value of 20 test results, and the average result is the mean value of each signal classification result between 0 and 20 dB SNR, where ARR represents the average recognition rate.

Table 1: Comparison of recognition accuracy under different SNRs.

| Signal type | Classification recognition rate (%) | | | | | |
|---|---|---|---|---|---|---|
| | 0 dB | 5 dB | 10 dB | 15 dB | 20 dB | ARR (%) |
| CW | 96.04 | 98.87 | 100 | 100 | 100 | 98.98 |
| LFM | 88.92 | 100 | 100 | 100 | 100 | 97.78 |
| BPSK | 82.31 | 84.52 | 89.92 | 96.63 | 100 | 90.68 |
| QPSK | 86.02 | 91.61 | 95.09 | 98.59 | 100 | 94.26 |
| FSK | 77.43 | 85.37 | 89.18 | 94.16 | 98.55 | 88.94 |
| NLFM | 81.15 | 84.62 | 93.62 | 98.13 | 100 | 91.50 |

It can be seen from Table 1 that in a special range of SNR, we can classify and identify the radar emitter signal well using the SVM classifier with the extracted deep feature as the eigenvector. Each radar emitter signal can obtain the high correct recognition rate. The signal recognition rate relates to signal complexity. For relatively simple signal forms such as CW and LFM-modulated signals, the average correct recognition rate can be up to 98.98% and 97.78%; for more complex signal forms such as FSK modulated signal, the average correct recognition rate is 88.94%, which is related to poor clustering of the deep feature and the partial overlap of features, but this result is acceptable in engineering applications. In addition, the average correct recognition result of these 6 types of emitter signals reach to 93.69%, and the recognition performance is good.

## 5. Conclusions

The key to sort and identify the radar signal efficiently is to extract the features which can reflect the signal intrinsic features. Aiming at the problem of lack of objectivity due to dependence on a priori knowledge in the extraction of radar signal features, an automatic extraction method of the deep pulse feature is proposed. This method optimizes the sparse autoencoder at the beginning and then extracts the deep pulse feature of the radar signal using the coding layer parameters automatically. The support vector machine is used to classify and identify the typical radiation source signals characterized by deep pulse features. The results show that the satisfactory results can be obtained in a large scale of signal-to-noise ratio, and the method of features extracted automatically is verified to be valid in this study.

## Data Availability

All data, models, and codes generated or used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (61971438) and in part by the Nature Science Foundation of Shaanxi Province (2020JM-345 and 2020JQ-482).

## References

[1] T. Ravi Kishore and K. D. Rao, "Automatic intrapulse modulation classification of advanced lpi radar waveforms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 2, pp. 901–914, 2017.

[2] S. Liu, X. Yan, P. Li, X. Hao, and K. Wang, "Radar emitter recognition based on sift position and scale features," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 12, pp. 2062–2066, 2018.

[3] F. Wang, Y. Zhang, and Y. Wang, "Reliable LPS features of multiple airborne radars against PRS," *IET Radar, Sonar & Navigation*, vol. 13, no. 10, pp. 1747–1754, 2019.

[4] L. Gao, X. Zhang, J. Gao, and S. You, "Fusion image based radar signal feature extraction and modulation recognition," *IEEE Access*, vol. 7, pp. 13135–13148, 2019.

[5] M. Zhu, X. Zhang, Y. Qi, and H. Ji, "Compressed sensing mask feature in time-frequency domain for civil flight radar emitter recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2146–2150, Calgary, AB, Canada, April 2018.

[6] C. M. Jeong, Y. G. Jung, and S. J. Lee, "Neural network-based radar signal classification system using probability moment and ApEn," *Soft Computing*, vol. 22, no. 13, pp. 4205–4219, 2018.

[7] Y. Huang, W. Jin, B. Li, P. Ge, and Y. Wu, "Automatic modulation recognition of radar signals based on manhattan distance-based features," *IEEE Access*, vol. 7, pp. 41193–41204, 2019.

[8] C. Aytekin, X. Ni, F. Cricri, and E. Aksu, "Clustering and unsupervised anomaly detection with l2 normalized deep auto-encoder representations," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, Shenzhen, China, June 2018.

[9] A. L. Caterini, A. Doucet, and D. Sejdinovic, *Hamiltonian Variational Auto-Encoder*, NeurIPS, Montréal, Canada, 2018.

[10] J. Feng and Z. Zhou, "Auto-encoder by forest," in *Proceedings of the The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pp. 2967–2973, Orleans, LA, USA, February 2018.

[11] H. Shao, H. Jiang, L. Xingqiu, and W. Shuaipeng, "Intelligent fault diagnosis of rolling bearing using deep wavelet auto-encoder with extreme learning machine," *Knowl.-Based Syst.*vol. 140, pp. 1–14, 2018.

[12] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: point cloud auto-encoder via deep grid deformation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 206–215, Salt Lake City, UT, USA, June 2017.

[13] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[14] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising auto-encoders as generative models," *News in Physiological Sciences*, vol. 1, pp. 899–907, 2013.

[15] N. Jiang, W. Rong, B. Peng, Y. Nie, and Z. Xiong, "An empirical analysis of different sparse penalties for autoencoder in unsupervised feature learning," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Killarney, Ireland, July 2015.

[16] Q. Xu and L. Zhang, "The effect of different hidden unit number of sparse auto-encoder," in *Proceedings of the 27th Chinese Control and Decision Conference (CCDC)*, pp. 2464–2467, Qingdao, China, May 2015.

[17] L. Luo, H. Su, and L. Ban, "Independent component analysis-Based sparse autoencoder in the application of fault diagnosis," *Proceeding of the 11th World Congress on Intelligent Control and Automation*, vol. 99, pp. 1378–1382, 2014.

[18] H. Xie, S. Wang, K. Liu, S. Lin, and B. Hou, "Multilayer feature learning for polarimetric synthetic radar data classification," *IEEE Geoscience and Remote Sensing Symposium*, vol. 120, pp. 2818–2821, 2014.

[19] J. Ma, S. Ni, W. Xie, and W. Dong, "Deep auto-encoder observer multiple-model fast aircraft actuator fault diagnosis algorithm," *International Journal of Control, Automation and Systems*, vol. 15, no. 4, pp. 1641–1650, 2017.

[20] X. Guo, A. A. Minai, and L. J. Lu, "Feature selection using multiple auto-encoders," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 4602–4609, IEEE, Anchorage, AK, USA, May 2017.

[21] Y.-B. Wang, Z.-H. You, X. Li et al., "Predicting protein-protein interactions from protein sequences by a stacked sparse autoencoder deep neural network," *Molecular BioSystems*, vol. 13, no. 7, pp. 1336–1344, 2017.

[22] H. H. Yang, Z. C. Luo, S. J. Jiang, X. B. Zhang, and L. H. Yin, "Sparse denoising autoencoder application in identification of counterfeit pharmaceutical," *Spectroscopy and Spectral Analysis*, vol. 36, no. 9, pp. 2774–2778, 2016.

[23] C. Zeng, C. Ma, Z. Wang, and J. Ye, "Stacked autoencoder networks based speaker recognition," in *Proceedings of the 2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 294–299, Chengdu, China, July 2018.