Few-shot RF fingerprinting recognition for secure satellite remote sensing and image processing

Di Lin, Su Hu, Weiwei Wu, Gang Wu
Corresponding author: Su Hu (husu@uestc.edu.cn)
University of Electronic Science and Technology of China, Chengdu

APPENDIX A: SECURE SYSTEM MODEL

The secure SRS image processing system comprises a satellite and its earth stations, and the system architecture is shown in Fig. 1. Each earth station applies for satellite access by sending a pilot signal. The satellite must determine whether the earth station is authenticated through RF fingerprinting-based authentication methods.

If the earth station is authenticated, the satellite sends original SRS images to it, and the earth station recognizes the target objects from the image via a few image recognition algorithms. Otherwise, if the earth station is unauthenticated, the satellite must send an SRS image with perturbation to confuse this station. These earth stations cannot correctly recognize the targets from SRS images with perturbation, e.g., a combat airplane might be incorrectly identified as a passenger plane.

Although the development of RF fingerprinting recognition technology is becoming increasingly mature, its performance dramatically degrades when only a small number of training samples are available [1], [2]. When the training data is insufficient, the RF fingerprinting recognition model cannot achieve the necessary knowledge, leading to overfitting of this model [3], [4]. Existing few-shot recognition methods are mainly used in natural scenarios, e.g., image processing and natural language processing. The I/Q signals emitted by the radio communication devices are quite different from the images, so the existing few-shot algorithms for image recognition cannot be directly employed. This paper proposes a few-shot RF fingerprinting recognition algorithm based on a matching network model. In the following, we present the detailed structure of this model.

Matching network model. The main frame of a matching network is shown in Fig. 2, which consists of two parts: the embedding function module and the full-text embedding (FCE) module. Among them, the embedding function module contains the embedding function g for processing the support set, the embedding function f for processing the query set, and an attention kernel function a. The FCE module contains a bidirectional LSTM that adds a memory mechanism to the embedding function g and an LSTM that adds an attention mechanism to the embedding function f. Embedding functions mainly extract RF feature vectors of I/Q signal samples in the support and query sets. The attention kernel function calculates the weight between each sample in the query and support sets. The predicted probability of a query set sample is the weighted sum of the actual class labels in the support set. The FCE module is optional; in some cases, it can efficiently enhance the recognition accuracy of the model.

In the training flow of a matching network model, we divide the I/Q signal data into a support set and a query set, sending the data to the training model. We use the embedding function to calculate the respective feature vectors of the support set and the query set and then determine whether the FCE module is added. If it is added, the feature vector is embedded in the full text, and then the attention kernel function is used to calculate the weight between the samples in a query set and a support set. Finally, we compute the weighted sum of the labels in the support set and combine the weights to achieve the probability distribution of samples in the query set. We present the design of the embedding function, the attention kernel function, and the FCE module as follows.

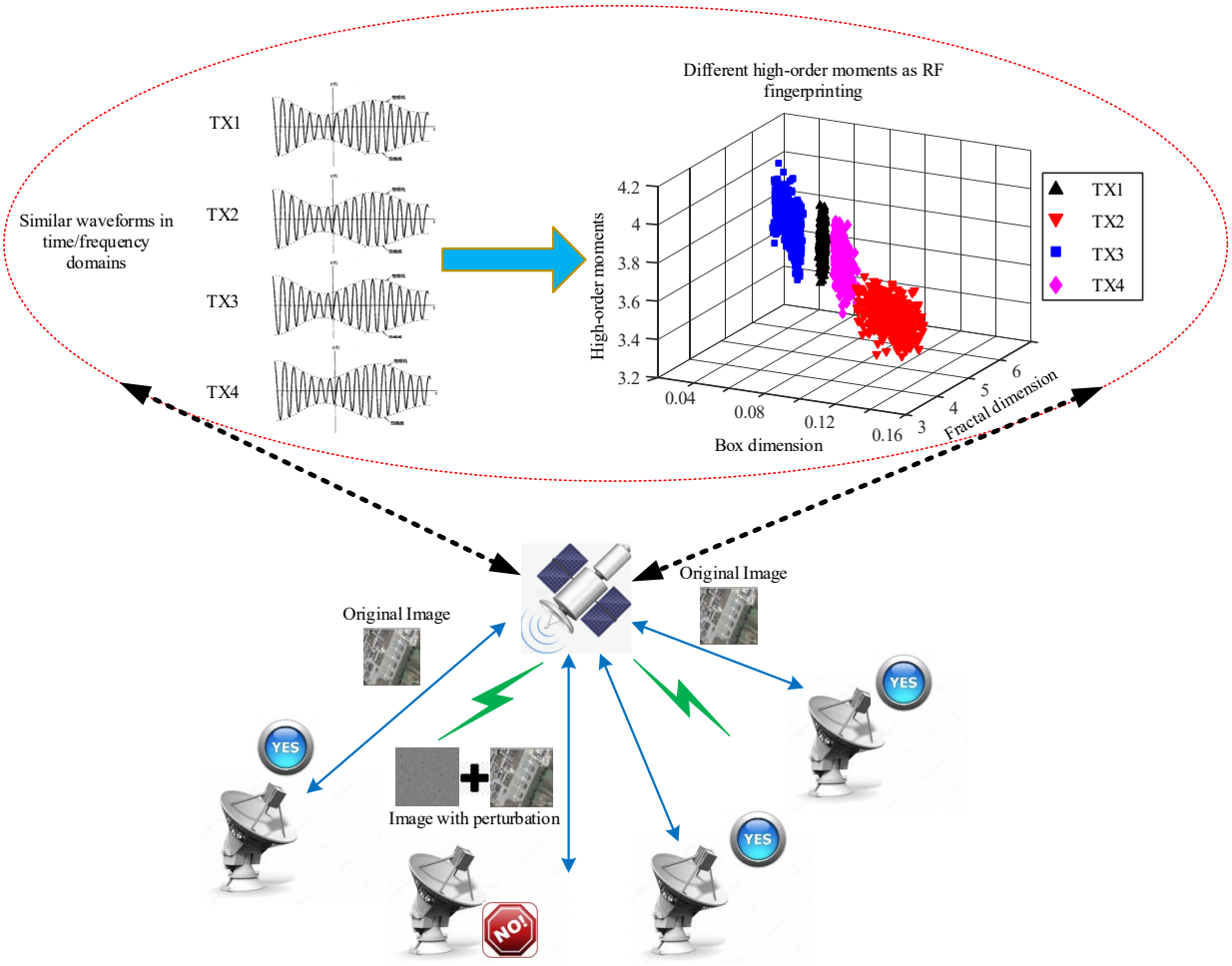

Fig. 1. Architecture of a secure SRS image processing system.

Design of embedding function. An embedding function is the basis of a matching network, which can be regarded as the RF fingerprint feature extractor of the I/Q signals. The feature vector trained by the embedding function represents the unique characteristics of the I/Q signals transmitted by each RF device. Therefore, the quality of the embedding function design determines the overall performance of recognition in the model.

To ensure that the embedding function can extract sufficient RF fingerprint features, we use four convolution operations to perform on the I/Q signal samples. The entire embedding function contains four convolution modules with the same structure, each consisting of a convolutional layer, a batch normalization, and a nonlinear activation function.

The convolution layer in each module consists of a convolution kernel, padding, and a stride as a filter, and the output is a feature matrix with 64 channels. Normalization and nonlinear activation operations are performed after the convolution operation. Normalization aims to prevent the gradient from disappearing and speed up the convergence, and the ReLU activation is to divide the feature space between different signals. The ReLU activation function (shown in equation (1)) retains its features when the sample x is greater than 0. Otherwise, its features are discarded.

$$ReLU(x) = \begin{cases} x, & x > 0, \\ 0, & x \leq 0. \end{cases} \tag{1}$$

Since there are only two channels of I/Q signals and the number of network parameters is small, we can ignore the pooling operation in the module at each layer to avoid losing useful information. After four convolutions, the flatten function is used

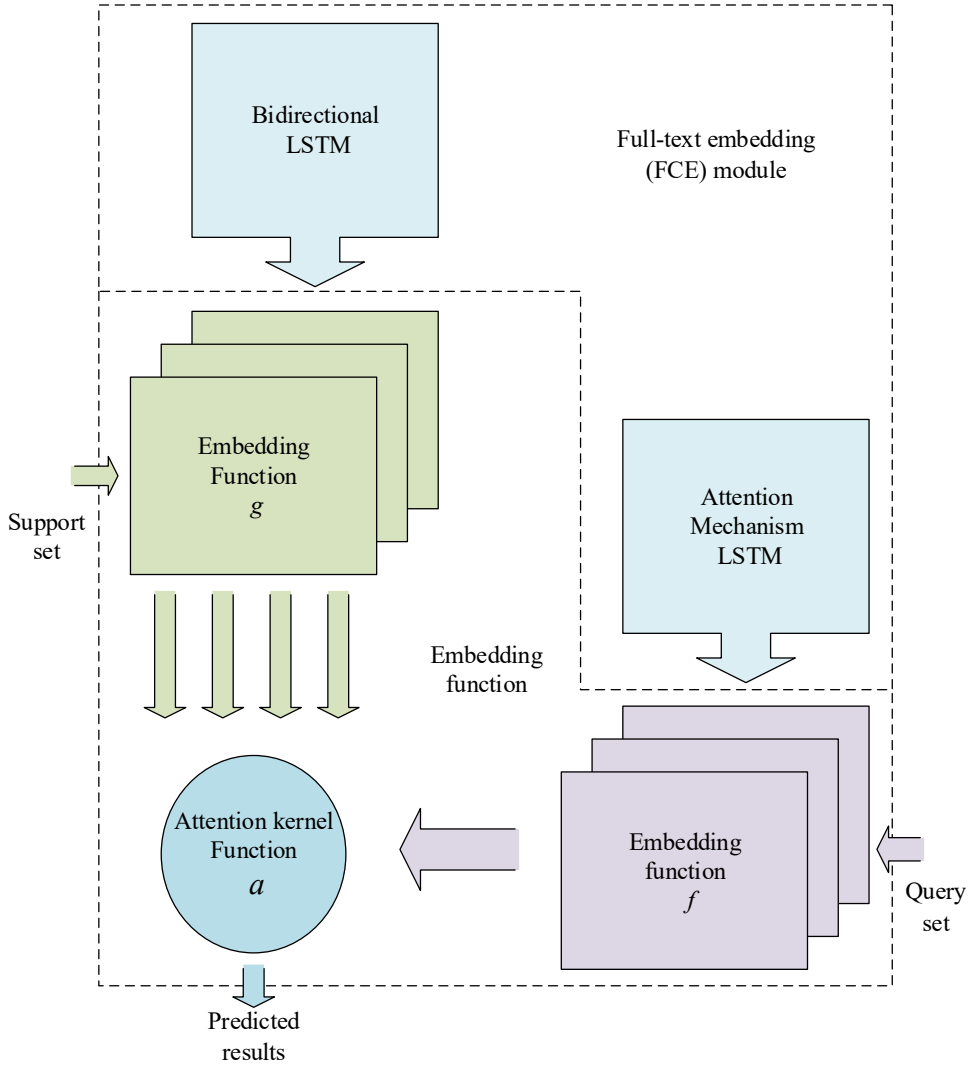

Fig. 2. Overall structure of the matching network model.

TABLE I
STRUCTURE OF THE EMBEDDING FUNCTION

Layer	Input size	Output size	Structure
conv1	(n, 16 × 16, 1)	(n, 16 × 16, 64)	Kernel size 3 × 3, stride 1, padding 1, batch normalization, nonlinear activation
conv2	(n, 16 × 16, 64)	(n, 16 × 16, 64)	Kernel size 3 × 3, stride 1, padding 1, batch normalization, nonlinear activation
conv3	(n, 16 × 16, 64)	(n, 16 × 16, 64)	Kernel size 3 × 3, stride 1, padding 1, batch normalization, nonlinear activation
conv4	(n, 16 × 16, 64)	(n, 16 × 16, 64)	Kernel size 3 × 3, stride 1, padding 1, batch normalization, nonlinear activation
Flatten	(n, 16 × 16, 64)	(n, 1 × 16384)	Flatten

to expand the feature vector extracted by the embedding function. The vector can be used as the input of the attention kernel function. Furthermore, since the general support set is in the same form as the query set, the embedding functions g and f employ the same structure.

Given n I/Q signal samples (including k categories) of size 2×128 as input, we set the number of channels as one and the size of each channel as 16×16. The initial input size is $(n, 16 \times 16, 1)$. Using the embedding function to extract the RF fingerprinting features, we can achieve n one-dimensional feature vectors. The specific structure is shown in Table 1.

Attention kernel. The attention kernel function calculates the similarity weight between the support and query sets' feature vectors. This paper uses the Euclidean distance normalized by a softmax function to calculate the similarity weight.

The sample data of I/Q signals can be represented as complex numbers. The real parts represent I-channel signals, and the

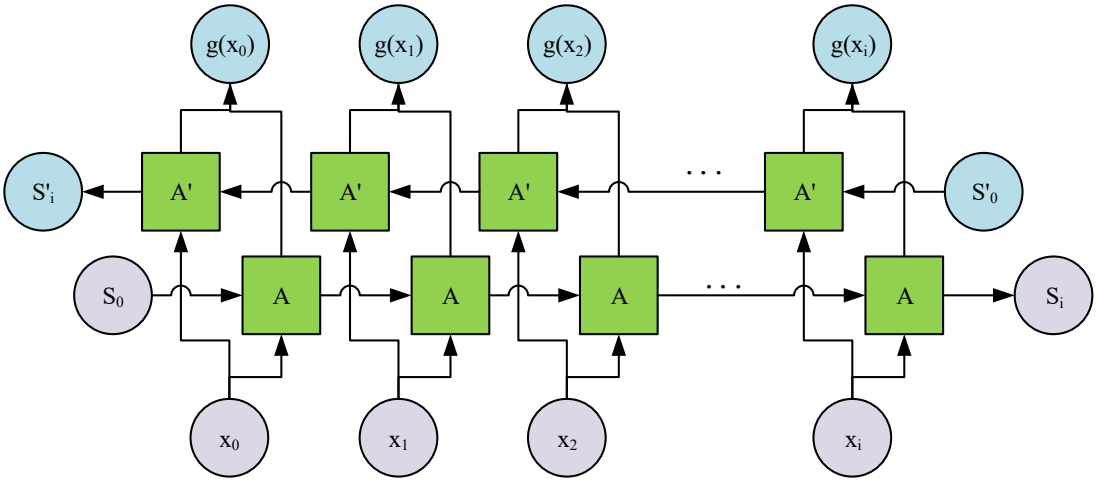

Fig. 3. The bidirectional LSTM structure.

imaginary represent Q-channel signals. The Euclidean distance can be represented as

$$L(f(x), g(\bar{x})) = \sqrt{(f(x_I) - g(\bar{x}_I))^2 + (f(x_Q) - g(\bar{x}_Q))^2}. \tag{2}$$

where x and $\bar{x}$ represent the data in the support and query sets, respectively. x_I and x_Q represent the I-channel and Q-channel signals, respectively.

The attention kernel function $A(f(x), g(\bar{x}))$ can be achieved by using softmax normalization on equation (2), shown as

$$A(f(x), g(\bar{x})) = e^{L(f(x), g(\bar{x}))} / \sum_{x, \bar{x}} e^{L(f(x), g(\bar{x}))}. \tag{3}$$

Therefore, the weighted sum calculation of the feature vector extracted by the embedding function and the attention kernel function represents the probability distribution of the samples in the support and query sets. Finally, we can obtain the classification labels in the query set samples to identify RF devices. In this work, we employ the cross-entropy function as the loss function of a matching network model, and this function can be shown as

$$Loss(x) = E\{\sum_{x} A(f(x), g(\bar{x}))\}. \tag{4}$$

where $E\{\cdot\}$ represents the expectation.

Full-Text embedding module. In this paper, we propose a full-text embedding (FCE) module which adopts the bidirectional LSTM network to strengthen the embedding functions g and f, respectively.

Unlike LSTM and RNN, bidirectional LSTM can combine context information to obtain context-related output results. Fig. 3 shows the bidirectional LSTM structure. Given the sample x in the support set, we can use the embedding function g to encode x as $g'(x)$. Afterward, we use the bidirectional LSTM model to obtain the forward output A and the backward output A'. Finally, we can attain the output vector $g(x)$ as

$$g(x) = g'(x) + A(g'(x)) + A'(g'(x)). \tag{5}$$

For the embedding function f of the query set samples, an LSTM network model with an attention mechanism is adopted. The function $g(x)$ obtained by equation (5) and the function $f'(x)$ by a CNN model are used as the input of this LSTM model with the attention mechanism, and the hidden feature h_k is calculated using the layer state h and unit state c. Specifically, we can achieve the state of final hidden layer h_k with a step size of k by the function of $LSTM(x, h, c)$ from equation (6). Also

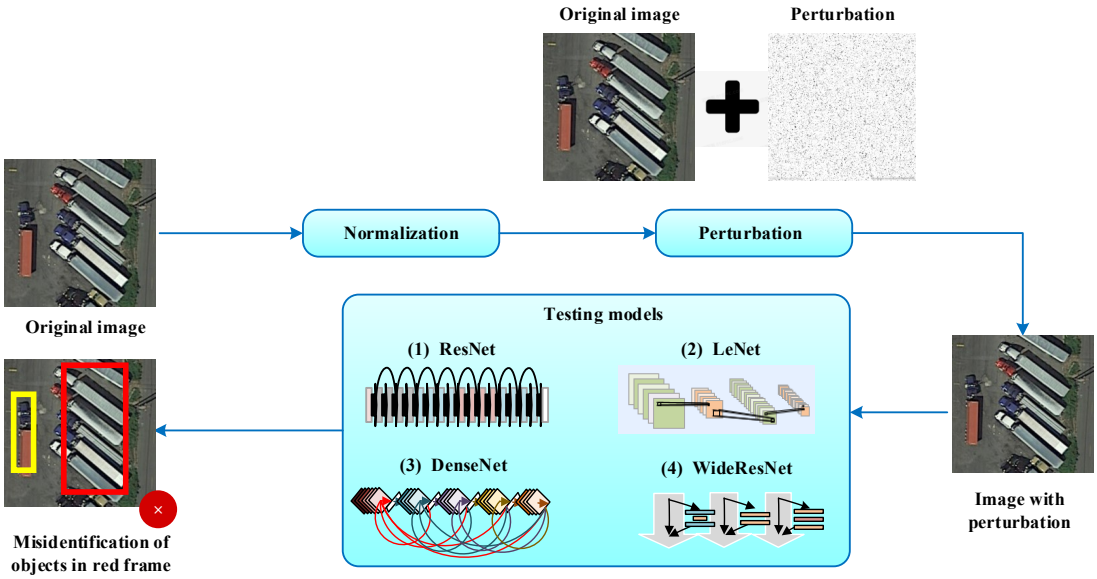

Fig. 4. Process of image perturbation.

we can calculate the attention kernel function with a bidirectional LSTM network A_{LSTM} as equation (7).

$$(h_k, c_k) = LSTM(f'(x), h_{k-1}, c_{k-1}) \tag{6}$$

$$A_{LSTM}(h_{k-1}, g(x)) = e^{h_{k-1}g(x)} / \sum_{x} e^{h_{k-1}g(x)} \tag{7}$$

Appendix B: Image perturbation with STA algorithm

In this section, we present the problem of image perturbation when a satellite user is identified as an unauthenticated user. The process of image perturbation is illustrated in Fig. 4. Q represents an actual SRS image before perturbation, and we normalize it to a range between -1 and $+1$ for simplicity. Also we set the labels of each pixel in the image as $Label = \{Label^1, Label^2, ..., Label^L\}$, in which $Label^i, i \in [1, L]$ denotes a one-hot vector and L represents the number of image pixels. $Z_{true} = \{Z^1_{true}, Z^2_{true}, ..., Z^L_{true}\}$ denotes the label of ground truth for the samples Q with L labels. Let C denote the classifier in the training process, i.e., $Z = C(Q)$ is in the same shape of Z_{true}. In the problem of image perturbation, we attempt to establish an adversarial sample Q_{adv} based on the input Q such that $C(Q_{adv}) = Z \neq Z_{true}$.

For SRS image recognition, we need to consider the rotation of objects in an image. Thus, a few off-the-shelf adversarial algorithms which modify the values of pixels in the original images may have low performance on SRS image recognition. Therefore, we slightly change the directions of objects in the image to achieve high robustness of SRS image recognition. Specifically, we propose a spatially transformed adversarial (STA) algorithm for a high level of robustness in image recognition. In the following, we present two primary concepts in the STA algorithm.

Let F denote the pixel difference between the original image Q and the adversarial sample Q_{adv} (i.e. the image Q with perturbation). If $Q^{(j)}_{adv}$ represents the jth pixel of Q_{adv}, $(x^{(j)}_{adv}, y^{(j)}_{adv})$ denotes the coordinate point of $Q^{(j)}_{adv}$. Let $F_j = (\triangle x^{(j)}, \triangle y^{(j)})$ be the difference between Q and Q_{adv} at their jth pixel point. We can calculate the coordinate point of $Q^{(j)}$ as $(x^{(j)}, y^{(j)}) = (x^{(j)}_{adv} + \triangle x^{(j)}, y^{(j)}_{adv} + \triangle y^{(j)})$. We employ a differentiable bilinear interpolation [9] to represent the values of pixels in the image as

$$Q_{adv}^{(j)} = \sum_{i \in S(x^{(j)}, y^{(j)})} Q^{(i)}(1 - \left|x^{(j)} - x^{(i)}\right|)(1 - \left|y^{(j)} - y^{(i)}\right|) \tag{8}$$

where $S(x^{(j)}, y^{(j)})$ denotes the neighborhood of $(x^{(j)}, y^{(j)})$, including top-left, top-right, bottom-left, bottom-right pixels. With the function of F, we can modify the direction of the objects in the images, thereby improving the robustness of SRS image recognition.

To achieve the function of F, we can denote the objective function as

$$F^* = \arg\min_F \mathcal{G}_{adv}(Q, F) + \epsilon \mathcal{G}_{smo}(F) \tag{9}$$

where $\mathcal{G}_{adv}(Q, F)$ denotes the loss of object misclassification. $\mathcal{L}_{smo}(f)$ indicates the loss of the smoothness of spatial changes. ϵ denotes the balance between two losses. We present two types of losses as

$$\mathcal{G}_{adv}(Q, F) = \max_{i \neq j} C(Q_{adv})_i - C(Q_{adv})_j \tag{10}$$

where $C(p)$ represents the classification of p.

$$\mathcal{G}_{smo}(F) = \sum_{i}^{all\ pixels} \sum_{j \in S(i)} \sqrt{\left\|\triangle x^{(i)} - \triangle x^{(j)}\right\|_2^2 + \left\|\triangle y^{(i)} - \triangle y^{(j)}\right\|_2^2} \tag{11}$$

where $\mathcal{G}_{smo}(F)$ represents the total difference of coordinate points around a pixel. The minimum loss in equation (11) can guarantee the similar difference and direction of the points around the pixel point i to smooth the spatial changes.

APPENDIX C: SRS IMAGE RECOGNITION

In this section, we present the problem of image recognition when a satellite user is identified as an authenticated user. He et al. in [5] propose a multitask gradient adversarial learning method, which is designed for the detection and reidentification of Satellite videos and can dramatically improve the accuracy of object tracking. Also He et al. in [6] propose a a lightweight network (DABNet) for the detection of images with high accuracy, and this network can also speed up the process of detection. Sun et al. In [7] propose a dataset (i.e. FAIR1M) by collecting over 40000 images for object recognItion in remote sensing scenarios. The FAIR1M dataset can represent objects in the real world well for remote-sensing applications. In this paper, we employ the DOTA SRS image dataset released by Wuhan University [8]. The DOTA dataset contains 180000 instances of 2806 images in 15 categories. Instead of processing these images directly, we need to preprocess them since the images in the dataset have a large size of between 800 $\times$ 800 and 4000 $\times$ 4000. We need to resize these images by dividing them into a few segments. Each segment has two primary parameters: segmentation size and the gap between segmentations to avoid these segments losing information in the preprocessing. Specifically, w e s et t he s egmentation s ize a s 1 024 a nd t he segmentation gap as 200. For an image at a size above 1024, we can divide this image into a few segmentations with a gap of 200, while we can pad the pixels to increase the size of an image to 1024 when the size of the original image is below 1024. After the preprocessing, we employ an attention mechanism-based rotating object detection model to recognize the SRS images. This model includes a convolutional attention mechanism module, a spatial pyramid pooling module, and an angle prediction module.

Convolutional neural networks. Convolutional neural networks (CNNs) have dramatically developed in object detection. A CNN algorithm uses convolution, activation, pooling, and other operations to extract features from an image [9], [10]. We can remove the useless background information of SRS images, which takes an unnecessarily long time in model training.

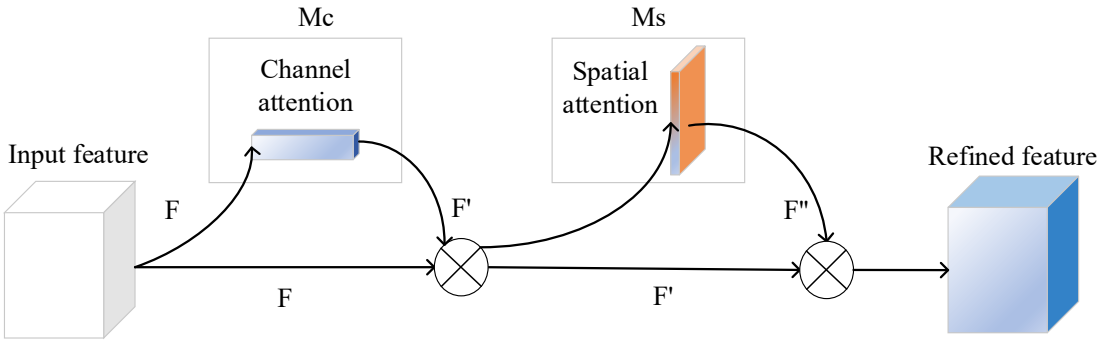

Fig. 5. Structure diagram of a CBAM model.

Such a problem is particularly serious in SRS image recognition. In SRS image recognition, an image usually contains a large amount of useless background information, and the proportion of useful data on target may be less than 1/10 [11]. Therefore, by introducing the attention mechanism, we propose a novel convolutional block attention module (CBAM) for image recognition.

Considering the rotation of targets in the SRS images, we propose a novel CBAM model appropriate for rotating target detection. The structure diagram of our CBAM model is illustrated in Fig. 5. A CBAM model is composed of two independent modules, including the channel attention module and the spatial attention module. The spatial attention module captures spatial attention information, and the channel attention is responsible for achieving the attention on the channel.

The mathematical representation of CBAM is shown in equations (12) and (13), where F represents the input feature map, M_c represents the operation of adding channel attention to F, and it can generate a one-dimensional channel attention weight. $\otimes$ represents the feature multiplication operation. It multiplies the channel attention weight with F to achieve the new feature F', which can increase the channel attention mechanism. M_s is the operation of adding spatial attention to F'. It can generate a two-dimensional spatial attention weight and multiplies the spatial attention weight map with F' to obtain the final feature map F''. The specific implementation of channel attention and spatial attention is as follows.

$$F' = M_c(F) \otimes F \tag{12}$$

$$F'' = M_s(F') \otimes F' \tag{13}$$

Channel attention is responsible for understanding which input features are expected to extract [12]. The channel attention can assign weights to feature maps by generating masks and scoring channels. The mathematical representation of channel attention is shown as

$$M_c(F) = \sigma(W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max}))) \tag{14}$$

where F^c_{max} and F^c_{avg} represent the maximum pooling and mean pooling on the feature map F, respectively. W_0 and W_1 are the weights of a two-layer neural network: multi-layer perception (MLP). σ represents the parameter of ReLU in the neural network. $M_c(F)$ represents the output of channel attention.

The detailed process of channel attention is as follows: firstly, the maximum pooling and mean pooling are performed on the feature map F, and the feature maps F^c_{max} and F^c_{avg} are obtained, respectively. Both feature maps are at a size of $1 \times 1 \times c$, and c denotes the number of channels in the feature map. Then, the pooled features are sent to a neural network with multi-layer perception (MLP). The number of neurons in each layer is c, and a ReLU is used as the activation function. Finally, element-wise and sigmoid activation operations are performed to achieve the channel attention feature M_c.

Spatial attention is responsible for understanding which pixels to focus on. By learning spatial features, the spatial attention model can attain the weight of each position and then assign the weight to feature maps. The mathematical representation of spatial attention is shown as

$$M_s(F) = \sigma(G(F^c_{max}, F^c_{avg})) \tag{15}$$

where F^c_{max} and F^c_{avg} represent the maximum pooling and mean pooling on the feature map F, respectively. $G(x)$ represents a convolutional layer to extract the features. $M_s(F)$ represents the output of spatial attention.

The detailed process of spatial attention is as follows: Firstly, a two-dimensional feature map is used to calculate the mean pooling and maximum pooling and then combine the features. To adapt the receptive field of the model to the SRS images, we design a convolutional layer to extract the features in the spatial dimension. We then activate a sigmoid function to attain the spatial attention feature $M_s(F)$.

Spatial pyramid pooling module. In a neural network, the feature dimension of a convolutional layer is strictly restricted by the neuron number at the fully connected layer [13]. The convolutional layer needs to match the feature dimension of the fully connected layer to perform classification tasks. Therefore, there is a restriction requirement on the size of the model input. We introduce the spatial pyramid pooling (SPP) method to meet the need for model input.

The SPP module can convert a model input at any size into an expected size. SPP is completely independent of the model's design and can be easily embedded in the model. Therefore, when designing the model, it is necessary to replace an embedded SPP layer with the pooling layer to achieve the feature map at a uniform size.

The SPP module comprises two convolution operations, three max-pooling operations, and a concatenation operation. We perform the maximum pooling on a feature map, and then conduct the concatenation and convolution operations on the pooled and convolutional features, which can strengthen the fusion of features. Specifically, the output contains the features at multiple scales, including both small-scale local and large-scaled global features. The fusion of features can enhance the accuracy of our model.

Angle prediction module. The angle prediction is the key point in the rotating target detection algorithm. A suitable angle prediction method can not only complete the representation of the angle more accurately but also improve the accuracy of the rotating target detection model [14]. This paper employs the circular smooth label (CSL) to represent the angle in the model training process.

In the angle prediction module, we divide a range of 180 degrees into 180 categories, each corresponding to 1 degree. Then, we can predict the target angle by classifying its angle into one sort. This paper employs a CSL method with a Gaussian function as the window function. During training, we must convert an angle into a Gaussian label. Specifically, we use a one-dimensional Gaussian function to represent the angled label of each target, shown in equation (16).

$$f(x) = Ae^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{16}$$

where x is the actual angle of a target, σ is the radius of the Gaussian function window, A and μ represent the peak and the center offset of a Gaussian function.

Among these parameters, the radius of window σ has a great influence on the model's ability to recognize angles. The conversion process from angles to Gaussian labels includes calculating the Gaussian values of all the angles from -90 to 90 degrees, offsetting the labels according to the actual angles of targets, and shifting the peak of the Gaussian function to the subscript of the angles of targets.

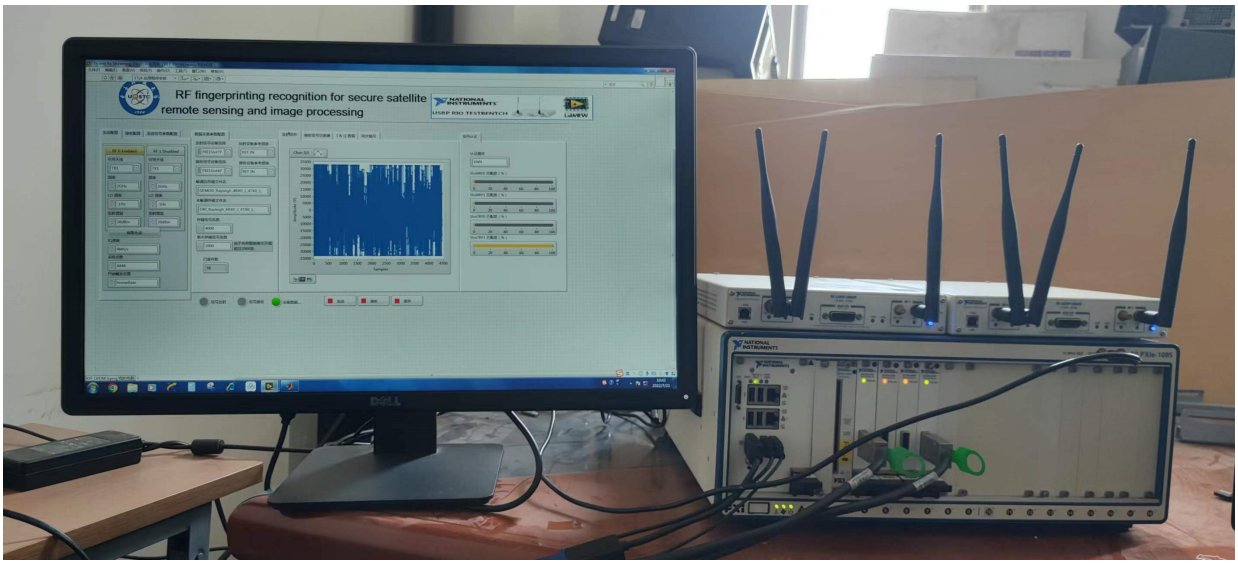

Fig. 6. USRP experiment hardware platform.

TABLE II
CHANNEL PARAMETERS

Setting	RDS (ns)	Path number
Indoor	60	15
Outdoor	90	15
Setting	Data rate (Mbps)	Guard interval (μs)
Indoor	15	0.7
Outdoor	15	0.7

APPENDIX D: EXPERIMENT RESULTS AND CONCLUSIONS

For RF fingerprinting recognition, we experiment by collecting real RF data with Universal Software Radio Peripherals (USRP), shown in Fig. 6. Specifically, we establish a hardware platform with a NI-PXIe 1085 device, and three USRP-RIO-2943 devices [15]. Two USRP-RIO-2943 devices with four antennas are used to simulate four satellite earth stations, and the other USRP-RIO-2943 device mimics the satellite. The NI-PXIe 1085 device is responsible for data storage and importing the data to Matlab to add noise and RF fingerprinting analysis. In total, we collect 100000 records of I/Q signals with USRP devices for RF fingerprinting authentication. We randomly select 70000 records as the training data, and 30000 records as the testing data.

For image recognition, we collect the SRS images from the dataset of DOTA. These SRS images are originally from different data sources, including Google earth, JL-1 satellite, and GF-2 satellite [8]. The dataset consists of 2806 SRS images in total, and each image has a pixel size ranging from 800×800 to 4000×4000. We randomly select 1964 images for data training and 842 images for data testing. These DOTA images contain various objects at different scales, orientations, and shapes, and they are annotated by experts using 15 common object classes, including ground runways, roundabouts intersections, and basketball court, etc [8].

Accuracy of identifying authenticated users. In the following, we investigate the accuracy of our proposed few-shot RF-fingerprinting authentication algorithm with a CNN-based learning algorithm, which is the most widely used algorithm for RF fingerprinting authentication [16]. In the simulation, we simulate 1000 RF signals transmitted by each simulated satellite earth station and also load the data into Matlab to simulate wireless channels. Specifically, we consider the Rayleigh channels at home and in the office, and the primary channel parameters are shown in Table 2 [1].

Fig. 7 illustrates the accuracy of various authentication algorithms under different signal-to-noise (SNR) ratio levels. The

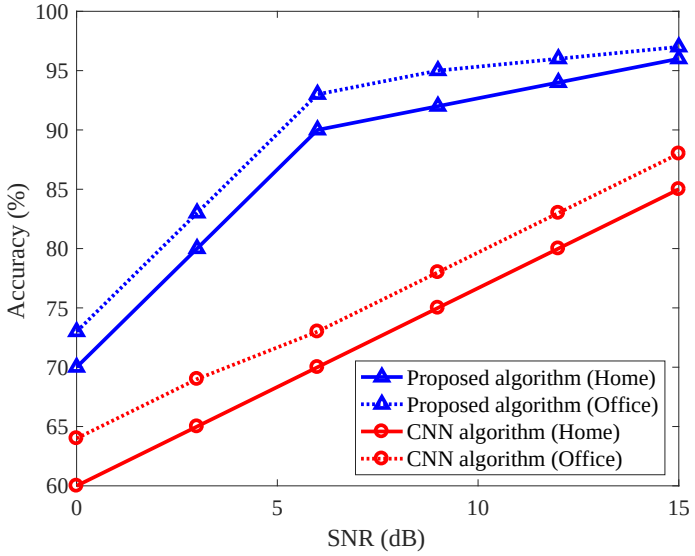

Fig. 7. Accuracy of RF-fingerprinting authentication algorithms.

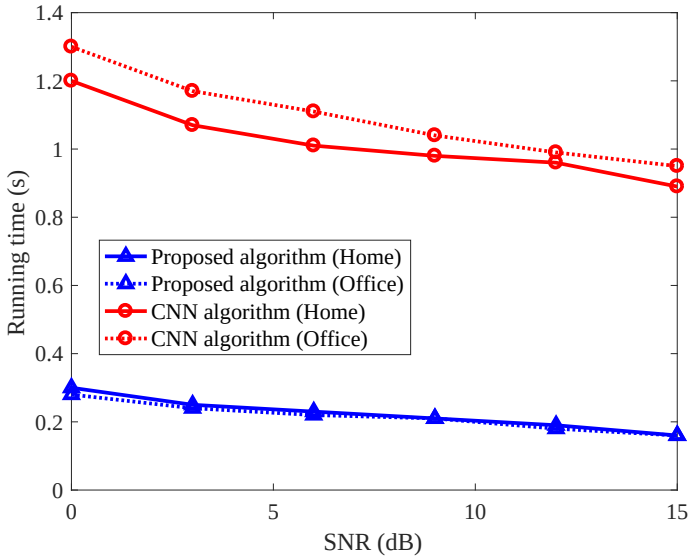

Fig. 8. Running time of RF-fingerprinting authentication algorithms.

results show that the proposed matching network-based few-shot RF-fingerprinting authentication algorithm can achieve a $10\% - 15\%$ higher accuracy than the CNN-based learning algorithm. The accuracy of the proposed algorithm can reach 96% at the SNR of 15dB, while the CNN learning algorithm can only attain an accuracy of 85%.

Also, our proposed RF fingerprinting authentication method is designed with a matching network. Considering the low-dimension characteristics of I/Q signals compared to images, we remove the pooling operations at each layer to speed up the RF fingerprinting authentication process. As shown in Fig. 8, the proposed authentication method can achieve a shorter running time than its benchmark, i.e. the CNN learning algorithm.

Ability of confusing unauthenticated users via image perturbation. To compare the ability of different image perturbation algorithms [17], we consider four image perturbation algorithms: fast gradient sign method (FGSM), basic iterative methods (BIM), DeepFool, and Jacobian-based saliency map attack (JSMA) as the benchmark algorithms of our proposed STA algorithm, and the user confusion capabilities (i.e., the decrease of recognition accuracy) of the above image perturbation algorithms are compared. Also, we consider four image recognition models, e.g., LeNet, ResNet, DenseNet, and Wide ResNet, to evaluate

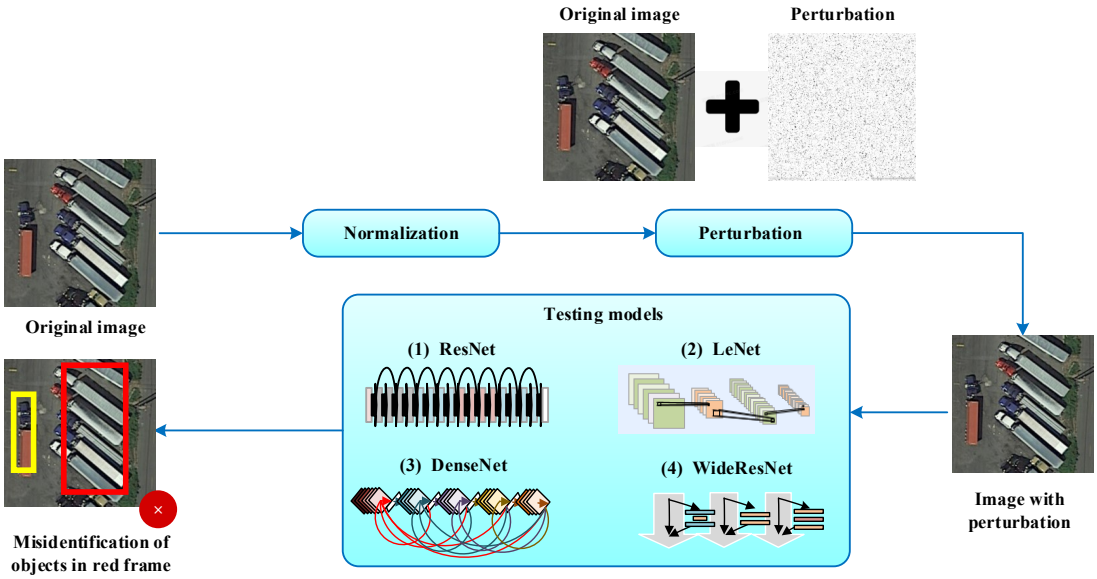

Fig. 9. Confusion ability of image perturbation algorithms.

TABLE III
PERFORMANCE OF IMAGE PERTURBATION

Recognition algorithm	Original	FGSM	BIM	DeepFool	JSMA	STA (Proposed)
LeNet	87.3%	32.2%	34.4%	44.2%	34.2%	22.2%
ResNet	90.2%	34.3%	35.5%	54.8%	44.2%	23.1%
DenseNet	91.1%	38.1%	45.3%	67.2%	49.2%	24.2%
Wide ResNet	91.1%	38.1%	59.2%	69.4%	54.2%	24.2%

the change in recognition accuracy. In the experiment, we randomly selected 900 and 200 SRS images as the training and test validation sets, respectively, from the DOTA dataset. The experiment completes image recognition tasks based on this dataset.

As shown in Table 3, when there is no image perturbation, the accuracy of image recognition by LeNet, ResNet, DenseNet, and Wide ResNet is 87.3%, 90.2%, 91.1%, 91.1%, respectively. As the benchmark algorithm of the proposed STA algorithm, the FGSM algorithm can reduce image recognition accuracy to between $30\%-40\%$. BIM, DeepFool, and the JSMA algorithm can reduce the accuracy to $30\% - 70\%$. The proposed STA algorithm can reduce the image recognition accuracy below 20%. Therefore, the STA algorithm can outperform the other algorithms in confusing unauthenticated users, i.e., reducing the image recognition accuracy.

Accuracy of image recognition for authenticated users. In the following, we discuss the accuracy of image recognition by the proposed CBAM algorithm and by various benchmark algorithms: the small cluttered rotated detection (SCRDet) algorithm, the refined single-stage detector (R3Det) algorithm, the YOLOv5 algorithm, the ROT-YOLOv5 algorithm which are the most widely used algorithms for rotating target detection [18].

Fig. 10 shows the accuracy of image recognition through various learning algorithms. As the benchmarks of the proposed CBAM algorithm, the SCRDet algorithm, the R3Det algorithm, the YOLOv5 algorithm, and the ROT-YOLOv5 algorithm can achieve an accuracy of 69%, 70%, 65%, 68%, respectively. Our proposed CBAM algorithm can achieve an accuracy of 77%. The recognition effect of our proposed CBAM algorithm is shown in Fig. 11.

Conclusions. This paper has proposed a secure SRS image processing system with RF fingerprinting for user authentication. Considering the small amount of RF fingerprinting data, we offer a few-shot RF fingerprinting recognition algorithm based on a matching network. Also, we propose an image perturbation method with STA algorithm to confuse unauthenticated users to reduce the accuracy of recognizing SRS images. For authenticated users, we present an image recognition algorithm

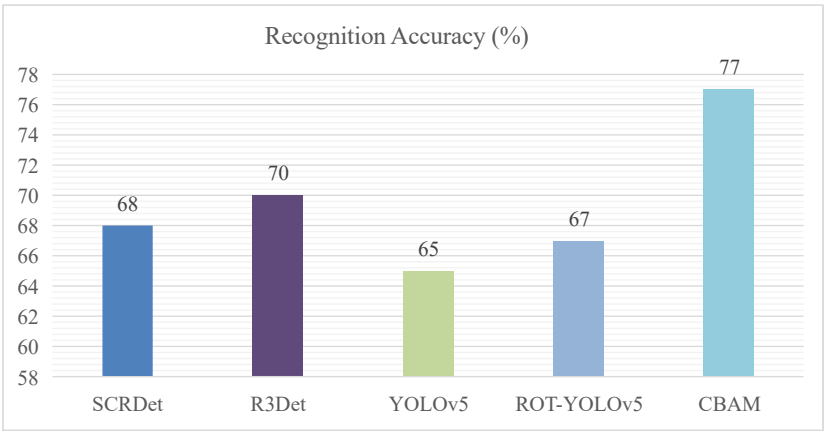

Fig. 10. Accuracy of image recognition for authenticated users.

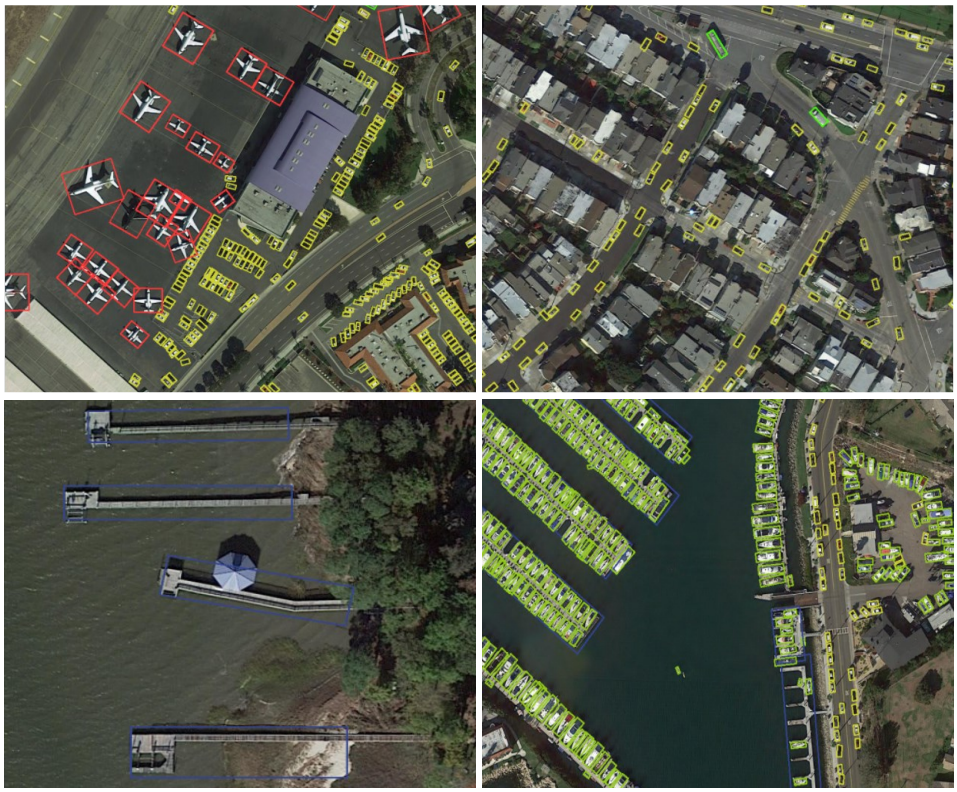

Fig. 11. Recognition effect of our proposed CBAM algorithm.

CBAM designed to detect rotating targets. By comparing with a few benchmark algorithms, the simulation results show that our matching network-based algorithm can outperform the other RF fingerprinting recognition algorithms when only a small amount of RF data is available, e.g., in the scenario of remote sensing. Also, our proposed STA algorithm has a higher performance in confusing unauthenticated users than its benchmark algorithms since the former can dramatically reduce image recognition accuracy at the unauthenticated user end. Our proposed CBAM algorithm for image recognition outperforms its benchmarks, including the SCRDet algorithm, the R3Det algorithm, the YOLOv5 algorithm, and the ROT-YOLOv5 algorithm, which are the most widely used algorithms for target detection.

REFERENCES

[1] W. Wu, S. Hu, D. Lin, G. Wu, "Reliable resource allocation with RF fingerprinting authentication in secure IoT networks." *Sci. China Inf. Sci.*, 2022, 65: 170304.

[2] Y. Xu, J. Tang, B. Li, N. Zhao, D. Niyato and K. -K. Wong, "Adaptive Aggregate Transmission for Device-to-Multi-Device Aided Cooperative NOMA Networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 4, pp. 1355-1370, April 2022, doi: 10.1109/JSAC.2022.3143267.

[3] J. Li, Y. F. Li, L. He, et al. "Spatio-temporal fusion for remote sensing data: an overview and new benchmark," *Sci. China Inf. Sci.*, vol. 63, no. 4, pp.140301, 2020. doi: 10.1007/s11432-019-2785-y.

[4] N. Li, S. D. Xia, X. F. Tao, et al. "An area based physical layer authentication framework to detect spoofing attacks," *Sci. China Inf. Sci.*, vol. 63, no. 10, pp.202302, 2020. doi: 10.1007/s11432-019-2802-x.

[5] Q. He, X. Sun, Z. Yan, B. Li and K. Fu, "Multi-Object Tracking in Satellite Videos With Graph-Based Multitask Modeling," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-13, 2022, Art no. 5619513, doi: 10.1109/TGRS.2022.3152250.

[6] Q. He, X. Sun, Z. Yan and K. Fu, "DABNet: Deformable Contextual and Boundary-Weighted Network for Cloud Detection in Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-16, 2022, Art no. 5601216, doi: 10.1109/TGRS.2020.3045474.

[7] Xian Sun et al., "FAIR1M: A Benchmark Dataset for Fine-grained Object Recognition in High-Resolution Remote Sensing Imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 184, pp. 116-130, 2022.

[8] G.S.Xia et al., "DOTA: A Large-Scale Dataset for Object Detection in Aerial Images," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, 3974-3983, doi: 10.1109/CVPR.2018.00418.

[9] Y. Wu, Z. Fan, Y. Fang and C. Liu, "An Effective Correction Method for AFM Image Distortion due to Hysteresis and Thermal Drift," *IEEE Transactions on Instrumentation and Measurement*, 2021, 70(5004212): 1-12, doi: 10.1109/TIM.2020.3038007.

[10] S. Shen, K. Zhang, Y. Zhou, et al., "Security in edge-assisted Internet of Things: challenges and solutions." *Sci. China Inf. Sci.*, 2020, 63(12): 220302.

[11] Li, S., Zhai, D., Du, P. et al. "Energy-efficient task offloading, load balancing, and resource allocation in mobile edge computing enabled IoT networks." *Sci. China Inf. Sci.*, 2019, 62: 29307.

[12] You, X., Wang, CX., Huang, J. et al. "Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts." *Sci. China Inf. Sci.*, 2021, 64: 110301.

[13] G. Kakkavas, K. Tsitseklis, V. Karyotis and S. Papavassiliou, "A Software Defined Radio Cross-Layer Resource Allocation Approach for Cognitive Radio Networks: From Theory to Practice," *IEEE Trans. on Cognitive Commun. and Networking*, 2020, 6(2): 740-755.

[14] J. Cui, L. Wei, J. Zhang, Y. Xu and H. Zhong, "An Efficient Message-Authentication Scheme Based on Edge Computing for Vehicular Ad Hoc Networks," *IEEE Trans. on Intelligent Transportation Systems*, 2019, 20(5): 1621-1632.

[15] X. Tian, X. Wu, H. Li and X. Wang, "RF Fingerprints Prediction for Cellular Network Positioning: A Subspace Identification Approach," *IEEE Transactions on Mobile Computing*, 2020, 19(2): 450-465, doi: 10.1109/TMC.2019.2893278.

[16] W. Wu, S. Hu, D. Lin and Z. Liu, "DSLN: Securing Internet of Things Through RF Fingerprint Recognition in Low-SNR Settings," *IEEE Internet of Things Journal*, 2022, 9(5): 3838-3849, doi: 10.1109/JIOT.2021.3100398.

[17] J. Kim, S. Kim, S. T. Kim and Y. M. Ro, "Robust Perturbation for Visual Explanation: Cross-Checking Mask Optimization to Avoid Class Distortion," *IEEE Transactions on Image Processing*, 2022, 31: 301-313, doi: 10.1109/TIP.2021.3130526.

[18] L. Peng, J. Zhang, M. Liu and A. Hu, "Deep Learning Based RF Fingerprint Identification Using Differential Constellation Trace Figure," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1091-1095, Jan 2020.