EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Intelligent radar HRRP target recognition based on CNN-BERT model

Penghui Wang[1*], Ting Chen[1], Jun Ding[1], Mian Pan[2*] and Sanding Tang[2]

*Correspondence: wangpenghui@mail.xidian.edu.cn; ai@hdu.edu.cn

[1] National Laboratory of Radar Signal Processing, Xidian University, Xi'an 710071, China
[2] School of Electronics and Information, Hangzhou Dianzi University, Hangzhou 310018, China

Abstract

Stable and reliable feature extraction is crucial for radar high-resolution range profile (HRRP) target recognition. Owing to the complex structure of HRRP data, existing feature extraction methods fail to achieve satisfactory performance. This study proposes a new deep learning model named convolutional neural network–bidirectional encoder representations from transformers (CNN-BERT), using the spatio–temporal structure embedded in HRRP for target recognition. The convolutional token embedding module characterizes the local spatial structure of the target and generates the sequence features by token embedding. The BERT module captures the long-term temporal dependence among range cells within HRRP through the multi-head self-attention mechanism. Furthermore, a novel cost function that simultaneously considers the recognition and rejection ability is designed. Extensive experiments on measured HRRP data reveal the superior performance of the proposed model.

Keywords: High-resolution range profile (HRRP), Convolutional neural network (CNN), Bidirectional encoder representations from transformers (BERT), Attention mechanism, Intelligent target recognition

1 Introduction

The wideband radar has a high range resolution, and its echo is called the target's high-resolution range profile (HRRP). HRRP has the advantages of easy acquisition and processing and contains rich target structure information such as radial dimension, scatterer's distribution, and echo intensity. Therefore, HRRP-based radar automatic target recognition (RATR) is becoming a research hotspot in intelligent radar signal processing [1–4].

Several methods have been proposed in recent years to extract the spatial and temporal features of HRRP. These methods can be roughly divided into two classes. (1) Considering HRRPs as random points scattered in high-dimensional space, HRRPs are assumed to follow specific statistical distributions when extracting the corresponding spatial structural features. For example, the HRRPs are assumed to follow Gaussian distribution in [5–7], and the adaptive Gaussian classifier, Gaussian mixture model (GMM), and support vector machine (SVM) are used to classify unknown radar targets, respectively; in [11], the HRRPs are modeled using the Gamma mixture model to describe their statistical characteristics accurately; in [12], the subspace

structure of HRRP is studied, while in [13], the multi-subspace structure of HRRP is exploited. (2) Viewing HRRP as a one-dimensional temporal sequence along the range dimension, sequential modeling of HRRP was conducted to extract the implicit evolution structures as features. For example, in [8, 9], the hidden Markov models (HMMs) are employed in HRRP target recognition; Pan et al. characterize the spectrogram feature extracted from HRRP via the TSB-HMM model, in which multi-aspect frames of one target are learned jointly [10]. In [14], the temporal factor analysis model is used. Though these methods achieved acceptable results in relatively simple tasks, their structural simplicity limits their description ability and performance in complex scenarios.

Owing to their excellent nonlinear feature extraction ability, deep learning models have gradually become the mainstream method in HRRP recognition. Deep learning-based methods focusing on spatial or temporal structures have recently been applied to the radar HRRP recognition field. The convolutional neural network (CNN) models extract the local spatial structure features from the HRRP envelope using the convolution operation [22]. Wan et al. used CNN for extracting multi-resolution spectrogram features of HRRP and then weighed them using an attention mechanism [15]. In [32], Chen et al. integrated the target recognition and rejection tasks using CNN by adding a deconvolution decoder. However, CNN models find it challenging to describe HRRP's global structure due to the limited receptive field. Furthermore, the failure to capture the time-series correlation across the HRRP range cells causes a loss of valuable information reflecting the target physical structure characteristics.

Deep learning recurrent neural network (RNN) models have demonstrated excellent sequence modeling ability in natural language processing, machine translation, and speech recognition. In RATR field, RNN is used to extract the long-range temporal structure embedded in the HRRP sequence. In [36], Xu et al. proposed a new attention-based RNN model to reveal the structural correlation inside the target. In [35], Li et al. proposed a bidirectional simple recurrent unit network (SAMBi-SRU) to extract robust features effectively from HRRP with good noise immunity. However, the dimension and length of input sequences in RNN models are coupled and cannot be adjusted independently, reducing the model's flexibility. Furthermore, the long-range dependence will be severely weakened as the sequence length grows, thereby limiting the application of deep models in HRRP target recognition.

This study overcomes the challenges of CNN and RNN-based models by characterizing the spatial structure of the HRRP envelope and temporal dependence across range cells simultaneously. Specifically, we developed a novel deep model named convolutional neural network–bidirectional encoder representations from transformers (CNN-BERT) for HRRP feature extraction. It comprises a convolutional token embedding module and BERT module and adjusts the feature importance at the backend using an attention mechanism. The main characteristics of the proposed model are summarized below:

1. The convolutional token embedding module finely describes the HRRP's local spatial structure and generates the sequence features. It considerably improves the proposed model's early expression capability and the efficiency and flexibility of the HRRP modeling task.

2. The BERT module models the long-range temporal dependence in HRRP. To the best of our knowledge, this is the first attempt to introduce a BERT-based model into the RATR field. The multi-head self-attention mechanism in the BERT module perfectly describes the dependency relationship between two range cells at any position and captures one-step local and global dependencies. Additionally, the BERT module has good parallelism.
3. In designing the cost function, recognition and rejection abilities are simultaneously considered. Furthermore, their roles can be adjusted according to the application scenario.
4. Experimental results and attention map visualization based on the measured data verify the effectiveness of the proposed method.

The remaining article is organized as follows. We analyze the principles and long-range feature capture capabilities of related deep models in Sect. 2. The proposed model is introduced in Sect. 3. Section 4 details the training and testing process. Section 5 presents the performance analysis based on various experiments. Finally, the conclusions are offered in Sect. 6.

2 Analysis of related deep learning models

Currently, deep neural networks are being used for HRRP recognition and have achieved good recognition performance. In particular, RNN and CNN can effectively describe the interdependence among range cells and are widely used. This section mainly analyzes the principles, structures, and long-range feature capture capabilities of CNN and RNN models to understand their ability to utilize the time-series information within HRRP range cells.

2.1 RNN model

RNN model is widely used for sequential data representation. However, the original RNN has short-term memory capability and suffers from gradient explosion and vanishing problems with relatively long input sequences [16]. The long short-term memory network (LSTM) based on the gated RNN was proposed to improve long-term memory ability [17–19]. A schematic of the LSTM structure, comprising the input, hidden, and output layers, is shown in Fig. 1. A linear sequence is formed between the hidden layer nodes, which propagates the extracted information from front to back in chronological order.

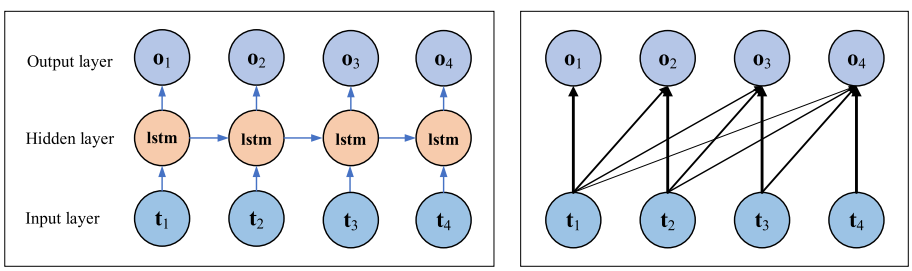

Fig. 1 Schematic of the LSTM. Left: LSTM structure; right: input sequence and output feature dependency

LSTM controls the importance weights of current inputs and historical information through input and forgetting gates. When important information appears in the input of the current moment, the input gate value is close to 1, while that of the forgetting gate is close to 0, and the historical information will be forgotten. The older the information, the higher the degree of forgetting. According to the principle of the gating-based RNN, we plot the dependency between the input sequences and output features in Fig. 1. The connecting lines indicate a dependency between the output and input, while the thickness indicates the dependency strength. For example, the output feature o_3 mainly depends on the current input t_3 and past inputs t_3 and t_2. The strongest dependency relationship is with the input at the current time t_3 and gradually decreases with the backward movement of time.

However, for HRRPs, the output of one moment is related to the past and subsequent inputs. Thus, the bidirectional LSTM (Bi-LSTM) model (Fig. 2), which extends the direction of information transfer, is adopted [20]. Compared with LSTM, Bi-LSTM allows bidirectional information transfer using two hidden layers in inverse order. The dependency of Bi-LSTM is shown in Fig. 2; the output feature o_3 is jointly influenced by the current input t_3, past inputs t_3 and t_2, and future input t_4. Therefore, Bi-LSTM overcomes the shortcomings of RNN and LSTM by obtaining the long-range dependency between sequence inputs using layer-by-layer recursion. However, this dependency will gradually weaken as the length of the sequences grows, limiting the sequence modeling ability of Bi-LSTM.

2.2 CNN model

General CNN models contain convolutional and pooling layers in their convolutional modules [21]. This causes information loss due to the pooling layer discarding the position information of the sequences. Thus, the pooling layer is discarded when dealing with sequential modeling problems, and one-dimensional convolutional layers are directly stacked to process sequences data. The CNN with a two-layer convolutional layer is shown in Fig. 3; the dashed box indicates the location of the convolutional operation.

The dependency between the input sequences and output features is shown in Fig. 3; each output neuron has the same local receptive field size and is directly associated with the three input neurons. The strength of dependency on the three input

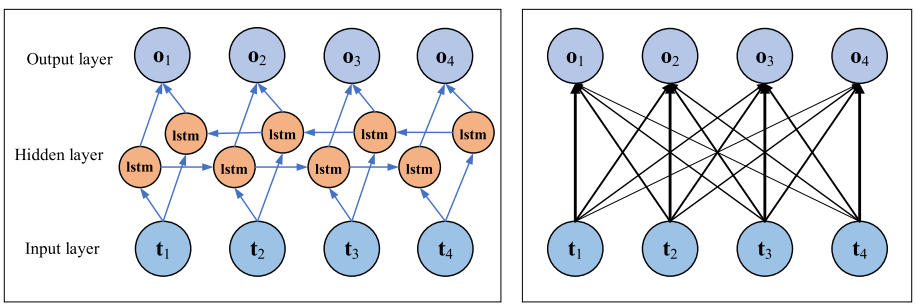

Fig. 2 Schematic of the bidirectional LSTM. Left: bidirectional LSTM structure; right: input sequence and output feature dependency

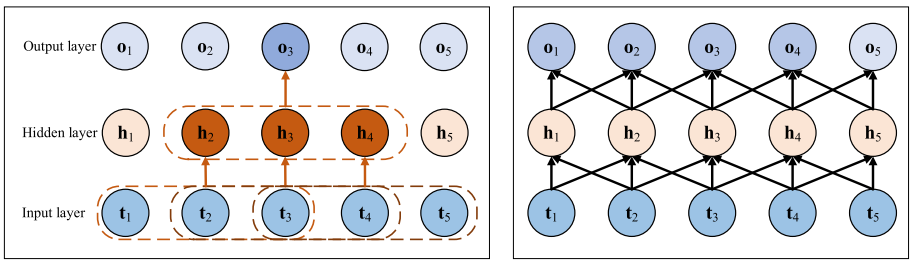

Fig. 3 Schematic of the long-range feature capture capability of CNN; left: CNN structure, right: input sequences and output feature dependency

neurons is equal. The longest sequence dependency distance captured by the first convolutional layer depends on the kernel size, while that captured by the second convolutional layer is 5. The convolutional layers required to associate any two inputs increase with increasing distance between the two inputs. Therefore, the CNN model is difficult to describe HRRP's global structure because the convolution operation can only capture limited local information. The connection between larger regions requires enhancing the perceptual field through multiple stacked layers.

3 HRRP recognition based on the CNN-BERT model

We propose a new deep learning framework for HRRP recognition named CNN-BERT, as shown in Fig. 4. The proposed framework contains four modules: data preprocessing, convolutional token embedding, BERT, and classifier modules. The functionalities of each module are discussed in this section.

3.1 Data preprocessing module

This module solves the intensity and translation sensitivity problems of HRRP. The intensity of HRRP is affected by many factors, such as target distance, radar transmitter power, and antenna gain; thus, the intensity of the same target's HRRP differs depending on observation conditions. This intensity sensitivity problem is solved by l_2 normalization in the preprocessing module. The raw HRRP sample can be expressed as $\mathbf{x} = [x_1, \ldots, x_l, \ldots, x_L]$, where x_l denotes the magnitude of the lth range cell within HRRP, and L denotes the total range cells. The intensity-normalized HRRP sample $\mathbf{x}_{\text{norm}}$ can be expressed as follows:

$$\mathbf{x}_{\text{norm}} = \frac{\mathbf{x}}{\sqrt{\sum_{l=1}^{L} x_l^2}}. \tag{1}$$

In addition, HRRP is obtained by intercepting the radar return with a range window. The translational motion of the target varies the position of the HRRP in the range window, a phenomenon known as the translational sensitivity of HRRP. Here, an absolute alignment method can overcome this sensitivity issue. Specifically, a cyclic shift operation on $\mathbf{x}_{\text{norm}}$ places the center of gravity G at the center of the range window as follows:

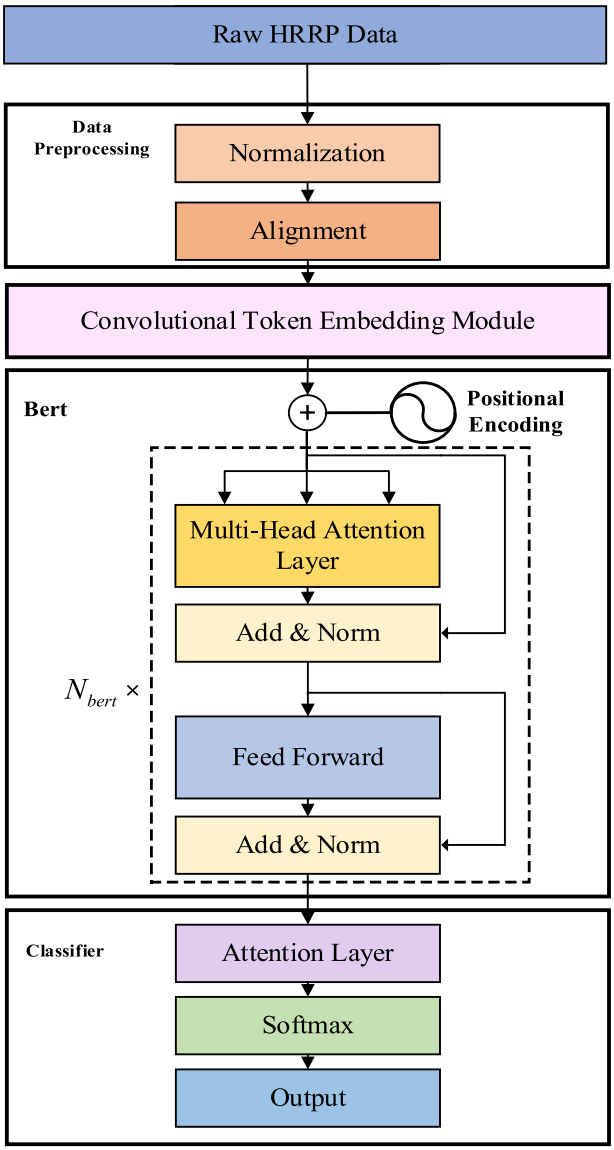

Fig. 4 HRRP recognition based on the CNN-BERT framework

$$G = \frac{\sum_{l=1}^{L} l \cdot \tilde{x}_l}{\sum_{l=1}^{L} \tilde{x}_l}, \tag{2}$$

where $\tilde{x}_l$ denotes the magnitude of the lth range cell within $\mathbf{x}_{\text{norm}}$.

The raw HRRP samples recorded consecutively for the same target and preprocessed HRRP samples are shown in Fig. 5.

3.2 Convolutional token embedding module

The convolutional token embedding module uses the convolutional operation to characterize the spatial structural features of the HRRP envelope and embeds the original HRRP to obtain the sequence features as input sequences for the BERT module. This idea

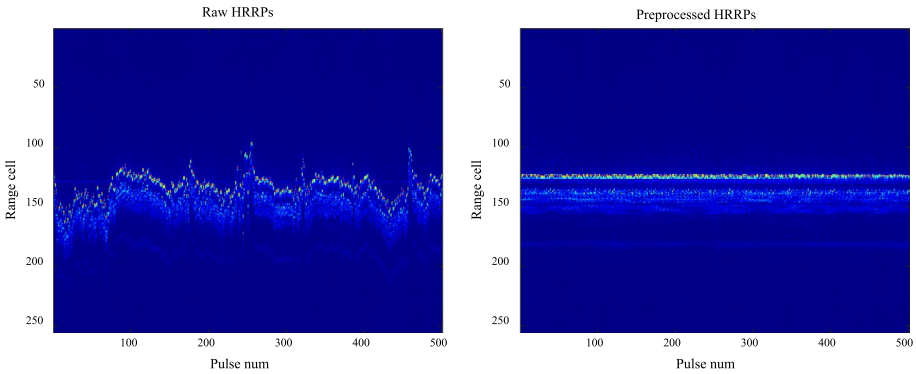

Fig. 5 Comparison of raw (left) and preprocessed (right) HRRP samples

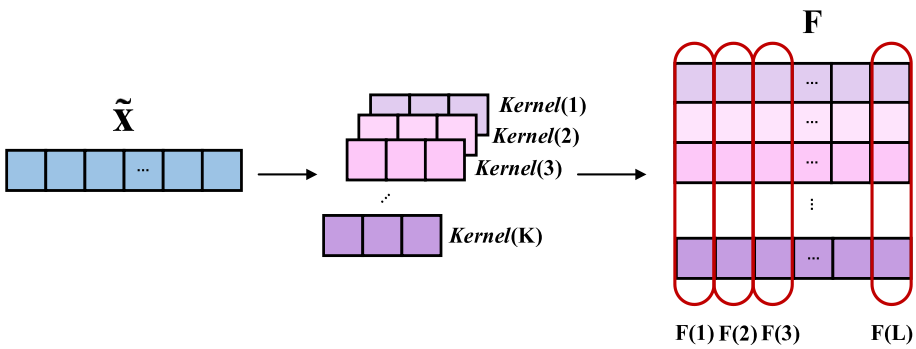

Fig. 6 Schematic of convolutional token embedding

was inspired by [25], which shows that early convolutions help transformers see better; we use convolutional operation instead of the time-domain segmentation and patchify methods to obtain input sequences for the BERT module. The time-domain segmentation method used in RNN or LSTM causes information redundancy and dimensionality-length constraints [23]. The direct patchify method, used by neural networks such as ViT [24], is implemented with a large convolutional kernel and large stride, violating the typical design of the convolutional layer. Moreover, the hard locality constraint in the early layer hinders the network's expressive ability. By contrast, the extracted sequence features by the convolutional token embedding module retain the local structure information in HRRP and have translation and scaling invariance. Moreover, the convolution kernel size and the number of the convolution channels independently control the number and dimension of token features to realize the decoupling of the dimensionality length. Furthermore, the sequence feature generation avoids hard locality constraints and enhances the initial expression ability of the network.

The convolutional token embedding module (Fig. 6) contains three parts: the convolutional layer, batch normalization (BN) layer, and activation layer. The preprocessed HRRP samples $\tilde{\mathbf{x}}$ are convolved by K one-dimensional convolutional kernels to obtain the output sequence $\mathbf{F}$, that is computed as

$$\mathbf{F}(l,k) = (\tilde{\mathbf{x}} \otimes \text{kernel}(k))(l), \tag{3}$$

where $\otimes$ denotes the convolutional operation and kernel(k) denotes the kth convolutional kernel. $\mathbf{F}(l) = \sum_{k=1}^{K} F(l,k)$ denotes the token embedding vector of the lth range cell.

The output sequence $\mathbf{F}$ passes through the BN and activation layers to generate the sequential embedding representation $\mathbf{F}_{\text{embedding}}$ of HRRP. The output sequence feature map is given in Fig. 7. The X-axis represents the range cell dimension, and the Y-axis represents the feature channel dimension. Further, the output feature of one channel is visualized on the right. It can be seen that the feature focuses more on the local characteristics of the target.

3.3 BERT module

BERT has demonstrated superior performance and is gradually replacing RNNs in long-term dependence modeling problems [26]. The BERT module uses the depth sequence encoding capability to extract temporal structural information embedded in input sequences and compensate for the lack of timing modeling capability of the convolutional token embedding module. The input sequence here refers to the sequential embedding representation of HRRP. The BERT module comprises a positional encoding layer and N_{bert} successive encoder blocks. Each encoder block comprises a multi-head self-attention layer that aggregates the relationship within the token embedding vector of the range cell, a feed-forward layer that extracts the feature representation at the position level, and an add and norm layer. The implementation details of each layer are as follows.

3.3.1 Positional encoding

The features extracted by the convolutional token embedding module do not explicitly include the positional relationship within range cell token embedding. The positional encoding technique fully uses the sequential relationship among range cells of HRRP. The sine and cosine functions can encode the odd and even bits of the input sequences, respectively, as follows:

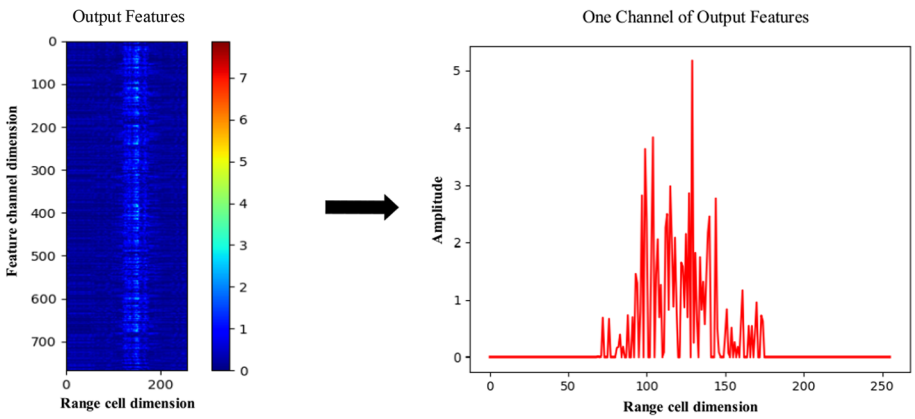

Fig. 7 Output sequence features a map of convolutional token embedding module; left: two-dimensional feature map, right: single channel feature

$$\mathbf{P}(l,k) = \begin{cases} \sin\left(l/10000^{k/d_{\text{model}}}\right) & \text{s.t. } k \bmod 2 = 0 \\ \cos\left(l/10000^{k-1/d_{\text{model}}}\right) & \text{s.t. } k \bmod 2 = 1 \end{cases}, \tag{4}$$

where l denotes the index of range cell in the input sequences, $P(l,k)$ denotes the kth element in the lth range cell of the positional encoding vector $\mathbf{P}$, with $0 \le l < L, k \le d_{\text{model}}$.

According to the properties of sine and cosine functions, $\mathbf{P}(l+i)$ of the $(l+i)$th range cell can be expressed as a linear combination of $\mathbf{P}(l)$ and $\mathbf{P}(i)$.

$$\mathbf{P}(l+i,k) = \begin{cases} \mathbf{P}(l,k) * \mathbf{P}(i,k+1) + \mathbf{P}(l,k+1) * \mathbf{P}(i,k) & \text{s.t. } k \bmod 2 = 0 \\ \mathbf{P}(l,k) * \mathbf{P}(i,k) - \mathbf{P}(l,k-1) * \mathbf{P}(i,k-1) & \text{s.t. } k \bmod 2 = 1 \end{cases} \tag{5}$$

The sequence feature map obtained by adding the output feature of the convolution module and positional encoding vector (Eq. 6) is shown on the right side of Fig. 8.

$$\mathbf{F}_{\text{conv\_emb}}(l,k) = \mathbf{F}_{\text{embedding}}(l,k) + \mathbf{P}(l,k) \tag{6}$$

The texture in the feature map after positional encoding represents the unique position information, strengthening the temporal structure in the extracted HRRP features.

3.3.2 Multi-head self-attention layer

The multi-head self-attention layer captures the local and global structure of input feature sequences and extracts the long-term dependency within range cells of HRRP.

3.3.2.1 Scaled dot-product attention The proposed framework adopts a scaled dot-product attention mechanism for fast execution and memory space efficiency. A transformation layer maps input sequences $\mathbf{F}_{\text{conv\_emb}} \in \mathbb{R}^{L \times d_{\text{model}}}$ to three different sequential vectors, i.e., query **Q**, key **K**, and value **V,** as follows:

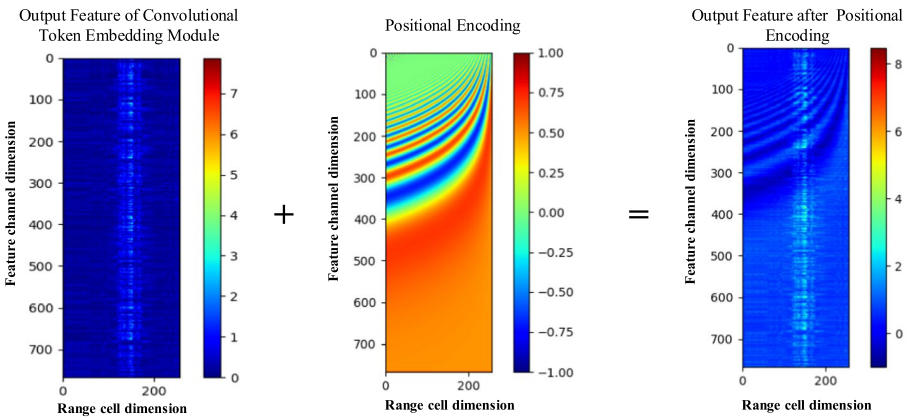

Fig. 8 Schematic of positional encoding; left: output feature map of the convolutional token embedding module, middle: positional encoding vectors, right: sequential feature map after positional encoding

$$\begin{aligned}\mathbf{Q} &= \mathbf{F}_{\text{conv\_emb}}\mathbf{W}_q \\ \mathbf{K} &= \mathbf{F}_{\text{conv\_emb}}\mathbf{W}_k, \\ \mathbf{V} &= \mathbf{F}_{\text{conv\_emb}}\mathbf{W}_v\end{aligned} \tag{7}$$

where $\mathbf{W}_q \in \mathbb{R}^{d_{\text{model}} \times d_q}$, $\mathbf{W}_k \in \mathbb{R}^{d_{\text{model}} \times d_k}$, and $\mathbf{W}_v \in \mathbb{R}^{d_{\text{mod el}} \times d_v}$ are the three weight matrices; d_q, d_k, and d_v are dimensions of the query, key, and value, respectively.

Secondly, as shown in Fig. 9, the query is explicitly aggregated with the corresponding key by calculating the product of $\mathbf{Q}$ and $\mathbf{K}$. A scaling factor $\sqrt{d_k}$ and Softmax operation are subsequently applied to get the attention weights of the value $\mathbf{V}$, also called an attention map. Combining the resulting attention weights with $\mathbf{V}$, we obtain the output features F_{selfatt} as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{8}$$

3.3.2.2 Multi-head self-attention mechanism The HRRP data have a typical multi-subspace structure [13], while the single-head self-attention module can only obtain limited information from one of these subspaces. Therefore, the multi-head attention mechanism extracts features from multiple subspaces to enrich the diversity of feature representations.

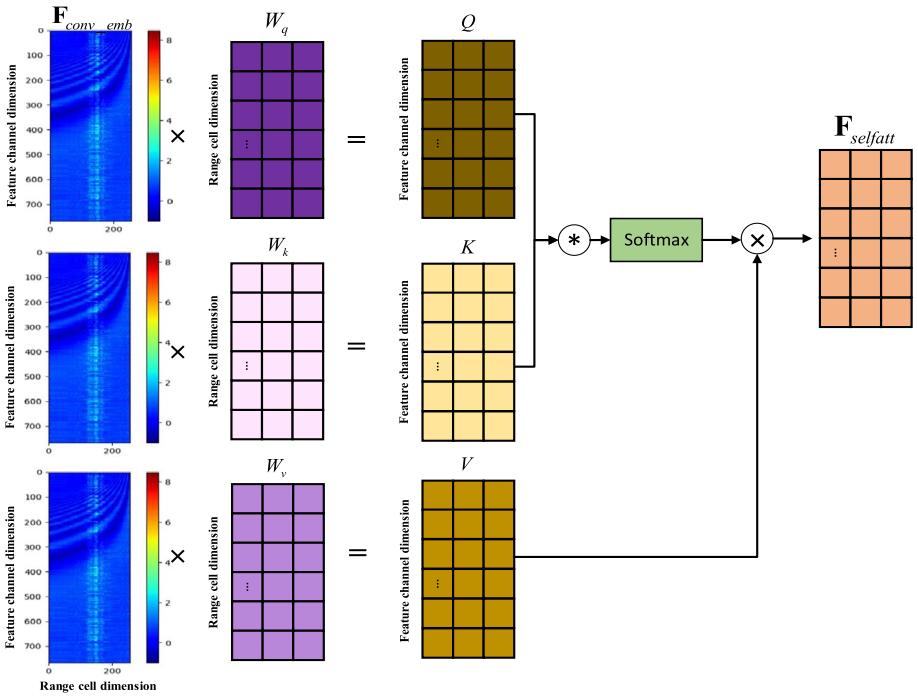

Fig. 9 Schematic of scaled dot-product attention

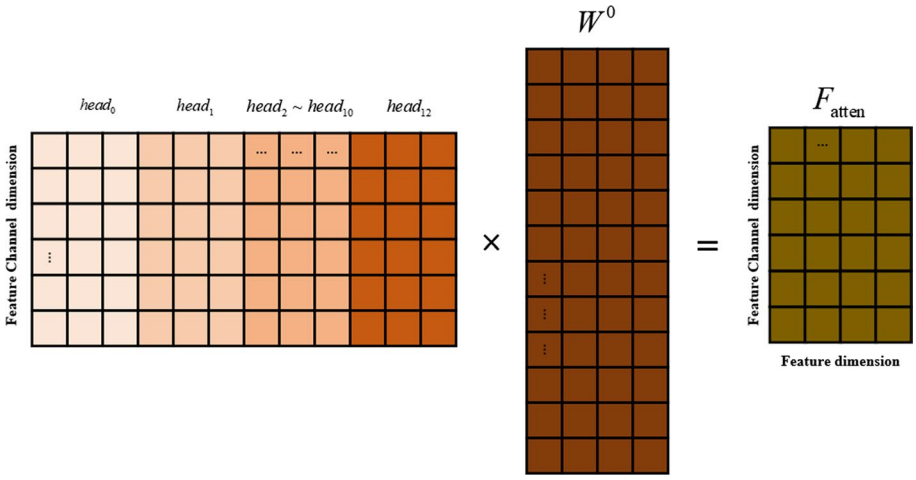

Fig. 10 Schematic of the multi-head self-attention mechanism

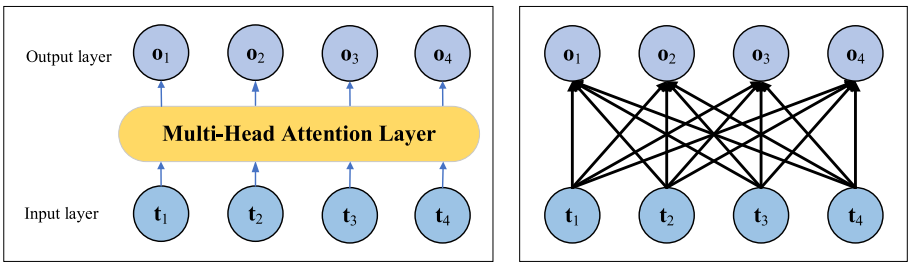

Fig. 11 Long-range feature capture capability of BERT module. Left: structure of the multi-head attention layer; right: the dependency relationship between input sequences and output features

As shown in Fig. 10, **Q**, **K**, and **V** are projected to multiple feature subspaces using several independent attention heads simultaneously. The resulting output vectors of each subspace are concatenated and mapped to the final output $\mathbf{F}_{\text{atten}}$ as follows:

$$\mathbf{F}_{\text{atten}} = \text{Concat}(\mathbf{head}_1, ...\mathbf{head}_h)\mathbf{W}^O, \tag{9}$$

where h is the head number, $\mathbf{head}_i = \text{Attention}(\mathbf{QW}_i^Q, \mathbf{KW}_i^K, \mathbf{VW}_i^V)$ denotes the resulting vectors of each head, $\mathbf{W}_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, and $\mathbf{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ are the three groups of weight matrices, and $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ is the output projected matrix.

3.3.2.3 Analysis of long-range feature extraction capability The multi-head attention mechanism is the core operation in the BERT module. All input sequences can be input into the multi-head attention layer simultaneously (Fig. 11 left), ensuring the parallelism capability of the model. Meanwhile, the feature dimension of the output and input layers are the same to facilitate the stacking of the BERT modules.

According to Eq. (8), the self-attention mechanism directly uses the product of **Q** and **K** to obtain the attention weights. Each element in the input sequences is compared with other elements, and the distance between each element is equal. Accordingly, the

schematic of the input sequences and output feature dependency is drawn on the right side of Fig. 11. Each output layer feature depends on the input sequences at all moments, and the dependency degree is the same without attenuation. Therefore, only one multi-head attention layer is needed, and the longest dependency distance captured by the output layer features is the length of the whole sequences. Thus, the BERT module can capture the global and local features using the multi-head attention mechanism.

3.3.3 Feed-Forward layer

The feed-forward layer enhances the separability of the extracted features using two successive feed-forward networks with a ReLU activation to map the feature representation to a high-dimensional hidden space. The output of the feed-forward layer is given as follows:

$$\mathrm{FFN}(x) = \max(0, x\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{10}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, and $\mathbf{b}_2$ represent the weight matrices and biases of two linear changes, and $\max(\cdot, \cdot)$ represent the maximum function.

3.3.4 Add and Norm layer

The add and norm layer performs residual connection and layer normalization (LN) operation. Since the gradient of the deep neural network during training will gradually vanish during the backpropagation process, adjusting the parameters of the previous layers is challenging. A residual connection can overcome the vanishing gradient problem caused by stacking multilayer BERT modules and facilitate the building of deeper models.

Moreover, LN can stabilize the model training process. Unlike BN in the convolutional token embedding module, LN can address the interval covariate shift problem [27]. Specifically, BN normalizes the features of the same channel among different samples, whereas LN normalizes the features of the same sample in different channels, and the computation is independent of the batch size parameter. The calculation process of LN can be expressed as follows:

$$\mathrm{LN}(x) = \alpha \times \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta, \tag{11}$$

where x is the input of the LN layer, μ and σ^2 denotes the mean and the variance, respectively, ε is a very small positive number, and α and β are the scaling and translation parameters, respectively. Let M denote the number of neurons in the LN layer; then μ and σ^2 can be calculated as follows.

$$\begin{aligned} \mu &= \frac{1}{M}\sum_{i=1}^{M} x_i, \\ \sigma^2 &= \frac{1}{M}\sum_{i=1}^{M} (x_i - \mu)^2 \end{aligned} \tag{12}$$

3.4 Classifier module

The classifier module comprises attention and Softmax layers. The attention mechanism strengthens the deep features useful for recognition by assigning weights to the output

features O_{bert} of the BERT module along the feature channel dimension. Thus, the features $\mathbf{F}_{\text{ATT}} = \{\mathbf{F}_{\text{ATT}}(l)\}_{l=1}^{L}$ can be obtained as follows:

$$\mathbf{F}_{\text{ATT}}(l) = \sum_{k=0}^{K} \alpha(l,k) O_{\text{bert}}(l,k), \tag{13}$$

where $O_{\text{bert}}(l,k)$ denotes the kth element in the lth range cell of the output feature vector and $a(l,k)$ denotes the weight of the corresponding elements of $O_{\text{bert}}(l,k)$. The proposed model can automatically learn $a(l,k)$ according to the importance of the features.

Next, linear mapping and Softmax operation are adopted to classify the feature $\mathbf{F}_{\text{ATT}}$. The posterior probability that $\mathbf{x}$ belongs to the cth target can be calculated as follows:

$$P_c(\mathbf{x}) = \frac{\exp\left(\mathbf{F}_{\text{s}}(c)\right)}{\sum_{i=1}^{C+1} \exp\left(\mathbf{F}_{\text{s}}(i)\right)}, \tag{14}$$

where $\mathbf{F}_s = \mathbf{W}_s \mathbf{F}_{\text{ATT}}$ and $\mathbf{W}_s$ is a weight matrix, $\mathbf{F}_{\text{s}}(i)$ refers to the ith element in the vector $\mathbf{F}_{\text{s}}$, C denotes the class number of in-library targets, $c \leq C+1$. Finally, an HRRP sample $\mathbf{x}$ is classified into the c_0-class as follows:

$$c_0 = \arg\max_{c} P_c(\mathbf{x}). \tag{15}$$

3.5 Cost function

The cost function determines the function and performance of the model. In RATR, besides recognition performance, identifying out-of-library targets is important. Thus, while designing the cost function, we consider the recognition and rejection performance simultaneously. The rejection function is integrated into our model by regarding the out-of-library samples as the $(C+1)$th class in the training process. A nonnegative regularization hyperparameter λ balances the recognition and rejection ability, and the cost function is defined as follows:

$$L = L_{\text{recognition}} + \lambda L_{\text{rejection}}, \tag{16}$$

where $L_{\text{recognition}} = -\frac{1}{N_1} \sum_{n=1}^{N_1} \sum_{c=1}^{C+1} z^{(n)} \ln p_c^{(n)}(\mathbf{x})$, and $L_{\text{rejection}} = -\frac{1}{N_2} \sum_{n=1}^{N_2} \sum_{c=1}^{C+1} z^{(n)} \ln p_c^{(n)}(\mathbf{x})$. N_1 denotes the total number of in-library samples identified as within the data library and outlier samples identified as out of the data library. N_2 denotes the total number of inner samples identified as out of the data library and outlier samples identified as within the data library. $N = N_1 + N_2$ represents the total number of samples in each mini-batch and $z^{(n)}$ represents the real label of the nth sample in the corresponding mini-batch. Positive and negative λ implies that the model is more concerned with rejection and recognition performances, respectively.

4 Training and testing procedure

The detailed training test flow is shown in Algorithm 1. We first preprocess the raw HRRP data and then initialize the model parameters in the training phase. After training the model with the mini-batch-based BP algorithm, the model parameters are saved for

testing. In the testing phase, we first preprocess the test HRRP samples, input these samples for forward propagation, and finally obtain the recognition results.

Algorithm 1 Training and testing procedure

Preprocessing	Input: HRRP data $\left\{\mathbf{x}_{c,n}\right\}_{c=1,n=1}^{C+1,N_c}$; N_c indicates the number of HRRP samples corresponding to the c-th target. For c = 1 to $C+1$ For n = 1 to N_c Apply Eqs. (1) and (2) to HRRP sample $\mathbf{x}_{c,n}$ to tackle the sensitivity problems. End End Output: Training data set $\left\{\tilde{\mathbf{x}}_{c,n}\right\}_{c=1,n=1}^{C+1,N_c}$.
Training stage	Input: Training dataset $\left\{\tilde{\mathbf{x}}_{c,n}\right\}_{c=1,n=1}^{C+1,N_c}$. Model initialization: Initialize parameter set $\mathbf{\Phi} = \{\theta_{CNN}, \theta_{\text{Bert}}, \theta_{ATT}\}$ and learning rate l_r; θ_{CNN}, θ_{Bert}, and θ_{ATT} represents the parameters of the convolutional token embedding module, BERT module, and multi-level attention layer, respectively. For epoch = 1 to Epochs Shuffle the dataset and divide it into K mini-batches. For *batch* = 1 to K (Forward propagation process) Calculate the features after the convolutional token embedding module using Eq. (3); Calculate the features after BN layer and ReLU layer; Calculate the features after positional encoding using Eq. (4). For n = 1 to N_{bert} Calculate the output of the BERT module using Eqs. (5) to (12). End Calculate the weight of attention using Eq. (13); Calculate the output label distribution using Eq. (14); (Backpropagation process) Calculate the cost function using Eq. (16); Calculate the stochastic gradient of each parameter in the parameter set using the gradient descent method for backpropagation. End End OUTPUT: Trained CNN-BERT model with $\mathbf{\Phi}$.
Testing stage	Input: Test data $\mathbf{x}_{\text{test}}$, trained model parameters $\mathbf{\Phi}$. Preprocess the test samples according to Eqs. (1) and (2); Compute the labels of test samples using forward propagation; Obtain the classification results of the test samples using Eq. (15).

Table 1 Radar and aircraft parameters

Radar parameters	**Center frequency**		**5520 MHz**
	Pulse repetition frequency		**400 Hz**
	Bandwidth		**400 MHz**
The plane	**Length (m)**	**Width (m)**	**Height (m)**
Yark-42	36.38	34.88	9.83
An-26	23.80	29.20	9.83
Cessna Citation S/II	14.40	15.90	4.57

5 Results and discussion

5.1 Experimental dataset

The recognition performance of the proposed model is examined by the measured data of three types of aircraft targets. Yark-42 is a large-sized jet, Cessna Citation S/II is a small jet, and An-26 is a medium-sized propeller aircraft [14, 29]. The division of training and test sets is consistent with that in [28]. The parameters of the radar and aircraft targets are shown in Table 1.

Furthermore, ten classes of simulated HRRPs are generated as train out-of-library samples to evaluate the recognition performance. Each class have 1600 HRRP samples. In addition, 1600 HRRPs of real aircraft targets are used as test out-of-library samples to detect the rejection performance in the model testing phase.

5.2 Model setup

5.2.1 Proposed model

The parameter settings of the CNN-BERT model are set as follows to evaluate the recognition and rejection performance of the proposed model. The kernel size S of the convolutional token embedding module is set to 5, the number of convolutional channels K to 768, and the step size to 1. For the BERT module, we set the attention head number h to 8, and N_{bert} to 6. The dimensions of W_1 and W_2 in the feed-forward layer are set to 768×3072 and 3072×768, respectively. For the cost function, λ in Eq. (16) is set to 2. The CNN-BERT model is built according to the above parameters.

5.2.2 Comparative models

The proposed model is compared with several conventional HRRP recognition models, including SVM and GMM in traditional models, AE and CNN in the deep non-time-series models, and RNN in the deep time-series models.

The SVM model is implemented using the LIBSVM toolbox and the kernel function with radial basis function. The GMM is implemented using the scikit-learn toolkit for python.

The AE model contains a stack of five AEs, where the number of neurons per layer is 300, 600, 900, 2000, and 3, respectively. The CNN model consists of three convolutional and two fully connected layers. The number of convolutional channels in

Table 2 Recognition results of different models (%)

Methods	An-26	Cessna	Yark-42	Average
GMM	94.98	81.33	99.09	91.80
SVM	94.95	92.20	98.67	95.27
CNN	91.88	85.86	97.7	91.81
AE	95.1	89.08	96.61	93.60
RNN	91.23	93.78	96.18	93.73
Proposed model	**99.90**	**98.80**	**99.80**	**99.50**

each layer is 8, 16, and 32, and the kernel size is 1×16 with a step size of 2. The two fully connected layers contain 300 and 3 neurons.

The RNN implementation is based on LSTM cells whose input sequences are extracted from HRRP samples based on the time-domain segmentation method with a sliding window. The sliding window step size is set to 16 and 8, respectively.

5.3 Recognition performance evaluation

5.3.1 Experimental results using all training data

Table 2 compares the recognition accuracy with three aircraft targets for the GMM, SVM, CNN, AE, RNN, and proposed model. Bold values in this Table means the highest average recognition rate (ARR). Compared with SVM and GMM models, the average recognition rate of the proposed model is 4.23% higher than the best SVM model. Compared with the AE and CNN models, the ARR of the proposed model is 5.90% higher than the best AE model. Compared with the RNN model, the ARR of the proposed model is improved by 5.77%.

The recognition performance needs to meet minimum standards for practical engineering applications by considering the overall recognition performance while balancing the recognition performance of each target type. Therefore, we compare the recognition balance of each method by analyzing the confusion matrix in Fig. 12. The difference between An-26 and Cessna aircraft with the highest and lowest recognition accuracy, respectively, is only 1.10%. Thus, the proposed method can model the characteristics of the three aircraft in a more balanced manner. Although the overall recognition accuracy of the comparative models exceeded 90%, only the SVM model exceeded 90% recognition accuracy for each type of aircraft target. Thus, based on the SVM model, Cessna and An-26 misjudge each other more. The difference between Yark-42 and Cessna aircraft with the highest and lowest recognition accuracy, respectively, is 6.47%. Thus, the SVM model fails to extract the unique attributes of each class of aircraft targets widening the gap between Cessna and An-26 and causing uneven recognition performance. The problem is considerably evident in CNN, GMM, AE, and RNN models. In contrast, our proposed model integrates local structure features of targets, and long-range features between range cells, fusing multi-level physical structure features for recognition. With its excellent nonlinear sequence modeling ability to extract better separable features, the recognition performance of each class is balanced.

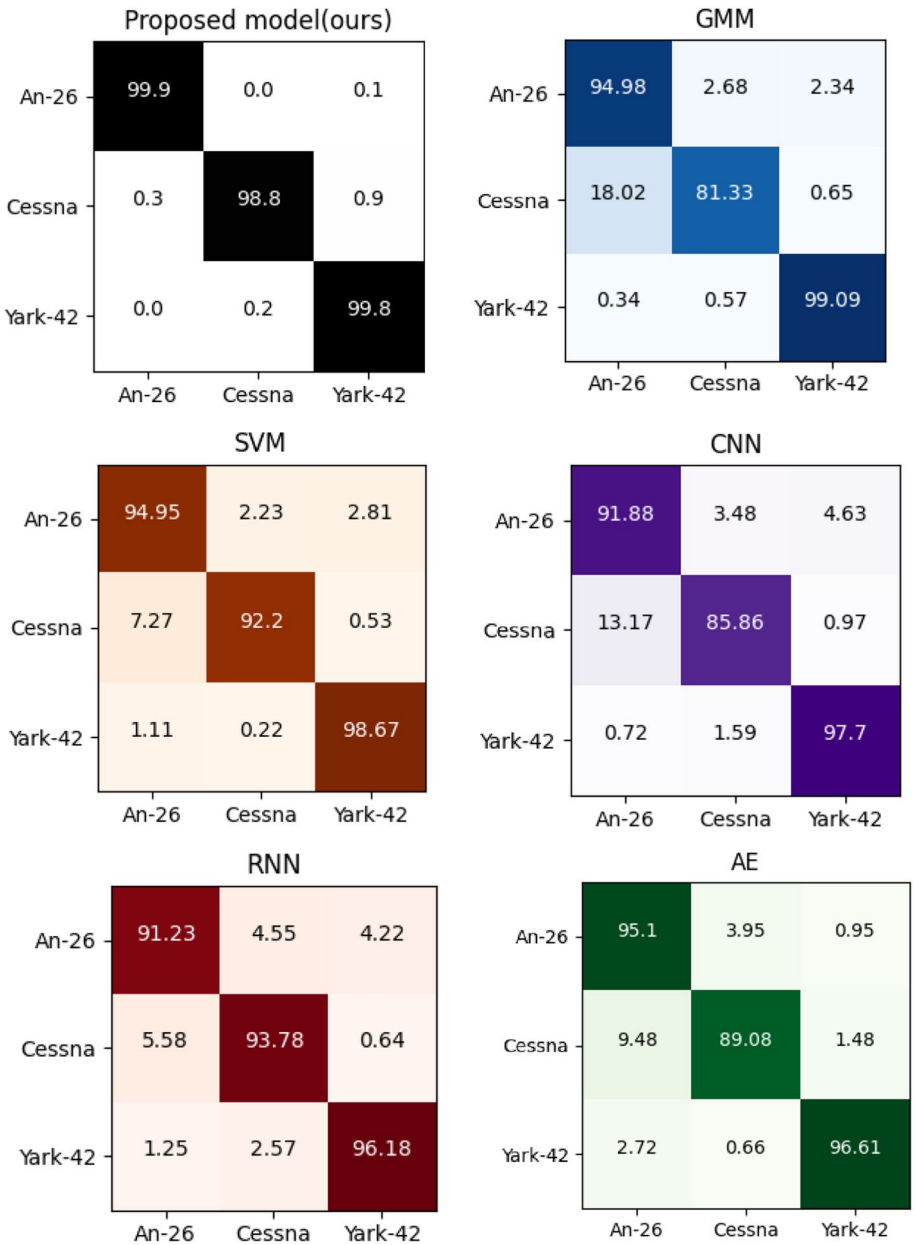

Fig. 12 Confusion matrix of different models

5.3.2 Recognition performance evaluation with different training sample sizes

To evaluate the impact of training set size on the recognition results, we sampled the training data set uniformly with different sampling rates and obtained 4 small sample sets of sizes 34,560, 8640, 2160, and 1080. The 137,880 HRRP training samples are divided into multiple frames, with 4 samples in each frame. Each frame randomly selects one HRRP sample to form the first small sample dataset. Those small sample sets are generated from the frames with sample sizes 4, 16, 64, and 128.

Table 3 compares the recognition accuracy with different training data sizes of the proposed model with that of the conventional models. The proposed model exhibits superior recognition performance under small samples condition compared with other

Table 3 ARRs of different models in different training sample sizes (%)

Training data number	34,560	8640	2160	1080
GMM	0.8731	0.8561	0.8312	0.7732
SVM	0.9167	0.8862	0.8591	0.8415
CNN	0.8807	0.8664	0.8554	0.7516
AE	0.9229	0.9179	0.8867	0.8511
RNN	0.9208	0.9159	0.8758	0.8564
Proposed model	**0.9910**	**0.9840**	**0.9810**	**0.9620**

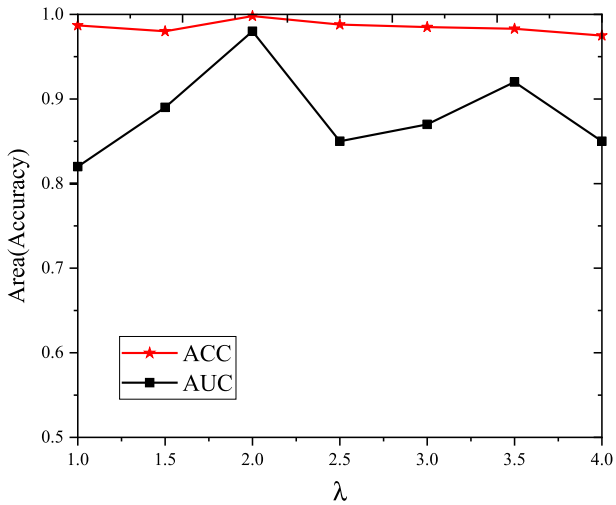

Fig. 13 Confusion matrix comparison of different methods using all training data

models. The smaller the number of training samples, the more prominent the effect of our model. Particularly, when the number of training samples is 1080, the proposed model can reach an ARR of 96.20%. Compared to our method, the performances of the SVM, GMM, AE, CNN, and RNN methods are lowered by 24.91%, 9.09%, 13.58%, 14.61%, and 12.10%, respectively. Moreover, the recognition accuracies of other models significantly decline when the training data number decreases. Thus, the proposed model can solve small sample problems.

5.4 Rejection performance evaluation

We integrate the out-of-library rejection task into the recognition model by introducing an importance parameter λ. We expect that the introduction of out-of-library samples in the training phase can widen the spacing between in-library and out-of-library samples without changing the differentiability of the in-library samples. Therefore, we used the idea of weighting to equalize the importance of in-library recognition loss and out-of-library rejection loss using λ.

Because unreasonable setting of λ can reduce the model's recognition and rejection performance, we first analyze the impact of λ on the proposed model before comparing the rejection performance of different models. We use the ARR and the area under the receiver operating characteristic curve (AUC) as the evaluation index of

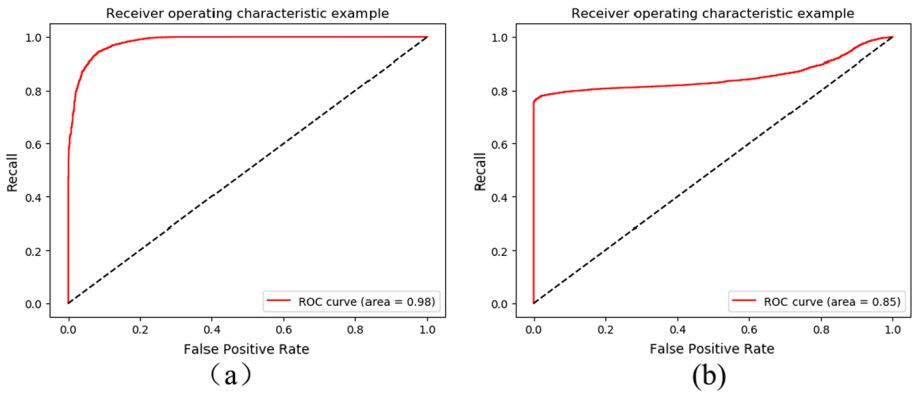

Fig. 14 AUC obtained by the proposed model. **a** AUC with $\lambda = 2$, **b** AUC with $\lambda = 2.5$

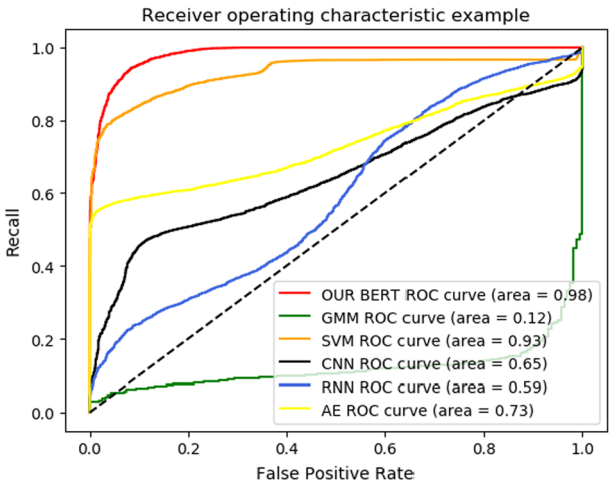

Fig. 15 Rejection performance of our model and comparative models on measured data

recognition and rejection performances, respectively [30–34]. We plot a line chart of AUC and accuracy with different λ, as shown in Fig. 13. The parameter λ influences the recognition and rejection performances; the rejection performance being more sensitive to λ variations. When $\lambda = 1$, the model degenerates to a general recognition model with a recognition function; the recognition accuracy is 0.987, while AUC is 0.82. When we increase λ, the model is more focused on the rejection performance. Thus, when $\lambda = 2$, the recognition and rejection performances of the model are improved, reaching the peak value; the recognition accuracy is 0.998, and AUC is 0.98. When λ continues to increase, recognition and rejection performance decreases. At this time, the model is overly concerned with rejection performance and ignores recognition performance, resulting in small loss weights for recognition, thereby decreasing recognition and rejection performances.

To graphically portray the influence of λ on the rejection performance, we plot AUCs with optimal and general λ as shown in Fig. 14. The AUC with $\lambda = 2$ and 2.5 is 0.98 and 0.85, respectively. Because λ significantly impact the rejection performance, we set $\lambda = 2$ for the subsequent evaluation of the rejection performance.

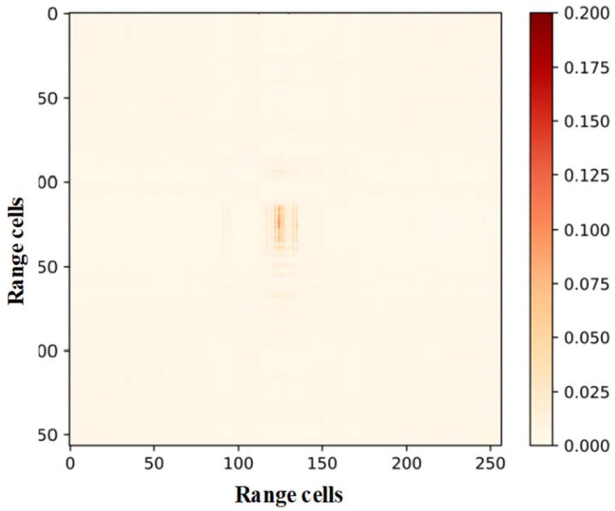

Fig. 16 Schematic of attention map

Figure 15 shows the receiver operating characteristic curve of models to quantify the rejection performance of each model. The AUC values are 0.98, 0.12, 0.93, 0.65, 0.59 and 0.73 for our proposed model ($\lambda = 2$), GMM, SVM, CNN, RNN, and AE models, respectively. Our proposed model rejects the out-of-library samples better than the other models. The introduction of out-of-library samples in the training phase and adjustment of the model cost function by the importance parameter λ enhances the rejection performance of the proposed model based on the guaranteed recognition performance.

5.5 Visualization

5.5.1 Visualization of long-range dependency

We also provides an intuitive and effective way to inspect the variation of long-range dependency at different layers. This is done by visualizing the attention map of the BERT module using Eq. (8). Attention values indicate the strength of the interdependence relationship between different range cell sequences and the importance of different range cells. In Fig. 16, the horizontal and vertical coordinates indicate the HRRP range cells. Brighter colors represent higher attention values in the attention map.

Attention maps of the self-attention layer in the shallow, middle, and deep layers of the BERT encoder block are shown in Figs. 17, 18, and 19, respectively. For simplicity, we only show the attention map of 5 heads, while the BERT encoder block has 8 heads. To show the interdependency between each range cell more comprehensively, we also give the average map of 8 different heads, which integrates the interdependency obtained by different heads from different perspectives. In Fig. 17a–i, the head learned interdependence relationship varies; the shallow layer BERT encoder block initially extracts the long-distance features and learns relatively strong interdependency within the range cells in the HRRP support area. Compared with Fig. 17, Fig. 18 shows the expansion of the strong correlation area, indicating that the middle layer BERT encoder block can better extract the long-range features. The attention map in the deep layer BERT encoder block in Fig. 19 shows that the important information

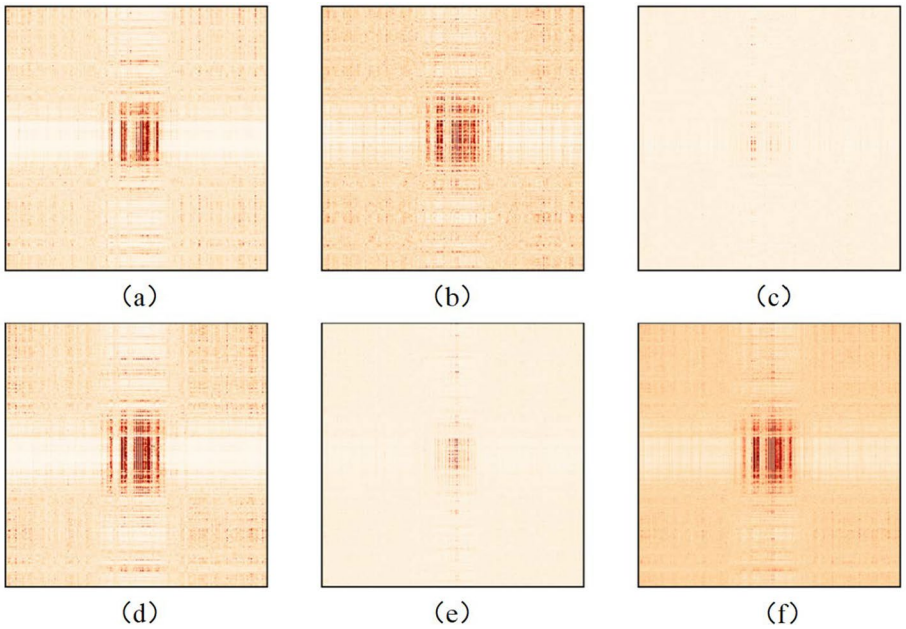

Fig. 17 Attention map of each head of shallow multi-head self-attention layer; **a**–**e** attention map of 5 different heads, **f** average attention map

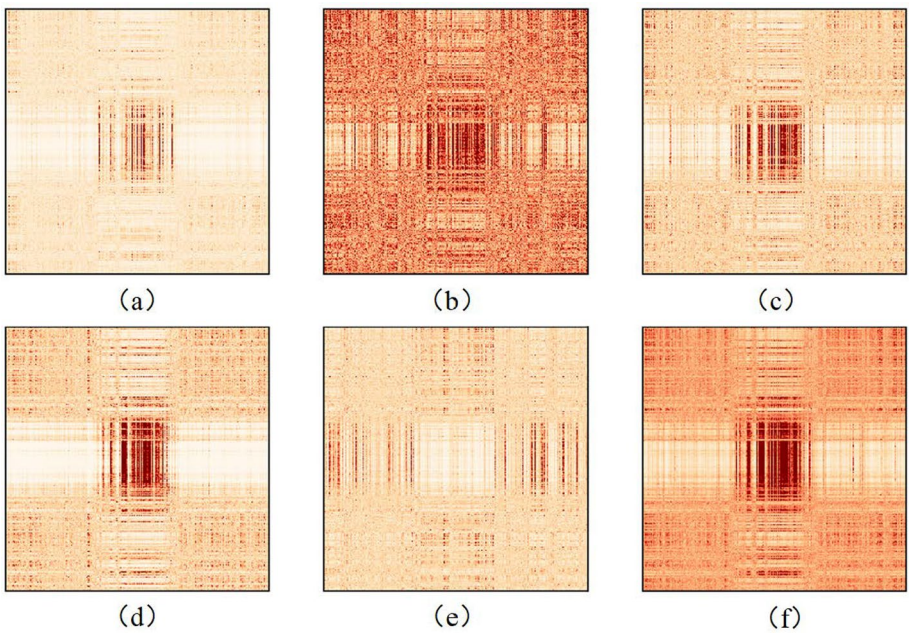

Fig. 18 Attention map of each head of middle multi-head self-attention layer; **a**–**e** attention map of 5 different heads, **f** average attention map

is aggregated to specific range cells, and the attention value is irrelevant to the query **Q**. As shown in Fig. 19f, the attention map shows vertical lines. Combined with the physical properties of HRRP, range cells in the support area can better reflect the radial size of the target and scattering point distribution. Moreover, it explains why

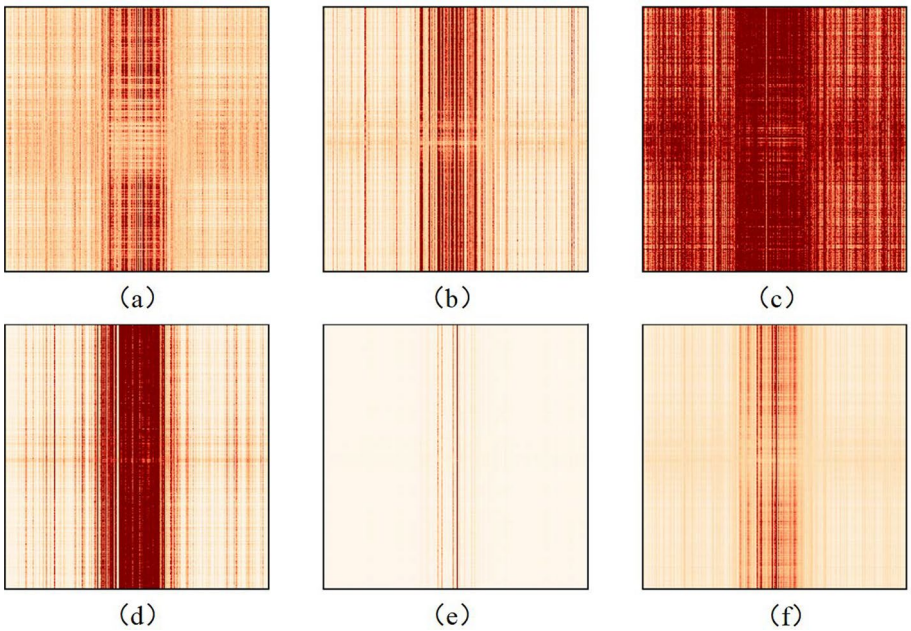

Fig. 19 Attention map of each head of deep multi-head self-attention layer; **a**–**e** attention map of 5 different heads, **f** average attention map

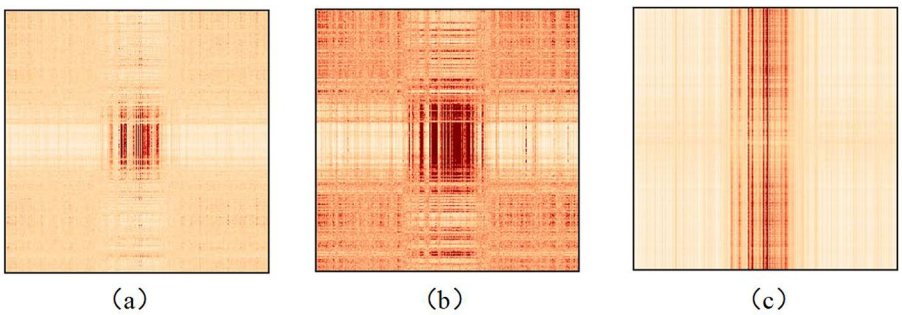

Fig. 20 Average attention map in the three encoder blocks of Yark-42 aircraft; **a** the shallow layer, **b** the middle layer, **c** the deep layer

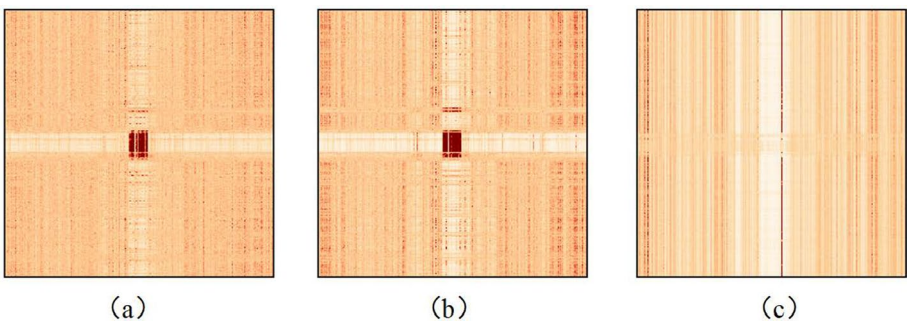

Fig. 21 Average attention map in the three encoder blocks of An-26 aircraft; **a** the shallow layer, **b** the middle layer, **c** the deep layer

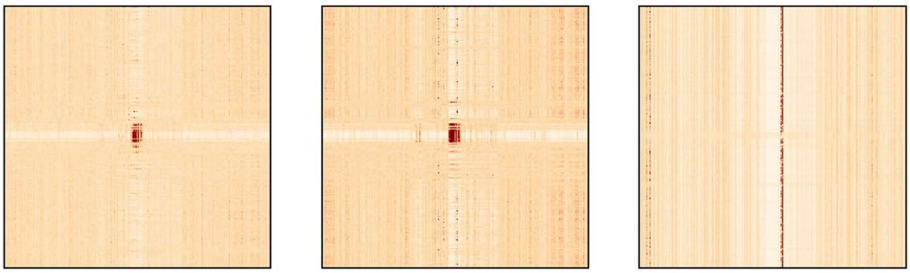

Fig. 22 Average attention map in the three encoder blocks of Cessna Citation S/II aircraft; **a** the shallow layer, **b** the middle layer, **c** the deep layer

the important information is mainly aggregated to the range cells in the support area and that the most core aggregation point is the peak position of HRRP.

We selected the representative attention map obtained from the three types of aircraft to observe the commonality and differences. In Figs. 20, 21 and 22a–c represent the average attention map in the shallow, middle, and deep layers of the BERT encoder block, respectively. A comparison of Figs. 20, 21 and 22 shows a significantly different strong correlation region size. The strong correlation region of Yark-42 aircraft is the largest, followed by that of An-26 and Cessna Citation S/II aircraft, which corresponds to the actual target size, as shown in Table 1. Therefore, the attention map extracted by the BERT module can reflect the target size information, indicating that the model has learned the physical structure variability between different targets.

5.5.2 Visualization of separability

For a simple and intuitive analysis of the separability of the features extracted by different models, the PCA visualization projections of the deep feature vectors extracted by our proposed model and the deep neural network model are given in Fig. 23; "other" indicates out-of-library samples. PCA operation is performed on the corresponding deep feature, and the 2D projection matrix is constructed using the principal components corresponding to the largest two feature values. The comparison of visualization performance reveals that our model has a smaller overlap region between in-library samples and between in-library and out-of-library samples than AE, CNN, and RNN models. The good separability and rejection performance further verify that the features extracted by the proposed model are suitable for recognition and rejection tasks.

6 Conclusion

This study proposed an improved BERT-based deep neural network for radar HRRP target recognition. The convolutional token embedding module provides the input sequence feature reflecting the local spatial structure of the target, and the BERT module describes the long-range dependency within the input sequence to extract deep temporal features. The experimental results reveal that the ARR of the proposed model is better than other comparative models when all training samples are applied and is more balanced across targets. In addition, even when the training sample size

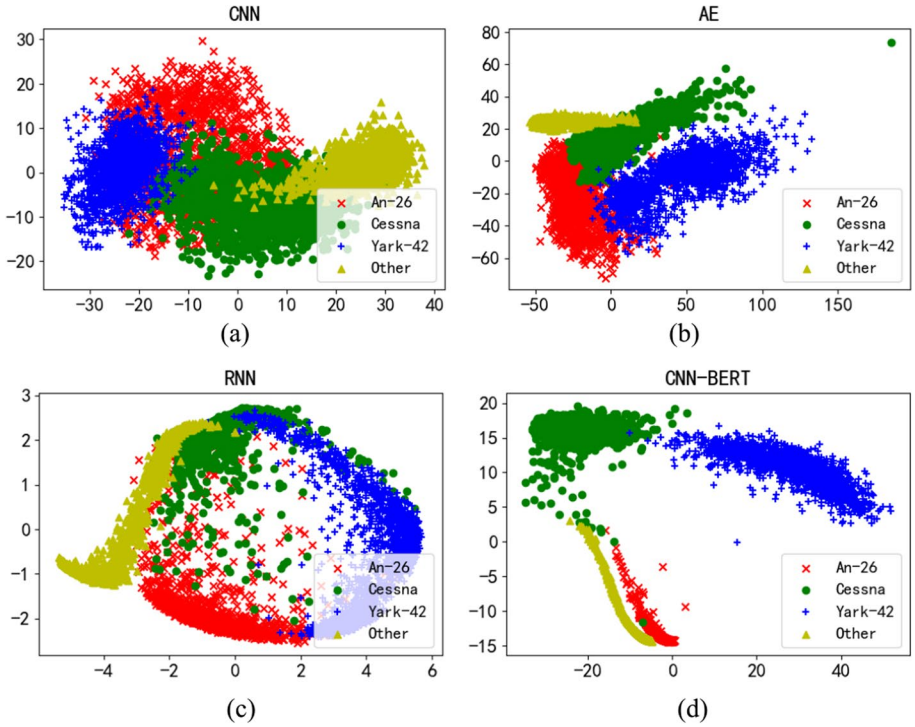

Fig. 23 Visualizations of the first two principal components of deep feature from different models. **a** CNN, **b** AE, **c** RNN, and **d** proposed model

is reduced to 1/128 of the original training samples, the ARR of the proposed model for each aircraft is over 96%. Finally, the proposed model has a much higher rejection capability with the AUC being 0.98 and can effectively deal with recognition tasks in complex environments. Thus, the proposed model has excellent engineering utility and extends the application of HRRP target recognition. In future work, we are devoted to lightweight deep learning model research and further improve computation and parameter efficiency of the proposed model.

Abbreviations

HRRP	High-resolution range profile
CNN	Convolutional neural network
RNN	Recurrent neural network
LSTM	Long short-term memory network
BERT	Bidirectional encoder representations from transformer

Acknowledgements
The authors would like to thank the handing editor and the anonymous reviewers for their valuable comments and suggestions for this paper. This work was supported in part by the National Natural Science Foundation under Grant No. 61701379 and the stabilization support of National Radar Signal Processing Laboratory under Grant No. KGJ202204.

Author contributions
PW and MP proposed the method and designed the experiments; PW, TC, and ST performed the experiments and wrote the paper; JD revised the paper. All authors read and approved the final manuscript.

Funding
This research was funded by the National Natural Science Foundation under Grant No. 61701379 and the stabilization support of National Radar Signal Processing Laboratory under Grant No. KGJ202204.

Availability of data and materials
Please contact author for data requests.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
The picture materials quoted in this article have no copyright requirements, and the source has been indicated.

Competing interests
The authors declare no competing interests.

Received: 4 May 2022 Accepted: 29 August 2022
Published online: 22 September 2022

References

1. O. Karabayır, O.M. Yücedağ, M.Z. Kartal et al. Convolutional neural networks-based ship target recognition using high resolution range profiles, in *2017 18th International Radar Symposium (IRS)*. IEEE, 2017, pp. 1-9.
2. L. Du, H. Liu, Z. Bao, Radar HRRP statistical recognition: parametric model and model selection. IEEE Trans. Signal Process. **56**(5), 1931–1944 (2008)
3. J. Lundén, V. Koivunen, Deep learning for HRRP-based target recognition in multistatic radar systems, in *2016 IEEE Radar Conference (RadarConf)* (IEEE, 2016), pp. 1–6
4. L. Du, H. He, L. Zhao et al., Noise robust radar HRRP target recognition based on scatterer matching algorithm. IEEE Sens. J. **16**(6), 1743–1753 (2015)
5. F. Chen, Q.Y. Hou, H.W. Liu et al., New adaptive angular-sector segmentation algorithm for radar ATR based on HRRP. J. Xidian Univ. **36**(3), 410–417 (2009)
6. J. Wang, Z. Liu, T. Li et al., Radar HRRP target recognition via statistics-based scattering centre set registration. IET Radar Sonar Navig. **13**(8), 1264–1271 (2019)
7. L.E.I. Lei, X.D. Wang, Y.Q. Xing et al., Multi-polarized HRRP classification by SVM and DS evidence theory. Control Decis. **28**(6), 861–866 (2013)
8. L. Du, P. Wang, H. Liu et al., Radar HRRP target recognition based on dynamic multi-task hidden Markov model, in *2011 IEEE RadarCon (RADAR)* (IEEE, 2011), pp. 253–255
9. J. Tu, T. Huang, X. Liu et al., A novel HRRP target recognition method based on LSTM and HMM decision-making, in *2019 25th International Conference on Automation and Computing (ICAC)* (IEEE, 2019), pp. 1–6
10. M. Pan, P.H. Wang, H.W. Liu et al., Radar HRRP target recognition based on truncated stick-breaking hidden Markov model. J. Electron. Inf. **35**(7), 1547–1554 (2013)
11. L. Du, H. Liu, Z. Bao et al., A two-distribution compounded statistical model for radar HRRP target recognition. IEEE Trans. Signal Process. **54**(6), 2226–2238 (2006)
12. D. Zhou, X. Shen, G. Wang et al., Orthogonal kernel projecting plane for radar HRRP recognition. Neurocomputing **106**, 61–67 (2013)
13. D. Zhou, Orthogonal maximum margin projection subspace for radar target HRRP recognition. EURASIP J. Wirel. Commun. Netw. **2016**(1), 1–11 (2016)
14. L. Shi, P. Wang, H. Liu et al., Radar HRRP statistical recognition with local factor analysis by automatic Bayesian Ying-Yang harmony learning. IEEE Trans. Signal Process. **59**(2), 610–617 (2010)
15. J. Wan, B. Chen, Y. Yuan et al., Radar HRRP recognition using attentional CNN with multi-resolution spectrograms, in *2019 International Radar Conference (RADAR)* (IEEE, 2019), pp. 1–4.
16. R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in *International Conference on Machine Learning* (PMLR, 2013), pp. 1310–1318
17. S. Kanai, Y. Fujiwara, S. Iwamura, Preventing gradient explosions in gated recurrent units, in *Advances in Neural Information Processing Systems* (2017), p. 30.
18. S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncert. Fuzziness Knowl. Based Syst. **6**(02), 107–116 (1998)
19. M. Sundermeyer, R. Schlüter, H. Ney, LSTM neural networks for language modeling, in *Thirteenth Annual Conference of the International Speech Communication Association* (2012).
20. M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
21. J. Wan, B. Chen, Y. Liu et al., Recognizing the HRRP by combining CNN and BiRNN with attention mechanism. IEEE Access **8**, 20828–20837 (2020)
22. J. Song, Y. Wang, W. Chen et al., Radar HRRP recognition based on CNN. J. Eng. **2019**(21), 7766–7769 (2019)
23. M. Pan, A. Liu, Y. Yu et al., Radar HRRP target recognition model based on a stacked CNN-Bi-RNN with attention mechanism. IEEE Trans. Geosci. Remote Sens. **60**, 1–14 (2021)
24. A. Dosovitskiy, L. Beyer, A. Kolesnikov et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020).
25. T. Xiao, M. Singh, E. Mintun et al., Early convolutions help transformers see better. Adv. Neural. Inf. Process. Syst. **34**, 30392–30400 (2021)
26. J. Devlin, M.W. Chang, K. Lee et al., Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

27. J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
28. L. Du, P. Wang, H. Liu et al., Bayesian spatiotemporal multitask learning for radar HRRP target recognition. IEEE Trans. Signal Process. **59**(7), 3182–3196 (2011)
29. B. Feng, B. Chen, H. Liu, Radar HRRP target recognition with deep networks. Pattern Recogn. **61**, 379–393 (2017)
30. Q. Li, B. Li, Z. Yang, Plane HRRP rejection based on SVDD technology, in *2011 3rd International Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)* (IEEE, 2011), pp. 1–4
31. D. Zhou, R. Wang, C. Zheng et al., Gamma model-based target HRRP rejection, in *Proceedings of the 2012 International Conference on Information Technology and Software Engineering* (Springer, Berlin, Heidelberg, 2013), pp. 349–356
32. X. Zhang, P. Wang, L. Du et al., New method for radar HRRP recognition and rejection based on weighted majority voting combination of multiple classifiers, in *2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)* (IEEE, 2011), pp. 1–4
33. Y. Wang, W. Chen, J. Song et al., Open set radar HRRP recognition based on random forest and extreme value theory, in *2018 International Conference on Radar (RADAR)* (IEEE, 2018), pp. 1–4
34. J. Wan, B. Chen, B. Xu et al., Convolutional neural networks for radar HRRP target recognition and rejection. EURASIP J. Adv. Signal Process. **2019**(1), 1–17 (2019)
35. X. Li, Z. Guo. A bi-sru neural network based on soft attention for hrrp target recognition, in *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)* (IEEE, 2019), pp. 1–5
36. B. Xu, B. Chen, J. Wan et al., Target-aware recurrent attentional network for radar HRRP target recognition. Signal Process. **155**, 268–280 (2019)

Publisher's Note