Digital Signal Processing 133 (2023) 103874

Contents lists available at ScienceDirect

Digital Signal Processing

journal homepage: www.elsevier.com/locate/dsp

Robust Bayesian attention belief network for radar work mode recognition ☆

Mingyang Du [a,*], Ping Zhong [b], Xiaohao Cai [c], Daping Bi [a], Aiqi Jing [d]

[a] College of Electronic Engineering, National University of Defense Technology, Hefei, 230037, China
[b] National Key Laboratory of Science and Technology on ATR, National University of Defense Technology, Changsha, 410000, China
[c] School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK
[d] School of Foreign Languages, Shanxi Datong University, Datong, 037000, China

ARTICLE INFO

Article history:
Available online 7 December 2022

Keywords:
Radar work mode
Pulse descriptor word
Attention mechanism
Bayesian neural network
Robustness
Recognition

ABSTRACT

Understanding and analyzing radar work modes play a key role in electronic support measure system. Many classifiers, for example those based on convolutional neural network (CNN) and recurrent neural network (RNN), are available for recognizing radar work modes as well as emitter types from their waveform parameters. However, the performance of these methods may suffer significantly when confronting different types of signal degradation, e.g., measurement error, lost pulse and spurious pulse. To tackle this issue, we in this paper develop a Bayesian attention belief network (BABNet) based on Bayesian neural networks in which the probability distribution over weights can help to enhance the model robustness for corrupted data. In particular, we adopt pre-trained CNN as the Bayesian inference prior. This not only accelerates the convergence speed, but also avoids the training process getting stuck in bad local minima. Meanwhile, instead of using RNNs which are difficult to be implemented in parallel, the combination of padding operation and attention module in the proposed BABNet enables CNN, as the backbone, to process sequential data with variable length. Extensive experiments are conducted to demonstrate the recognition capability and robustness of the BABNet in different environments.

1. Introduction

Specific emitter identification (SEI) is the process of discriminating or identifying different emitters (e.g., cognitive radios [1], Wi-Fi devices [2] or Internet of things [3]) by exploiting the radio frequency fingerprint features extracted from the intercepted signals. SEI in military application serves to identify particular copies of radars with the same type in electronic intelligence system on the contemporary battlefield [4]. This task is connected with developing effective methods to generate distinctive features extracted from radar signals [5].

For radar emitters, radio frequency features can be ambiguously categorized into inter-pulse and intra-pulse characteristics [6]. The complex modulation patterns in both classes, as well as the dense electromagnetic environment, result in emerging high diversity and unpredictability of radiated signals [7]. Specifically, inter-pulse parameters of incoming radar signals include for example radio frequency (RF), pulse repetition interval (PRI), pulse width (PW), pulse amplitude, angle of arrival, time of arrival and type of scan [8]. The aforementioned parameters can be utilized to depict and analyze the generalized state (i.e., work mode) of the radar emitter including search, acquisition, tracking, imaging and missile guidance [9]. Therefore, after determining these parameters, it is possible to identify the emitter type, functional purpose and platform, providing intelligence for the electronic support measures (ESM) system to perform threat detection and area surveillance tasks [10]. Furthermore, radar signals may also contain modulations within each pulse, i.e., intra-pulse modulation, consisting of the intentional modulation and unintentional modulation. The former refers to the specific variations on amplitude, phase, and frequency versus time. An SEI method based on the intentional modulation patterns can be found in e.g. [11]. The unintentional modulation characteristics, such as pulse front-edge, instantaneous amplitude [12] and temporal cumulants of pulse signals [13], are related to the hardware imperfection generally. In this paper, we mainly focus on the inter-pulse feature for SEI.

Deep learning technology has dramatically improved state-of-the-art in various recognition tasks. Some supervised classification methods based on the convolutional neural network (CNN) or re-

☆ This document is the results of the research project funded by the National Natural Science Foundation of China (Grant no. 61971428).
* Corresponding author.
E-mail address: dumingyang17@nudt.edu.cn (M. Du).

https://doi.org/10.1016/j.dsp.2022.103874
1051-2004/

current neural network (RNN) have been introduced to SEI [14], as well as inter-pulse modulation type recognition in recent years. Since RNNs are often better to process the sequential input by learning the implicit dependency of all past elements of the sequence, some RNN-based methods were proposed to tackle the radar signal stream data. In [15], sequential patterns of consecutive pulses were firstly represented by a series of PRI and PW, and then a vanilla gated recurrent unit (GRU) was trained to perform pattern classification, denoising, and deinterleaving of pulse streams. The work in [16] assigned a single GRU to individual inter-pulse feature and introduced attention mechanism to radar emitter classification, which was proved effective for highly missing and spurious pulse ratio. Compared to these coarse-grained "seq2one" schemes that obtain one class label with one pulse sequence as input, the work in [17] presented a hierarchical long short-term memory (LSTM) which yields a label sequence for each pulse, such that it can accurately determine transition boundaries of each class. In addition, forming distinctive local motifs by CNN provides another optional solution for extracting the correlated semantic information of radar sequence data. The work in [18] constructed a one-dimension CNN model with 4 convolutional layers, 2 max-pooling layers and 2 fully-connected (FC) layers to distinguish different PRI modulation types. In particular, CNNs and RNNs can be combined so as to utilize joint temporal and spatial characteristics to recognize the sequence-based radar signals as proposed in [19,20].

The performance of deep learning methods (e.g. the abovementioned ones) may experience significant degradation when confronting different types of non-ideal scenarios, e.g., measurement error, lost pulse and spurious pulse. One of the main reasons is that these methods are prone to incur extrapolations with unjustified high confidence for the training data and thus cannot generalize well on the noisy test data [21]. Several ways have been investigated to tackle this problem. For example, the work in [22] proposed to adopt transfer learning to exploit the common knowledge between data with different signal-to-noise ratio (SNR) so as to obtain robust features against noise corruption. However, this method needs both clean data and noisy data for training, and the recognition performance could degrade if the SNR of the test data is out of the range of the training data. Bayesian neural networks (BNNs) [23–27] have been explored in e.g. visual analysis as well as speech recognition [24,25]. They place probability distributions over the weights of the neural networks to model uncertainty and obtain predictive results for the *out of distribution* data. In this paper, we focus on the problem of radar work mode recognition and construct a Bayesian attention belief network (BABNet) to learn the time-dependency of radar sequential inter-pulse data. In particular, since the representation learning capabilities of the standard Transformer structure are not that effective for training samples with limited size, we simplify the encoder-decoder attention layers in BABNet with the "key-query-value" framework adopted in the previous work (e.g., [26,27]), without sacrificing the recognition performance.

Our main contributions are as follows.

1) Inspired by the previous work regarding the Bayesian neural networks, we model the determined weights in traditional deep neural networks as probabilistic distributions. This strategy will help to enhance the robustness of deep models on unseen and corrupted test data and mitigate the overfitting issue. The details about the objective function and derivation of the posterior probability by variational approximation can be found in Section 2.3. In addition, we adopt the pretrained CNN as the prior of Bayesian inference in neural networks, which avoids potential detrimental consequences resulting from initialization with standard Gaussian distribution, as well as speeds up the convergence in the training procedure. As demonstrated in the numerical experiments, our method enhances the representation power and yields systematic improvements in terms of test accuracy and robustness compared to the state-of-the-art.
2) We introduce the combination of padding operation and attention mechanism for CNNs as an alternative to RNNs when processing the variable-length sequential data. Specifically, the input pulse data will firstly be aligned to unified length by padding, and then fed into the subsequent operations, e.g., convolution, pooling and linear transformation, in the same way as the common existing practice. After that, the attention module assigns weights corresponding to position information, which can highlight dominant features of informative signals while weaken dispensable features of padding elements. In addition, the summation operation after the attention module aggregates the weights along the length direction so as to reduce the dimension of the feature vectors. Thus, the number of units in the following classifier is vastly pruned, and the computation costs and storage requirement are dramatically reduced.
3) We conduct comprehensive experiments on two synthetic radar inter-pulse sequential data sets. One is the collection simulating various work modes of an airborne MFR; the other is composed of several typical inter-pulse parameter modulation patterns. We demonstrate an obvious boost in performance of the proposed method compared to the RNN-based and CNN-based methods on both data sets. Furthermore, we extensively evaluate our approach on practical challenges, such as robustness ability across data sets in the case of non-ideal conditions, e.g., measurement error, lost pulse and spurious pulse. In view of the test performance on the aforementioned situations and the one-dimension loss landscape, the proposed BABNet with well-designed prior information can achieve significant advances in radar work mode recognition, yielding favorable results than those of the ones compared.

The remainder of this paper is organized as follows. We first recall the radar pulse stream structure, different inter-pulse modulation types and BNNs in Section 2. The novel BABNet is designed in Section 3. In Section 4 we report extensive experimental results in evaluating the detection accuracy and robustness of the proposed method against state-of-the-art techniques. The conclusion and future work are presented in Section 5.

2. Preliminary

2.1. Radar pulse stream structure and work mode

Radar signals are passively intercepted by the receiver portion of the ESM system [28]. Firstly, the stream of successive pulses from several observed emitters is fed to perform deinterleaving (or sorting), grouping pulse trains from the same emitter. Upon detection of a radar pulse, most receivers measure parameters including RF of the carrier wave, PW, pulse amplitude, angle-of-arrival and time-of-arrival. They are then digitized and assembled into a data structure called pulse descriptor word (PDW). After that, some of the statistical PDW parameters, as well as other parameters that are derived from the sequence of the grouped PDWs (e.g. PRI), are collected to reflect the characteristics of each emitter.

The modulation of forgoing parameters can operate under several different modes to perform various functions such as searching, tracking and missile guidance [29]. Three typical parameters, RF, PW and PRI, are used to form the feature vector in this paper to perform the radar work mode identification task. All of them

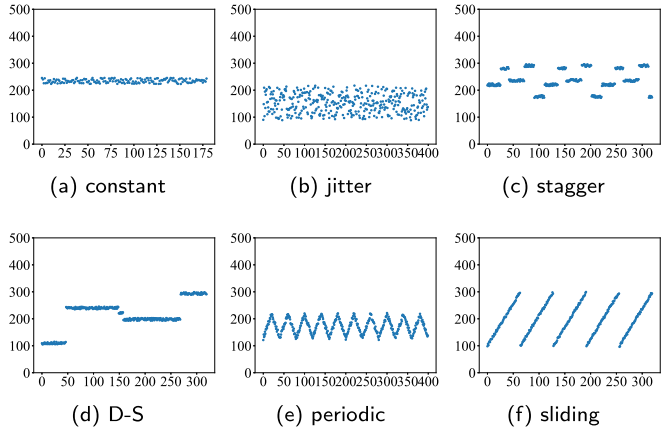


Fig. 1. Different inter-pulse modulation types with the individual representations of the radar pulse parameter sequence shown in each figure.

are determined by the radar's mission and specifications [30]. Basically, the characteristics associated with the low RF radar (i.e., under 3 GHz) are low atmosphere absorption and easy to generate high power. The parameter PW can be used to provide coarse information on radar types. For example, weapon radars generally have short pulses. Regarding PRI, it can limit the maximum unambiguous range of a pulse radar.

2.2. Inter-pulse parameter modulation

PRI modulation patterns, as well as PW and RF agility patterns, are important information to recognize the radar work modes and identify the emitters. Basically, different inter-pulse modulation types can be divided into constant, jitter, stagger, dwell and switch (D-S), periodic and sliding [7], see Fig. 1. In detail, if the variation of one parameter is less than 1% of its mean value, it can be categorized as *constant* type. Instead, if that variation exceeds about 30% of the mean value, it is the *jitter* type. The parameter in the *stagger* pulse train switches in a periodic manner and the number of stagger positions can vary from 2 even up to 64. A *D-S* pulse type consists of two or more stages of stable values while may not be repeated periodically. The *periodic* pulse train means the parameter performs sinusoidal or possibly triangular variation. The *sliding* pulse train indicates that its parameter is either successive/monotonic increase or decrease.

2.3. Bayesian neural network

The work in [31] presents that large networks normally require large amounts of training data; otherwise, the overfitting issue occurs prohibitively. Generally, the model complexity increase leads to overfitting [32]. BNNs use probabilistic model weights. The amount of information they contain could simplify the network [23]. Thus, BNNs offer robustness to overfitting and can easily learn from small datasets [33]. Below we summarize the mathematical derivation of the back-propagation, reparameterization and sampling in BNNs.

Let $\boldsymbol{w}$ be the set of parameters or weights of a neural network. For an input signal x, a neural network could be viewed as a probabilistic model $P(y|x, \boldsymbol{w})$ returning a probability output y given an input x. For the pulse sequence recognition task in this paper, $P(y|x, \boldsymbol{w})$ is a categorical distribution corresponding to the softmax loss [34].

According to the Bayesian inference, a prior distribution is introduced firstly over the space of functions $P(\boldsymbol{w})$. Then, we can calculate the posterior distribution of the weights, i.e. $P(\boldsymbol{w}|x, y)$, given the training dataset. For the test data $\hat{x}$ with unknown label $\hat{y}$, its distribution can be calculated by taking expectation, i.e., $P(\hat{y}|\hat{x}) = \mathbb{E}_{P(\boldsymbol{w}|x,y)}[P(\hat{y}|\hat{x}, \boldsymbol{w})]$. However, this is generally intractable since the number of parameters $\boldsymbol{w}$ for neural networks can be rather large and the functional form of a neural network is not exact as to integral. To address this issue, variational approximation was proposed to find the Bayesian posterior distribution regarding the weights [35]. For a variational posterior distribution $q(\boldsymbol{w}|\theta)$ with a set of parameters θ, its approximation to the true Bayesian posterior on the weights $P(\boldsymbol{w}|x, y)$ can be achieved by minimizing their Kullback-Leibler (KL) divergence regarding θ, i.e.,

$$
\begin{aligned}
&\min_{\theta} \mathrm{KL}[q(\boldsymbol{w}|\theta)||P(\boldsymbol{w}|x, y)] \\
&= \min_{\theta} \int q(\boldsymbol{w}|\theta) \log \frac{q(\boldsymbol{w}|\theta)}{P(\boldsymbol{w}|x, y)} \mathrm{d}\boldsymbol{w} && (1) \\
&= \min_{\theta} \int q(\boldsymbol{w}|\theta) \log \frac{q(\boldsymbol{w}|\theta) P(x, y)}{P(\boldsymbol{w}) P(x, y|\boldsymbol{w})} \mathrm{d}\boldsymbol{w} && (2) \\
&= \min_{\theta} \int q(\boldsymbol{w}|\theta) \Big[\log \frac{q(\boldsymbol{w}|\theta)}{P(\boldsymbol{w})} - \log \frac{P(x, y|\boldsymbol{w})}{P(x, y)} \Big] \mathrm{d}\boldsymbol{w} && (3) \\
&= \min_{\theta} \big\{ \mathrm{KL}[q(\boldsymbol{w}|\theta)||P(\boldsymbol{w})] - \mathbb{E}_{q(\boldsymbol{w}|\theta)}[\log P(x, y|\boldsymbol{w})] \big\}. && (4)
\end{aligned}
$$

Note that the term $\log[P(x, y)]$ is omitted in the last step of the above minimization since it is a constant regarding θ. The first term in Eqn (4) is the KL divergence between the variational posterior and the prior distribution on the weights. Since we expect $q(\boldsymbol{w}|\theta) \to P(\boldsymbol{w}|x, y)$, the second term in Eqn (4) can be calculated by the log likelihood loss in the training procedure.

We here suppose that the variational posterior $q(\boldsymbol{w}|\theta)$ follows Gaussian distribution, i.e., $q(\boldsymbol{w}|\theta) \triangleq \mathcal{N}(\mu, \sigma^2)$, where $\sigma = \log[1 + \exp(\rho)]$; then θ can be parameterized as $\theta = (\mu, \rho)$. The prior distribution $P(\boldsymbol{w})$ is also modeled as Gaussian distribution, i.e., $P(\boldsymbol{w}) \triangleq \mathcal{N}(\mu_0, \sigma_0^2)$. Thus, the KL divergence shown in Eqn (4) becomes the comparison between two Gaussian distributions, i.e.,

$$
\begin{aligned}
&\mathrm{KL}[q(\boldsymbol{w}|\theta)||P(\boldsymbol{w})] \\
&= \mathrm{KL}[\mathcal{N}(\mu, \sigma^2)||\mathcal{N}(\mu_0, \sigma_0^2)] && (5) \\
&= \int \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(t-\mu)^2}{2\sigma^2}} \log \frac{\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(t-\mu)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi}\sigma_0} e^{\frac{-(t-\mu_0)^2}{2\sigma_0^2}}} \mathrm{d}t && (6) \\
&= \int \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(t-\mu)^2}{2\sigma^2}} \Big[\log \frac{\sigma_0}{\sigma} - \frac{(t-\mu)^2}{2\sigma^2} + \frac{(t-\mu_0)^2}{2\sigma_0^2} \Big] \mathrm{d}t && (7) \\
&= \log \frac{\sigma_0}{\sigma} - \frac{1}{2} + \frac{\sigma^2 + (\mu - \mu_0)^2}{2\sigma_0^2} . && (8)
\end{aligned}
$$

For the mini-batch training, we follow the stochastic variational method [35] and sample the t-th weight of the network, say w_t, from $q(\boldsymbol{w}|\theta)$ in the backward propagation. We could construct an unbiased estimator given by

$$
\begin{aligned}
&\mathrm{KL}[q(\boldsymbol{w}|\theta)||P(\boldsymbol{w})] \\
&= \int q(\boldsymbol{w}|\theta) \log \frac{q(\boldsymbol{w}|\theta)}{P(\boldsymbol{w})} \mathrm{d}\boldsymbol{w} && (9) \\
&\approx \frac{1}{T} \sum_{t=1}^{T} \log q(w_t|\theta) - \log P(w_t) , && (10)
\end{aligned}
$$

and

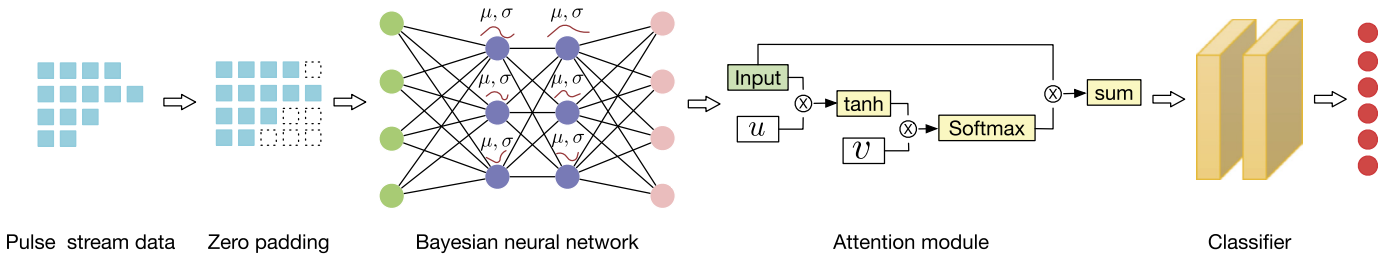


Fig. 2. Structure of the proposed BABNet. Every filter weight in the Bayesian neural network follows a probability distribution rather than a fixed value. The radar pulse sequential data with non-identical length is firstly padded to a unified length. Then the Bayesian neural network and attention module perform the high-level feature extraction task for the classifier to distinguish different category samples and yield robust detection/classification results.

$$\mathbb{E}_{q(\boldsymbol{w}|\theta)}[P(x, y|\boldsymbol{w})]$$
$$= \int q(\boldsymbol{w}|\theta) P(x, y|\boldsymbol{w}) \mathrm{d}\boldsymbol{w} \tag{11}$$
$$\approx \frac{1}{T} \sum_{t=1}^{T} P(x, y|w_t) , \tag{12}$$

where T is the number of elements in the weights $\boldsymbol{w}$. Thus, the cost function to be minimized in Eqn (4) reads

$$\mathrm{KL}[q(\boldsymbol{w}|\theta)||P(\boldsymbol{w})] - \mathbb{E}_{q(\boldsymbol{w}|\theta)}[\log P(x, y|\boldsymbol{w})]$$
$$\approx \frac{1}{T} \sum_{t=1}^{T} \log q(w_t|\theta) - \log P(w_t) - \log P(x, y|w_t) . \tag{13}$$

We remark that the BNN module shown in Fig. 2 only depicts the sketch idea about parameterizing the weights to probability distributions. The BNN model architecture in practice is similar to the non-probabilistic deep neural networks, containing e.g. convolutional layers, deconvolutional layers and FC layer, with its weights updated using Eqn (13) in the Bayesian manner.

3. Methodology

In this section, we present our developed BABNet (i.e., Bayesian attention belief Network) for radar work mode recognition, with advantages of preventing overfitting in deep neural networks and enhancing robustness in tackling new and corrupted data, see Fig. 2 for its overall architecture. It is based upon *Bayes by Backprop* [34] and *Bayesian CNN* [36], which introduced probability distribution over the weights in deep neural networks. Its main components are as follows: i) zero elements padding of the original input data utilized to attain unified length sequence data; ii) a Bayesian neural network whose weights follow probability distributions; iii) an attention module with a self-attention mechanism being developed to formulate feature's importance in the radar work mode recognition; and iv) a classifier for final radar work mode recognition. It is worth emphasizing that the proposed method provides an alternative and competitive solution in processing sequence data with arbitrary length against the RNN-based methods, which process the input sequence one by one and therefore are difficult, if not impossible, to be implemented in parallel.

3.1. Sequential radar pulse data processing

Taking radar pulse sequence as time series signal, we define the work mode as the integration of several inter-pulse parameters, e.g., PRI, PW and RF. From this point of view, let $\boldsymbol{X} = [\boldsymbol{x}_{\mathrm{PRI}}, \boldsymbol{x}_{\mathrm{PW}}, \boldsymbol{x}_{\mathrm{RF}}]$ denote the pulse sequence data, where $\boldsymbol{x}_{\mathrm{PRI}} = [x^1_{\mathrm{PRI}}, x^2_{\mathrm{PRI}}, \cdots, x^N_{\mathrm{PRI}}]$, $\boldsymbol{x}_{\mathrm{PW}} = [x^1_{\mathrm{PW}}, x^2_{\mathrm{PW}}, \cdots, x^N_{\mathrm{PW}}]$, $\boldsymbol{x}_{\mathrm{RF}} = [x^1_{\mathrm{RF}}, x^2_{\mathrm{RF}}, \cdots, x^N_{\mathrm{RF}}]$, and N is the length of the intercepted pulse sequence data. However, N is generally different for the data obtained from different work modes, plus the existence of some non-ideal conditions such as lost pulse or spurious pulse, which brings great challenges in implementation of deep neural networks.

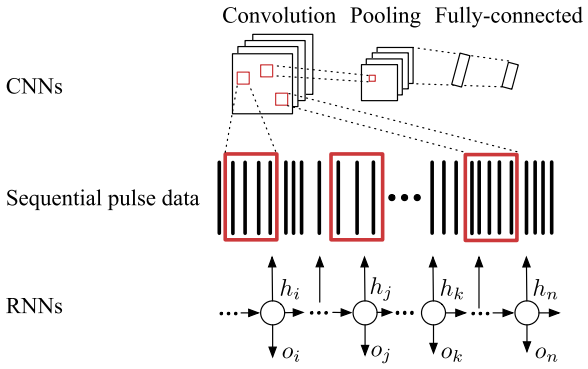


Fig. 3. The schematic diagram of the difference between CNNs and RNNs in processing sequential pulse data.

RNNs can process sequential data with variable length, i.e., only the hidden state updated up to the "end of character" will be fed to the following module (e.g., the classifier). In contrast, the convolution and pooling operations in CNNs gather local patches in the feature map and connect them with previous layers. In doing so, the distinctive local motifs are easily detected for the classification task with different patterns or categories [37]. The schematic diagram of the difference between RNNs and CNNs in processing sequential data is displayed in Fig. 3. This motivates us to use CNN as a substitute for RNN to process sequential data.

Using CNN, the number of nodes in the FC layer being followed by the classifier is required to be fixed, which is incompatible with changeable input data. To overcome this challenge, an intuitive solution is firstly padding all the sequence data to the unified length. Then we can feed it into the network and implement subsequent operations including convolution, pooling, linear transformation, etc. Unavoidably, the introduction of padding elements might disturb the network in finding the most salient and vital high-level features for the given task. To address this issue, we introduce the idea of utilizing the attention mechanism to divert the weights to the most important region of the sequence data and disregard the irrelevant parts, i.e., the padded elements.

As illustrated in Fig. 2, our attention module contains two randomly initialized hyper parameters, u and v, assigning weights to the output h from the BNN by the operation

$$h \cdot \mathrm{Softmax}[v \tanh(u \cdot h)]. \tag{14}$$

It is then followed by a summation operation summing all the hidden feature states $s \in \{1, \cdots, S\}$, i.e.,

$$\hat{h}_s = \sum_{s=1}^{S} h_s \cdot \mathrm{Softmax}[v_s \tanh(u_s \cdot h_s)], \tag{15}$$

where S is the number of elements in the hidden state and u_s, v_s, h_s are u, v, h at state s, respectively. Note that the summation of all hidden states on the length direction can significantly reduce the dimension of the feature space. For example, assuming the size of the feature vectors extracted from the variable-length input is [batch size × channels × length], it then turns into [batch size × channels] after forward passing the attention module. Here, the number of channels is determined by the previous module, i.e., the BNN in our method. Therefore, the discrepancy between the input variable-length sequence data is eliminated before being fed into the classifier. Moreover, the summation operation in Eqn (15) also greatly prunes the nodes in the FC layer, i.e. from [channels × length] to [channels], especially for long sequence data.

3.2. Bayesian prior information selection

In Bayesian theory, the prior should be chosen in a way that reflects the belief about the parameter [38]. It is crucial but rather hard to map the subjective beliefs onto the probability distributions unambiguously. As presented in Section 2.3 and Eqn (8), it is common practice to choose a "uninformative" prior distribution, such as a standard Gaussian distribution in BNNs. However, a bad or misspecification prior may result in a collapse consequence for the inference.

Before performing experiments on BNNs, we find that there exists a gap for some CNN-based models between their learning curves and test accuracy on the corrupted data sets (see Fig. 8 and Fig. 10). However, it has been proved in practice that local minima and saddle points have very similar values for the objective function and have a similar quality compared to the global minima for large models [37]. Thus, the possible reason for this problem is that these models are shallow (with only four convolutional layers at most) and might have got stuck at saddle points in the training procedure.

Since those shallow CNN-based models are able to converge to high training accuracy (see Fig. 8), continuing to deepen them to improve the robustness is not appropriate or else leading to overfitting. To overcome the gap between the training accuracy on the clean data and test accuracy on the corrupted data, we suggest utilizing pre-trained weights from CNN as the prior of the BNN counterpart. Specifically, instead of sampling from the standard Gaussian distribution as the Bayesian inference prior, we directly load the weights of the convolutional layers and FC layers in a pre-trained CNN to the corresponding modules in BNN. Different from the standard Gaussian prior with explicit mathematical formula, the suggested pre-trained CNN weights can be viewed as a *learning prior* computed by gradient descent to inform the prior choices [38]. This way could speed up the convergence especially at the early training stage and find more stable minima close to the ones in the pre-trained CNN. Compared to the initialization with Gaussian distribution, training on the basis of the pre-trained CNN will avoid BNNs converging to arbitrary unpromising states, since the pre-trained CNN with high accuracy has reached the position close to the robust and even global minima. Finally, variational Bayesian inference replacing the traditional approach is adopted to update weights of the neural network, further enhancing the network's robustness on corrupted test data. From the experiments in the next section we indeed find that this way could avoid some potential detrimental performance compared to the uninformative Gaussian prior.

4. Experimental results

In this section we validate the effectiveness of the proposed BABNet and compare its performance with state-of-the-art methods in terms of recognizing different radar work modes and robustness in non-ideal conditions. The radar inter-pulse signal data sets detailed below are generated by Numpy, a fundamental package for scientific computing with Python. All the experiments are run on Jetbrains Pycharm 2020 with a CPU of Intel(R) Core(TM) i9-9900k @ 3.60 GHz and a GPU of NVIDIA GeForce RTX 3090.

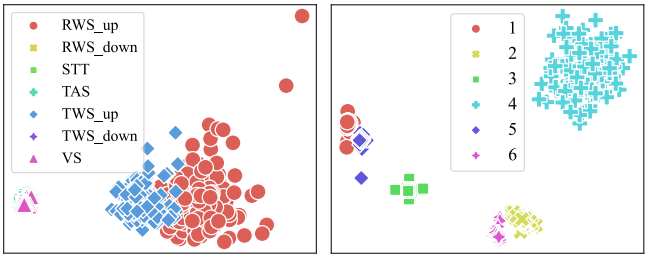


Fig. 4. Visualization of the distribution of databases I and II utilized in our experiments by t-SNE.

4.1. Data

Two databases were created for validation and comparison. Both are collections of the PDW sequences composed of three parameters, i.e., RF, PW and PRI. In particular, database I simulates the pulse stream data which comes from an airborne MFR with several work modes, consisting of tracking while scanning (TWS), tracking and scanning (TAS), single target tracking (STT), velocity searching (VS) and ranging while scanning (RWS) [39]. With respect to the pitching angle of radar line of sight, the individual scanning modes (i.e., TWS and RWS here) in the airborne radar can be further categorized as looking up and looking down (shorthand for "-up" and "-down" in Fig. 4 (a)) corresponding to executing the air-to-air and air-to-ground (or air-to-sea) missions, respectively [40]. Database II is a direct combination of multiple inter-pulse modulation types, which is slightly adapted based on the parameter setting in [41,17]. The inter-pulse parameter modulation types can be categorized to constant, stagger, periodic, sliding, jitter and D-S, as described in Section 2.

We apply t-SNE [42] to visualize the distribution of these two databases, see Fig. 4. Tables 1 and 2 depict the details of the attributes of databases I and II, respectively. There are 10,000 samples for each pattern in both databases. For the difference of these two databases, in database I, the Euclidean distance between different work modes is fairly close, with a big difference in terms of the number of pulse for each pattern (e.g., 16 pulses for TWS-up versus 512 pulses for RWS-down); the sophistication of database II is introduced by multiple inter-pulse modulation types, while the length of each sample is relatively similar, ranging from 180 to 400.

There are three kinds of corrupted scenarios which are utilized to validate the robustness of the proposed method and the methods compared, namely the measurement error, lost pulse and the spurious pulse [15]. In detail, the measurement error is the additive Gaussian white noise with zero mean and standard deviation ranging from 10% to 65% of the signal magnitude with step size 5%, shorthand for [10%, 65%, 5%]. Components absence and spuriousness in radar ESM processing arise due to sensor limitations, mistakes in deriving parameters such as PRI (see Fig. 5), as well as external electronic jamming. This implies that the classifier may encounter partial input patterns and randomly added pulses [28]. The indexes of the lost pulses and spurious pulses are totally random and discrete, where the proportions of those are set to [40%, 95%, 5%] and [10%, 90%, 10%], respectively.

Table 1
The parameter setting of different work modes of the airborne multi-function radar in Database I.

Index	Work mode	PRI/μs	PW/μs	RF/MHz	Number
1	TWS-up	1300	5.2	6.25	16
2	TWS-down	10.6	2.6	10	512
		11.4	2.6	10	512
		11.6	2.6	10	512
		11.8	2.6	10	512
		12.2	2.6	10	512
3	STT	12.2	2.6	10	8
		13.4	2.6	10	8
		14.6	2.6	10	8
		15.4	2.6	10	8
		15.8	2.6	10	8
		16.2	2.6	10	8
		16.4	2.6	10	8
		16.6	2.6	10	8
4	VS	5.3	1.55	2	1024
		5.7	1.67	2	1024
		5.8	1.70	2	1024
		6.1	1.79	2	1024
		6.7	1.96	2	1024
		7.3	2.14	2	1024
		7.7	2.26	2	1024
5	RWS-up	2000	50	6.25	16
6	RWS-down	13.0	2.6	10	512
		14.2	2.6	10	512
		14.6	2.6	10	512
		15.4	2.6	10	512
		16.2	2.6	10	512
7	TAS	13.4	2.6	10	512
		14.2	2.6	10	512
		14.6	2.6	10	512
		15.4	2.6	10	512
		15.8	2.6	10	512
		16.2	2.6	10	512
		16.6	2.6	10	512
		17.2	2.6	10	512

Table 2
The setting of the modulation type, ranges and pulse number in Database II.

Index	PRI/μs	RF/MHz	PW/μs	Number
	[100, 300]	[9000, 9500]	[1, 50]	
1	Constant	Stagger	D-S	180
2	D-S	D-S	D-S	200
3	Stagger	Stagger	D-S	240
4	Periodic	Sliding	Stagger	300
5	Sliding	Constant	Constant	320
6	Jitter	Jitter	Constant	400

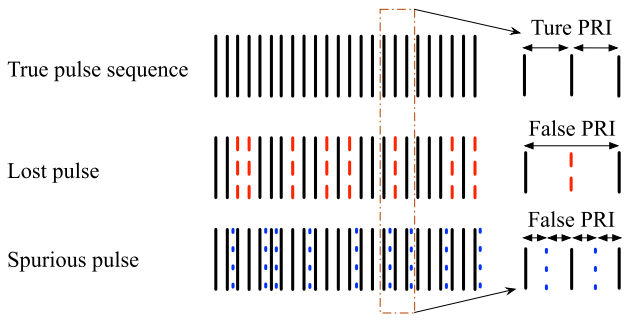


Fig. 5. The schematic diagram of the measurement imprecision for pulse repetition interval (PRI) in the lost pulse and spurious pulse scenarios.

4.2. Baseline methods and implementation details

Both the CNN-based and RNN-based radar work mode recognition approaches, e.g., vanilla CNN, auto-encoder [43], LSTM-CNN [19], GRU-attention [44] and multi-RNNs [16], are employed in our comparison. In particular, after padding the variable-length sequence PDW stream to a unified format, we replace the BNN module in our BABNet by the foregoing methods and then treat them as baselines. An FC layer is appended to all these models, with softmax acting as the activation function. Their implementation details and hyper parameters involved are listed below.

i) **Vanilla CNN.** There is only one convolutional layer (unless otherwise specified) to extract the pulse sequence features. The number of filters, kernel size and padding size are set to 32, 3 and 1, respectively.
ii) **Auto-encoder.** The encoder is composed of two convolutional layers with 32 and 64 neurons, respectively. The stride and kernel size are respectively set to 1 and 3. The decoder has the symmetrical structure as the encoder, with Batch Normalization (BN) followed to stabilize the training process.
iii) **LSTM-CNN.** The LSTM has a single layer with hidden size 64. The shallow CNN has two convolutional layers with 16 filters and zero padding, followed by the BN layer. The kernel size and stride are set to 5 and 3, respectively.
iv) **GRU-attention.** The GRU has three layers with hidden size 64. The attention weight is initialized as uniform distribution between -0.1 and 0.1.
v) **Multi-RNNs.** There are three GRUs for the three features of the pulse stream. Each GRU has a single layer with hidden size 16. We only applied one attention mechanism on the GRU for the PRI pulse stream. The stochastic gradient descent (SGD) algorithm is adopted with an initial learning rate 10^{-4}.

The whole data set is separated into training, validation and test sets, which account for 60%, 20% and 20% samples, respectively. The batch size is respectively set to 40, 256 and 512 for the BNN-based architectures, CNN-based architectures and RNN-based architectures. The number of filters in each convolutional layer in BNN is 32, with stride set to 1 and no padding. The mean value μ and variance σ^2 of Gaussian distribution in BNN are initially set to 0 and 0.1, respectively. All the models are trained 200 epochs. Adam optimizer is used unless particularly indicated, with an initial learning rate 10^{-4} and then exponential decay every 20 epochs.

The principle of choosing batch size for different architectures mainly depends on their experimental performance. Specifically, we find that the convergence rate of the CNN-based architectures is fast on these two databases. In addition, as suggested in [45], large-batch methods tend to converge to sharp minimizers of the training and test functions, which notoriously lead to poor generalization. Based on experimental results, a medium batch size (i.e., 256) is used. The BNN-based method has a similar steady learning curve to the CNN-based method; however, due to the high dimensionality of the weight space in the BNN-based architectures incurring high computation cost, a smaller batch size (i.e., 40) is used. Compared to the CNN-based architectures, RNN-based architectures are generally harder to train. As illustrated in [46], a large batch size in RNNs with SGD can achieve more accurate gradient. Based on experimental results, a larger batch size (i.e., 512) is used to help RNN yield a smooth and stable learning curve. Overall, the best choice about the value of batch size used in our experiments is: 40 for the BNN-based methods, 256 for the CNN-based methods, and 512 for the RNN-based methods.

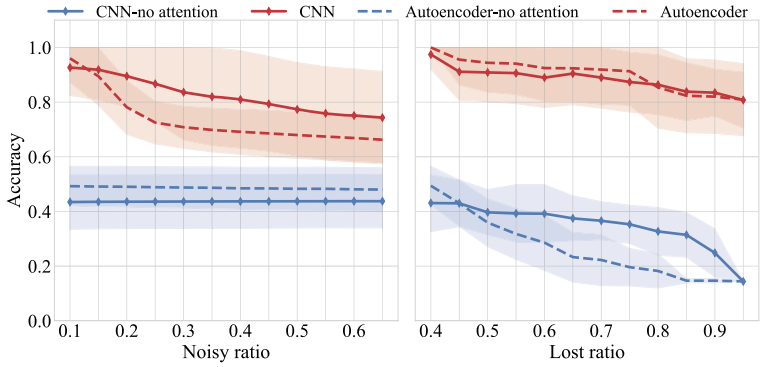


Fig. 6. Accuracy performance of two CNN-based models, i.e., vanilla CNN and auto-encoder, with and without the developed attention module in the BABNet on Database I. The left and right panels are regarding the noisy environment and lost pulse environment, respectively. It shows that, with the attention module (results in red color), the accuracy is much higher than the ones without involving the attention module (results in blue color).

Table 3
Quantitative comparison in terms of detection accuracy (*cf.* Fig. 6) about the CNN-based methods (i.e., vanilla CNN and CNN-encoder) with and without the developed attention module on databases I under the **noisy** pulse environment.

Noise ratio	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65
CNN (two layers)	0.927	**0.919**	**0.895**	**0.867**	**0.836**	**0.820**	**0.810**	**0.793**	**0.773**	**0.759**	**0.751**	**0.743**
Autoencoder	**0.960**	0.894	0.781	0.725	0.708	0.698	0.691	0.686	0.679	0.674	0.669	0.662
CNN (two layers)-no attention	0.434	0.435	0.436	0.436	0.436	0.436	0.437	0.437	0.437	0.438	0.437	0.437
Autoencoder-no attention	0.492	0.491	0.490	0.489	0.488	0.487	0.485	0.484	0.483	0.483	0.481	0.480

Table 4
Quantitative comparison in terms of detection accuracy (*cf.* Fig. 6) about the CNN-based methods (i.e., vanilla CNN and CNN-encoder) with and without the developed attention module on databases I under the **lost** pulse environment.

Lost ratio	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
CNN (two layers)	0.974	0.911	0.909	0.906	0.889	0.904	0.889	0.874	**0.863**	**0.838**	**0.834**	0.808
Autoencoder	**0.999**	**0.955**	**0.944**	**0.941**	**0.925**	**0.924**	**0.919**	**0.913**	0.854	0.823	0.820	**0.809**
CNN (two layers)-no attention	0.431	0.430	0.397	0.393	0.392	0.375	0.366	0.353	0.327	0.314	0.248	0.144
Autoencoder-no attention	0.494	0.430	0.359	0.318	0.286	0.233	0.222	0.196	0.182	0.147	0.147	0.144

4.3. Results and comparison

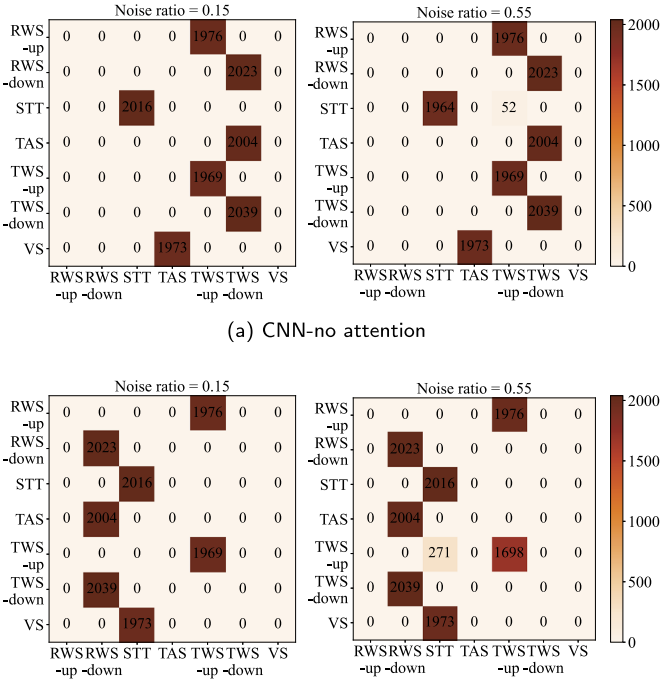
4.3.1. Validation of the attention module

We first validate the effectiveness of the developed attention module in our BABNet, see the middle of Fig. 2. Fig. 6, Tables 3 and 4 show the accuracy performance of two CNN-based models (i.e., vanilla CNN and auto-encoder) in processing the padded sequence data with and without involving the attention module. We remark that the vanilla CNN contains two convolutional layers, matching the auto-encoder (which also has two convolutional layers). The left panel in Fig. 6 shows that the test accuracy of the CNN and auto-encoder without the attention module on the noisy environment of Database I appears to be constant for varying noisy ratio. To investigate this further, Fig. 7 presents the confusion matrix about the classification performance corresponding to the left panel in Fig. 6. We see that both the CNN and auto-encoder without the attention module cannot distinguish "RWS-up" and "RWS-down" from "TWS-up" and "TWS-down", respectively, indicating that the dense noise corruption has no obvious impact. In general, Fig. 6, Tables 3 and 4 show that both the CNN and auto-encoder achieve similar test performance in both the noisy and lost pulse environments. It is evident that the introduction of the attention module can indeed greatly improve the test accuracy of both models on noisy and lost pulse environments. As we discussed before, padding elements may weaken the representation capacity of the convolutional operation. Fortunately, the developed attention module can re-weight the feature map of the padded data and can select important regions for the classification task through position information, which boosts the performance of the vanilla CNN and auto-encoder models.

The weighted sum of the attention module over the products of the hidden states and their corresponding attention values reduce the dimension of the feature vectors, and thus decrease the units of the subsequent FC layer as well as the number of model parameters. In detail, with the input signal size of $256 \times 8192 \times 3$ for database I (i.e., after padding and batch size set to 256), the number of the trainable parameters in the attention-based CNN and auto-encoder is 0.5K and 16K, respectively. However, without using the attention module, those numbers are increased by 1.83M and 1.85M, respectively, with no profits on test accuracy as shown in Fig. 6.

4.3.2. Comparison between RNN-based and CNN-based methods

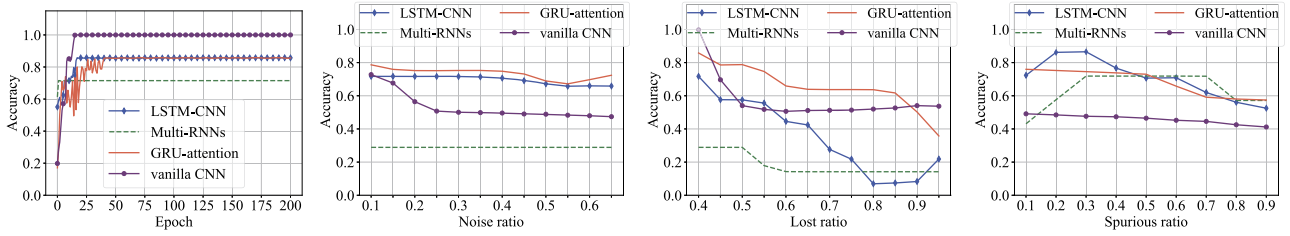
We now compare the vanilla CNN with several RNN methods, including GRU-attention, LSTM-CNN and multi-RNNs, which are designed for tasks involving sequential radar signal inputs. The results are given in Fig. 8. It is evident that all these models could reach to the convergence state less than 50 epochs, see Fig. 8 (a). However, only the vanilla CNN achieves nearly 100% training accuracy, compared to 71.5% for multi-RNNs and 85.5% for LSTM-CNN and GRU-attention. The comparison about the test accuracy performance regarding the noisy, lost pulse and spurious pulse environments (see Fig. 8 (b)–(d)) indicates that GRU-attention model outperforms the rest of methods in most cases. We can also observe that adding more GRU modules to multi-RNNs might not lead to better performance on the test data; moreover, it fails in the noisy environment (i.e., data with additive Gaussian measurement error), see Fig. 8 (b). The LSTM-CNN model achieves a moderate performance overall, but there exists a serious collapse in the lost pulse scenario, see Fig. 8 (c). In comparison, the vanilla CNN model with a single convolutional layer achieves almost the



(a) CNN-no attention

(b) Autoencoder-no attention

Fig. 7. The confusion matrix about the classification performance of the CNN and auto-encoder without attention module on the noisy environment of Database I.



(a) Convergence curves (b) Test on noisy data (c) Test on lost pulse (d) Test on spurious data

Fig. 8. Comparison between the CNN-based method (the vanilla CNN) and the RNN-based methods (i.e., GRU-attention, LSTM-CNN and multi-RNNs) on Database I. (a): training convergence analysis. (b)–(d): test accuracy comparison regarding the noisy, lost pulse and spurious pulse environments, respectively.

best robustness except for a sharp degradation in the lost pulse environment with lost ratio increasing from 40% to 50%.

It is observed that the test accuracy of Multi-RNN and LSTM-CNN rises slightly as the spurious ratio goes up at the beginning (see Fig. 8 (d) when the spurious ratio changes from 0.1 to 0.3). This accidental result may be due to the way of data generation. For the spurious pulse scenario, the parameters of pulses (e.g., time of arrival, RF and PW) are set with uniform distribution within the range of true values. Although the PRI component of the pulses is significantly broken (see Fig. 5), the RF and PW can still be informative for the recognition task. Therefore, in such a case, the spurious pulses may act like data augmentation, helping to enhance the recognition performance.

Now we investigate the test performance of the vanilla CNN with different number of convolutional layers. Let *CNN1*, *CNN3* and *CNN4* denote the vanilla CNNs containing one, three and four convolution layers, respectively. We remark that when the number of the convolutional layers larger than four, the classifiers become hard to converge for both databases due to the limited scale of the data sets. The number of filters of the first to the fourth convolutional layers of these CNNs is 32, 32, 64 and 64, respectively. Each model is trained for 10 times. Their detection performance with mean and standard deviation is shown in Fig. 9. It is evident that the model depth has positive effect on the model performance in all the data scenarios. Moreover, Fig. 9 also shows that deeper models (e.g. CNN3 and CNN4) can achieve higher mean accuracy, but their standard deviations are also larger (than e.g. CNN1).

Note that RNNs process one element of the input sequence at a time, i.e., not supporting parallel processing, and thus lack efficiency compared to CNNs. Furthermore, Transformers [47] and their variants, which are increasingly popular in recent years, overcome this issue and are able to model long time-dependencies. However, when the size of the training samples is limited and the models are shallow, their representation learning capabilities could be weakened [48]. For further experiments and comparison, we therefore focus more on CNN-based methods and compare them with the proposed BABNet.

4.3.3. Comparison between CNN-based and BNN-based methods

In this part we first adopt the well-designed auto-encoder with symmetrical convolutional and deconvolutional operations as described in Section 4.2, which is denoted as CNN-encoder here, and compare it and vanilla CNN with a series of BNN-based method, including BNN, BNN-prior and BNN-encoder. In particular, the BNN method is formed by placing constant weights in the vanilla CNN as probabilistic distribution, with only one convolutional layer. The *BNN-prior* method is formed by replacing the Gaussian prior in BNN by the pre-trained weights of CNN, and the *BNN-encoder*

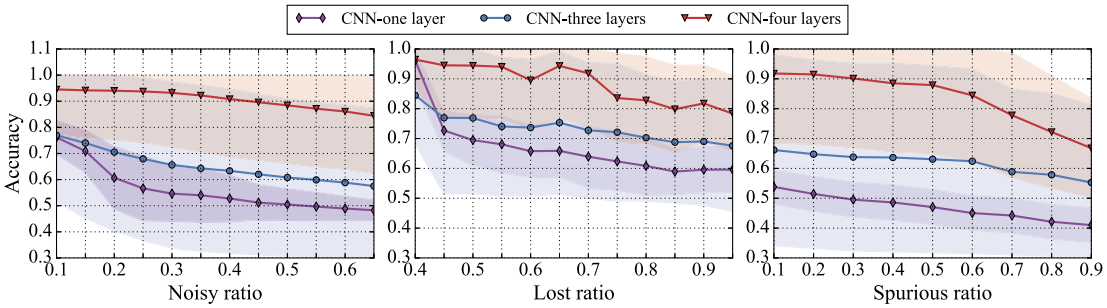


Fig. 9. Detection performance comparison between the vanilla CNNs (i.e., CNN1, CNN3 and CNN4) with different number of convolutional layers (i.e., 1, 3 and 4 layers) on Databases I under the noisy, lost pulse and spurious pulse environments.

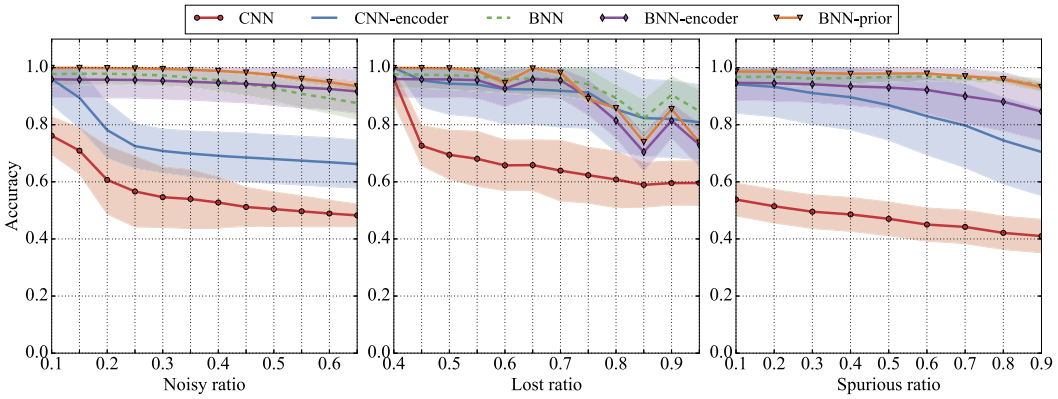


Fig. 10. Comparison between the CNN-based methods (i.e., vanilla CNN and CNN-encoder) and the BNN-based methods (i.e., BNN, BNN-prior and BNN-encoder) on databases I and II (i.e., figures (a) and (b), respectively). Left to right on each row represent the comparison results regarding test accuracy under the noisy, lost pulse and spurious pulse environments, respectively.

method is formed by modeling every filter weight in the auto-encoder as Bayesian probabilistic distribution. We train each model for 10 times on both databases I and II to eliminate random error, and test them under the noisy, lost pulse and spurious pulse environments.

Fig. 10 presents the test accuracy including the mean values and standard deviations. It is evident that BNN-based methods outperform the CNN-based methods in almost all the corrupted cases, demonstrating the superiority of the proposed BABNet. Quantitative comparison between the CNN-based methods and the BNN-based methods on databases I and II under several conditions is summarized in Table 5.

By comparing the test accuracy between two pairs of models, i.e., vanilla CNN vs. BNN and CNN-encoder vs. BNN-encoder in Table 5, we see the Bayesian backward propagation significantly enhances the robustness across almost all tasks (at most 39% on both databases). Except for the lost pulse scenario on database I, the BNN-prior method achieves the highest average test accuracy with the lowest fluctuation. Moreover, comparing with the BNN model, the BNN-prior model achieves further improvement regarding the average and the standard deviation of the test accuracy, validating the effectiveness of the suggested pre-trained process.

4.3.4. Comparison between Bayesian belief networks and BABNet

To further validate the recognition performance of the proposed BABNet, we below compare it with some of the state-of-the-art Bayesian belief networks, i.e., the ones in [34,36] denoted as *BBN*, on Database I. In particular, to stress the effectiveness of placing module weights as probabilistic distributions, the performance of

Table 5
Quantitative comparison between the CNN-based methods (i.e., vanilla CNN and CNN-encoder) and the BNN-based methods (i.e., BNN, BNN-prior and BNN-encoder) on databases I and II under the noisy, lost pulse and spurious pulse environments.

Data	Models	Noisy	Lost	Spurious
Database I	CNN	0.562 ± 0.120	0.669 ± 0.132	0.470 ± 0.071
	CNN-encoder	0.736 ± 0.127	0.902 ± 0.133	0.845 ± 0.151
	BNN	0.946 ± 0.056	$\mathbf{0.932 \pm 0.077}$	0.962 ± 0.025
	BNN-encoder	0.945 ± 0.069	0.886 ± 0.112	0.916 ± 0.088
	BNN-prior	$\mathbf{0.981 \pm 0.023}$	0.916 ± 0.096	$\mathbf{0.973 \pm 0.018}$
Database II	CNN	0.954 ± 0.103	0.431 ± 0.185	0.768 ± 0.208
	CNN-encoder	0.882 ± 0.199	0.844 ± 0.252	0.936 ± 0.140
	BNN	0.972 ± 0.051	0.707 ± 0.251	0.929 ± 0.084
	BNN-encoder	0.993 ± 0.019	0.721 ± 0.233	0.948 ± 0.059
	BNN-prior	$\mathbf{0.998 \pm 0.002}$	$\mathbf{0.964 \pm 0.088}$	$\mathbf{0.996 \pm 0.002}$

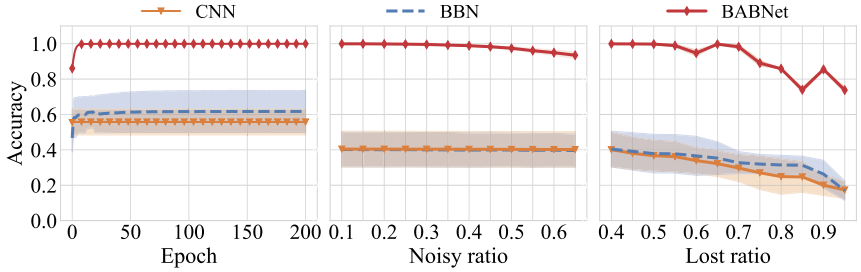


Fig. 11. Comparison between the vanilla CNN, BBN and the proposed BABNet on Database I in terms of the learning curves (left) and the test accuracy under the noisy (middle) and lost pulse (right) environments.

the vanilla CNN (without attention module) is presented as the baseline. There are two convolutional layers and one FC layer in this CNN. In BBN, all of these layers are replaced by Bayesian probabilistic counterparts with Gaussian prior (without attention module either). The proposed BABNet is formed by the attention module and the pre-trained CNN prior.

The comparison results are presented in Fig. 11 in terms of the learning curves (left panel) and the test accuracy (middle and right panels). In the training procedure and lost pulse scenario (i.e., left and right panels in Fig. 11), we see that there exists a slight yet sound performance improvement of BBN against the vanilla CNN, illustrating the superiority of treating weights as probabilistic distributions. Comparing with the proposed BABNet, we can clearly see the significant accuracy boost delivered by BABNet over BBN and the vanilla CNN with regard to the mean value and standard deviation in multiple repetitive experiments, demonstrating that the combination of the attention module and pre-trained CNN prior can indeed provide higher training accuracy, as well as more accurate and robust predictions on the corrupted data, e.g., noisy and lost pulse environments.

4.3.5. Robustness analysis

We now investigate the robustness of the CNN-based methods and the proposed BABNet in terms of the learning curve and loss landscape to further explore the key factors related to the robustness of deep models.

Fig. 12 presents the training accuracy of the vanilla CNN model and the BABNet on databases I and II and the test accuracy under the noisy, lost pulse and spurious pulse environments. It shows that the training performance of the vanilla CNN on both databases is close, i.e., it could converge after about 20 training epochs. On the other hand, it is also evident that its test accuracy encounters an obvious collapse and fluctuations on the non-ideal test environments. The early stopping mechanism may improve its generalization [49] but to a very limited extent as shown in Fig. 12. In contrast, Fig. 12 also shows that the proposed BABNet has far more robust performance on both databases comparing with the CNN-based method (i.e., vanilla CNN). Specifically, in database I, the vanilla CNN method suffers from almost 60% accuracy degradation among the training procedure and the non-ideal test conditions, yet our BABNet just encounters maximum 7.5% accuracy degradation. Analogously, in the worst scenario (lost pulse environment) in database II, the BABNet promotes the test accuracy of the vanilla CNN method up to 20%; moreover, the BABNet is capable of handling the measurement error case in database II and achieves test accuracy that is rather close to the training accuracy. Recall that the main difference between the two methods in Fig. 12 lies in the distribution over weights of each module and the backward propagation in the models. The introduction of uncertainty by the proposed BABNet helps deep models generalize well to unseen test data, as well as enhance their robustness.

The geometry of neural minimizers (i.e., their sharpness or flatness) could affect their robustness properties as illustrated in [50]. Thus, to capture and visualize the endogenous robustness properties of the proposed BABNet and its comparison to the CNN-based methods, we re-weight these models by filter-wise normalized directions proposed in [51] and plot the one-dimensional loss landscape. In detail, for the hyper parameter, i.e., the obtained weights $\boldsymbol{w}$ in a deep model, we parameterize a line from $\boldsymbol{w}$ to $\boldsymbol{w}'$ by choosing a scalar parameter α, and define the weighted average $\boldsymbol{w}(\alpha) = (1-\alpha)\boldsymbol{w} + \alpha\boldsymbol{w}'$. We then plot function $f(\boldsymbol{w}(\alpha))$ which denotes the output (i.e., classification error or accuracy here) of the re-weighted deep model with hyper parameter $\boldsymbol{w}(\alpha)$. The results on the two databases are depicted in Fig. 13. It is widely thought that "flat" minimizers, from the perspective of the curvature of the loss surface, are more robust compared to "sharp" ones since this implies a model is not sensitive to the perturbations of the network weights. Fig. 13 shows that the curve of the CNN-based methods (i.e., vanilla CNN and CNN-encoder) is rather sharper compared to those of the BNN-based methods (i.e., BNN and BNN-prior), implying that the CNN-based methods may have a catastrophic collapse on test accuracy as weights slightly change. Therefore, it also reveals that the BNN-based methods, i.e., the proposed BABNet, are able to find more robust minimizers with higher test accuracy compared to the CNN-based methods.

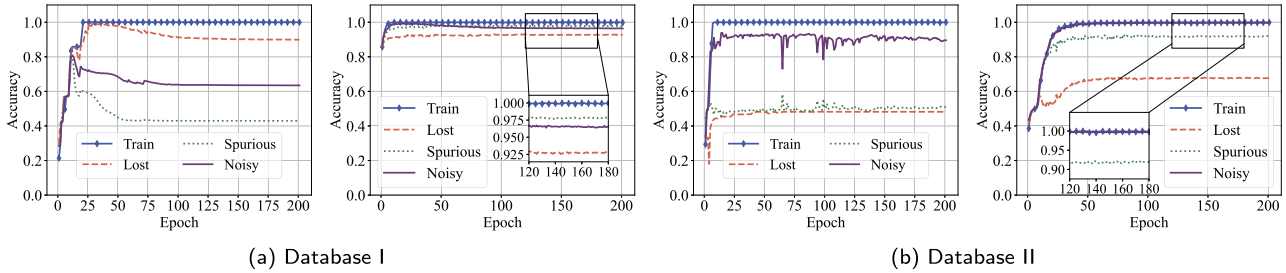


Fig. 12. Robustness comparison between the CNN-based method (i.e., vanilla CNN model) and our BABNet. Panels (a) and (b) present the training accuracy of both models respectively on databases I and II and its test accuracy under the noisy, lost pulse and spurious pulse environments. The left and right in each panel present the results of the vanilla CNN model and the proposed BABNet, respectively.

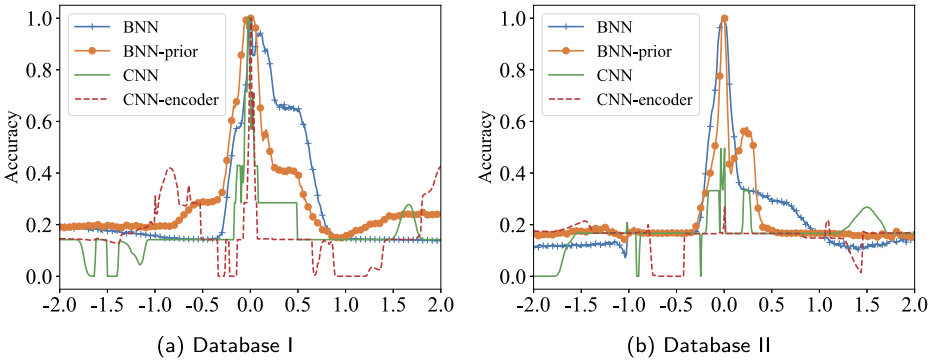


Fig. 13. Robustness comparison between the CNN-based methods (i.e., vanilla CNN and CNN-encoder) and the BNN-based methods (i.e., BNN and BNN-prior). Panels (a) and (b) present the validation accuracy landscape traversing from $-2\boldsymbol{w}$ to $2\boldsymbol{w}$ on databases I and II, respectively.

5. Conclusion

In this paper we developed a novel deep learning approach, BABNet, for radar work mode recognition from inter-pulse parameters. We showed that conventional CNNs can be reformulated into BNNs by modeling the weights of each module as probability distributions. The proposed approach, BABNet, learns a more robust representation of the sequential data with pre-trained CNNs as its prior. Besides, the combination of the padding operation and the attention mechanism enables the CNN, replacing RNN as a backbone, to effectively and efficiently process the sequential input data with variable length. Experimental results demonstrate that the proposed BABNet has a lightweight structure and can achieve superior performance than the state-of-the-art methods based on CNN, RNN or auto-encoder. Moreover, the proposed BABNet with well-designed prior possesses a flatter one-dimensional loss landscape, indicating its advantage in robustness as well. Potential future research could be focalized on further improving the recognition performance in the lost pulse situation by considering frameworks like few-shot learning with Bayesian inference.

CRediT authorship contribution statement

Mingyang Du: Conceptualization, Data curation, Formal analysis, Methodology, Resources, Validation, Writing – original draft, Writing – review & editing. **Ping Zhong:** Formal analysis, Funding acquisition, Validation. **Xiaohao Cai:** Validation, Writing – original draft, Writing – review & editing. **Daping Bi:** Project administration, Resources, Supervision, Validation. **Aiqi Jing:** Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

References

[1] O. León, J. Hernández-Serrano, M. Soriano, Securing cognitive radio networks, Int. J. Commun. Syst. 23 (5) (2010) 633–652.
[2] L. Sun, X. Wang, A. Yang, Z. Huang, Radio frequency fingerprint extraction based on multi-dimension approximate entropy, IEEE Signal Process. Lett. 27 (2020) 471–475.
[3] Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, O.A. Dobre, H.V. Poor, An efficient specific emitter identification method based on complex-valued neural networks and network compression, IEEE J. Sel. Areas Commun. 39 (8) (2021) 2305–2317.
[4] J. Dudczyk, A. Kawalec, Specific emitter identification based on graphical representation of the distribution of radar signal parameters, Bull. Pol. Acad. Sci., Tech. Sci. 63 (2) (2015) 391–396.
[5] K.I. Talbot, P.R. Duley, M.H. Hyatt, Specific emitter identification and verification, Technol. Rev. 11 (2003) 113–133.
[6] Z. Liu, Multi-feature fusion for specific emitter identification via deep ensemble learning, Digit. Signal Process. 110 (2021) 102939.
[7] J.-P. Kauppi, K. Martikainen, U. Ruotsalainen, Hierarchical classification of dynamically varying radar pulse repetition interval modulation patterns, Neural Netw. 23 (10) (2010) 1226–1237.
[8] C.-M. Lin, Y.-M. Chen, C.-S. Hsueh, A self-organizing interval type-2 fuzzy neural network for radar emitter identification, Int. J. Fuzzy Syst. 16 (1) (2014) 20–30.
[9] N. Visnevski, S. Haykin, V. Krishnamurthy, F.A. Dilkes, P. Lavoie, Hidden Markov Models for Radar Pulse Train Analysis in Electronic Warfare, IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, IEEE, 2005, pp. 597–600.
[10] J. Matuszewski, Specific emitter identification, in: International Radar Symposium, IEEE, 2008, pp. 1–4.
[11] M. Du, P. Zhong, X. Cai, D. Bi, Dncnet: deep radar signal denoising and recognition, IEEE Trans. Aerosp. Electron. Syst. 58 (4) (2022) 3549–3562, https://doi.org/10.1109/TAES.2022.3153756.
[12] S. D'Agostino, Specific emitter identification based on amplitude features, in: 2015 IEEE International Conference on Signal and Image Processing Applications, 2015, pp. 350–354.
[13] A. Aubry, A. Bazzoni, V. Carotenuto, A. De Maio, P. Failla, Cumulants-based radar specific emitter identification, in: 2011 IEEE International Workshop on Information Forensics and Security, 2011, pp. 1–6.

[14] M. Du, X. He, X. Cai, D. Bi, Balanced neural architecture search and its application in specific emitter identification, IEEE Trans. Signal Process. 69 (2021) 5051–5065.
[15] Z. Liu, S.Y. Philip, Classification, denoising, and deinterleaving of pulse streams with recurrent neural networks, IEEE Trans. Aerosp. Electron. Syst. 55 (4) (2018) 1624–1639.
[16] X. Li, Z. Liu, Z. Huang, W. Liu, Radar emitter classification with attention-based multi-rnns, IEEE Commun. Lett. 24 (9) (2020) 2000–2004.
[17] Y. Li, M. Zhu, Y. Ma, J. Yang, Work modes recognition and boundary identification of mfr pulse sequences with a hierarchical seq2seq lstm, IET Radar Sonar Navig. 14 (9) (2020) 1343–1353.
[18] X. Li, Z. Huang, F. Wang, X. Wang, T. Liu, Toward convolutional neural networks on pulse repetition interval modulation recognition, IEEE Commun. Lett. 22 (11) (2018) 2286–2289.
[19] S. Wei, Q. Qu, X. Zeng, J. Liang, J. Shi, X. Zhang, Self-attention bi-lstm networks for radar signal modulation recognition, IEEE Trans. Microw. Theory Tech. 69 (11) (2021) 5160–5172, https://doi.org/10.1109/TMTT.2021.3112199.
[20] G. Ruan, Y. Wang, S.L. Wang, Y. Zheng, Q. Guo, S. Shulga, Automatic recognition of radar signal types based on cnn-lstm, Telecommun. Radio Eng. 79 (4) (2020) 305–321.
[21] J. Cheng, R. Greiner, Learning bayesian belief network classifiers: algorithms and system, in: Conference of the Canadian Society for Computational Studies of Intelligence, Springer, 2001, pp. 141–151.
[22] Z. Yang, W. Qiu, H. Sun, A. Nallanathan, Robust radar emitter recognition based on the three-dimensional distribution feature and transfer learning, Sensors 16 (3) (2016) 289.
[23] Y. Gal, Uncertainty in deep learning, Ph.D. thesis, University of Cambridge, 2016.
[24] W. Zhu, J. Pelecanos, A bayesian attention neural network layer for speaker recognition, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 6241–6245.
[25] K. Miok, B. Škrlj, D. Zaharie, M. Robnik-Šikonja, To ban or not to ban: Bayesian attention networks for reliable hate speech detection, Cogn. Comput. 14 (1) (2022) 353–371.
[26] X. Fan, S. Zhang, B. Chen, M. Zhou, Bayesian attention modules, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 16362–16376.
[27] S. Zhang, X. Fan, B. Chen, M. Zhou, Bayesian attention belief networks, in: Proceedings of the 38th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, PMLR, vol. 139, 2021, pp. 12413–12426.
[28] E. Granger, M.A. Rubin, S. Grossberg, P. Lavoie, A what-and-where fusion neural network for recognition and tracking of multiple radar emitters, Neural Netw. 14 (3) (2001) 325–344.
[29] C.-S. Shieh, C.-T. Lin, A vector neural network for emitter identification, IEEE Trans. Antennas Propag. 50 (8) (2002) 1120–1127.
[30] J. Matuszewski, The analysis of modern radar signals parameters in electronic intelligence system, in: 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science, IEEE, 2016, pp. 298–302.
[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[32] E. Goan, C. Fookes, Bayesian neural networks: an introduction and survey, in: Case Studies in Applied Bayesian Data Science, Springer, 2020, pp. 45–87.
[33] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: representing model uncertainty in deep learning, in: International Conference on Machine Learning, in: PMLR, vol. 48, 2016, pp. 1050–1059.
[34] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural network, in: International Conference on Machine Learning, PMLR, 2015, pp. 1613–1622.
[35] A. Graves, Practical variational inference for neural networks, in: Advances in Neural Information Processing Systems, VOl. 24, Curran Associatesm Inc., 2011, pp. 2348–2356.
[36] K. Shridhar, F. Laumann, M. Liwicki, A comprehensive guide to bayesian convolutional neural network with variational inference, CoRR, arXiv:1901.02731 [abs], arXiv:1901.02731.
[37] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[38] V. Fortuin, Priors in bayesian deep learning: a review, Int. Stat. Rev. (2022) 1–29, https://doi.org/10.1111/insr.12502.
[39] W. Xiaofang, T. Zhongcheng, L. Jingxiu, S. Xiaowei, Investigation of aesa radar signal description and database design, Electron. Inf. Warf. Technol. 29 (4) (2014) 31–38.
[40] P. Lacomme, J.-C. Marchais, J.-P. Hardange, E. Normant, Air and Spaceborne Radar Systems: An Introduction, vol. 108, William Andrew, 2001.
[41] K. Chi, J. Shen, Y. Li, L. Wang, S. Wang, A novel segmentation approach for work mode boundary detection in mfr pulse sequence, Digit. Signal Process. 126 (2022) 103462, https://doi.org/10.1016/j.dsp.2022.103462.
[42] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (11) (2008) 2579–2605.
[43] X. Li, Z. Liu, Z. Huang, Denoising of radar pulse streams with autoencoders, IEEE Commun. Lett. 24 (4) (2020) 797–801.
[44] X. Li, Z. Liu, Z. Huang, Attention-based radar pri modulation recognition with recurrent neural networks, IEEE Access 8 (2020) 57426–57436.
[45] N.S. Keskar, J. Nocedal, P.T.P. Tang, D. Mudigere, M. Smelyanskiy, On large-batch training for deep learning: generalization gap and sharp minima, in: 5th International Conference on Learning Representations, 2017, pp. 1–16.
[46] M. Neishi, J. Sakuma, S. Tohda, S. Ishiwatari, N. Yoshinaga, M. Toyoda, A bag of useful tricks for practical neural machine translation: embedding layer initialization and large batch size, in: Proceedings of the 4th Workshop on Asian Translation (WAT2017), 2017, pp. 99–109.
[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 6000–6010.
[48] S. Khan, M. Naseer, M. Hayat, S.W. Zamir, F.S. Khan, M. Shah, Transformers in vision: a survey, ACM Computing Surveys, https://doi.org/10.1145/3505244.
[49] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning (still) requires rethinking generalization, Commun. ACM 64 (3) (2021) 107–115.
[50] I.J. Goodfellow, O. Vinyals, Qualitatively characterizing neural network optimization problems, in: International Conference on Learning Representations, 2015, pp. 1–20.
[51] H. Li, Z. Xu, G. Taylor, C. Studer, T. Goldstein, Visualizing the loss landscape of neural nets, in: Advances in Neural Information Processing Systems, vol. 31, Curran Associates, Inc., 2018, pp. 1–11.

Mingyang DU received his B. E. and M. E. from Electronic Engineering Institute and National University of Defense Technology in 2016 and 2018, respectively. He is currently a PhD student in the College of Electronic Engineering, National University of Defense Technology. Mr. Du's current research interests include signal processing and machine learning, focusing on radar emitter identification with deep learning.

Ping ZHONG (Senior Member, IEEE) received the M.S. degree in applied mathematics and the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2003 and 2008, respectively. Dr. Zhong was a recipient of the National Excellent Doctoral Dissertation Award of China in 2011 and the New Century Excellent Talents in the University of China in 2013. From March 2015 to February 2016, he was a visiting Scholar with the Department of Applied Mathematics and Theory Physics, University of Cambridge, Cambridge, U.K. He is currently a Professor with the National Key Laboratory of Science and Technology on ATR, NUDT. He has authored more than 40 peer-reviewed articles in international journals, such as the IEEE transactions and letters. His research interests include computer vision, machine learning and pattern recognition.

Xiaohao CAI is a Lecturer (Assistant Professor equivalent) in the School of Electronics and Computer Science at the University of Southampton. He received his PhD degree in mathematics from The Chinese University of Hong Kong in 2012. He afterwards was a Postdoctoral Researcher at the Department of Mathematics of the Technische Universitat Kaiserslautern in Germany. After that he was a Research Fellow (Wellcome Trust and Issac Newton Trust) affiliated with the Department of Plant Sciences and Department of Applied Mathematics and Theoretical Physics at the University of Cambridge. Thenceforth, before joining Southampton, he was a Research Fellow in the Mullard Space Science Laboratory (MSSL) at University College London (UCL). He is Fellow of Advance HE in the UK. He has served as a peer reviewer of over 50 international journals and has published over 40 peer reviewed papers in journals and conferences such as SIAM and IEEE transactions. He has broad multi-disciplinary research interests in applied mathematics, statistics, and computer science, with main focus and applications in image/signal/data processing, optimization, machine learning and computer vision.

Daping BI received his BSc and MSc degrees from the Electronic Engineering Institute in 1987 and 1990, respectively. He is currently a professor in College of Electronic Engineering, National University of Defense Technology. His research interests include radar signal processing, new technology of electronic countermeasures reconnaissance and interference.

Aiqi JING received her M.E. degree in Teaching English to Speakers of Other Languages from Sydney University in 2019. Currently, she is a teacher in Shanxi Datong University. Her research interests include second language learning and teaching.