A Radar Signal Deinterleaving Method Based on Semantic Segmentation with Neural Network

Wang Chao, Sun Liting, Liu Zhangmeng, and Huang Zhitao

Abstract—Radar signal deinterleaving is an important part of electronic reconnaissance. This study proposes a new radar signal deinterleaving method based on semantic segmentation, which we call “semantic segmentation deinterleaving” (SSD). We select representative sequence modeling neural network (NN) architectures and input the difference of time of arrival of the pulse stream into them. According to semantics contained in different radar signal types, each pulse in the pulse stream is marked according to the category of semantics contained, and radar signals are deinterleaved. Compared to the traditional deinterleaving method, the SSD method can adapt to complex pulse repetition interval (PRI) modulation environments without searching for the PRI or PRI period. Multiple rounds of search and merging operations are not required for radar signals with multiple pulses in a period. Compared to other deinterleaving methods based on NNs, the SSD method does not need to digitize the data and train a network for each target type. The SSD method also does not need to iterate input and output data. The proposed method has high robustness to pulse loss and noise pulses. This research also shows that recurrent NNs still have more advantages than convolutional NNs in this sequence modeling task.

Index Terms—Bidirectional gated recurrent unit (BGRU), Bidirectional long short-term memory (BLSTM), Difference of time of arrival (DTOA), Dilated convolutional network (DCN), Radar signal deinterleaving, Semantic segmentation.

I. INTRODUCTION

In ELECTRONIC warfare, to obtain information about a target radar, it is necessary to use electronic reconnaissance equipment for reconnaissance and interception of the corresponding target signal. The process involves detection of signals first [1], [2], and then analysis of the detected signals. In an actual electromagnetic environment, there are often other electromagnetic signals besides the target radar signal. In that case, data collected by electronic reconnaissance equipment may contain information from different targets, and the intercepted pulse stream may also contain interleaved pulses from different radiation sources, as shown in Fig. 1. Full pulse data (FPD) are pulse description words for each pulse output in chronological order in electronic reconnaissance equipment. Pulse description words include the time of arrival (TOA), direction of arrival (DOA), pulse width (PW), radio frequency (RF), pulse amplitude (PA), and other information about each pulse. Radar signal deinterleaving, which is an important part of electronic reconnaissance, is essentially to deinterleave the interleaved pulse description words belonging to different radiation sources in the FPD.

Semantic segmentation is an important task in image processing. It uses feature information about different categories of targets, namely semantics, to mark each pixel in an image according to the category of the target the pixel belongs to. This method can segment different target types in an image and is also known as dense prediction [3]. In recent years, neural networks (NNs) have become the most important tool and a topic of active research in image semantic segmentation.

In this paper, a new radar signal deinterleaving method based on semantic segmentation is applied to radar signal deinterleaving, which we call “semantic segmentation deinterleaving” (SSD). In this method, the radar signal deinterleaving problem is regarded as a semantic segmentation problem. According to this concept, pulses belonging to the same radar constitute the corresponding semantics. For a pulse stream containing multiple radars, pulses are classified according to the corresponding semantics, and radar signal deinterleaving is realized. In this method, deinterleaving is transformed into a mapping operation with input dimensions equal to output dimensions, making it easy to understand and manipulate. This method can realize signal deinterleaving of multiple radar targets through...
one network and one step, as shown in Fig. 2. It is innovative and has significant advantages over other methods.

Compared to the traditional methods [4], [5], [6], [7], [8], [9], [10], the SSD method does not need to find pulse repetition interval (PRI) or PRI period first and then conduct a sequence search so it can adapt to complex PRI modulation environments. It does not need to iterate over data. For radar signals with multiple pulses in one period, multiple rounds of search and merging operations are not required. After deinterleaving with the SSD method, the PRI modulation mode of the target radar signal is known, and PRI modulation mode recognition is no longer needed.

Compared to other deinterleaving methods based on NNs and automata [11], [12], [13], [14], the SSD method changes the way data is used. It does not require data to be digitized, avoiding the resolution problem. This method can output multiple targets in one step with one network, without training a network for each target category and without iterating input and output data repeatedly. Only the PRI modulation modes and parameter range are required for training without very accurate PRI values as prior information in this method.

This paper is organized as follows. Section II introduces literature on radar signal deinterleaving and NNs for image semantic segmentation and sequence modeling. The characteristics and data model of this task are discussed in Section III. Based on the analysis presented in Sections II and III, we determine the NN architectures and deinterleaving strategy adopted by the proposed method in Section IV. Section V presents the experimental results and analysis. Section VI concludes the entire research.

II. RELATED WORK

A. Radar Signal Deinterleaving

The research of radar signal deinterleaving can be divided into two categories: deinterleaving based on multi-parameter and deinterleaving based on arrival time information. The former makes comprehensive use of the TOA, DOA, PW, RF, and PA, while the latter uses only the TOA. This paper studies the deinterleaving method based on arrival time information.

The radar PRI is the interval between fronts of adjacent pulses when the radar transmits signals. In the deinterleaving methods based on TOA, an important concept is to use the periodicity of radar PRI. This concept first finds the radar PRI or PRI period from information about the difference of TOA (DTOA) of the pulse stream and then uses the found PRI or PRI period to search for the target radar pulses from the pulse stream, as shown in Fig. 3 [4], [5], [6], [7], [15]. Some methods use histograms of DTOA to obtain radar PRI, such as cumulant difference histogram (CDIF) [4] and sequential difference histogram (SDIF) [5]. The latter regards pulse stream as a Poisson process, which significantly reduces the computational cost compared with the former and is widely used in practice. Another method involves obtaining the spectrum of PRI by the transformation of arrival time and then extracting the real PRI [6], [7]. The PRI transform algorithm (PRI-Tran) has attracted considerable attention because of its excellent harmonic suppression performance. The improved PRI-Tran algorithm adapts to jittered PRI better than the original version [6]. In the method based on the discrete Fourier transform algorithm, a spectrum’s maximum peak is the pulse stream frequency, but it is not accurate for a staggered pulse stream [7]. The CDIF algorithm has been improved in [10]. It optimizes the estimated PRI values and extracts pulse pairs on the basis of pulse correlation directly instead of searching for pulses. Tao et al. proposed a correlation matching method (CMM) in [16] that can be used for deinterleaving staggered PRI signals but cannot deinterleave mixed pulse streams with other PRI-modulated signals. Cheng et al. combined the CMM algorithm and PRI-Tran method to deinterleave mixed pulse streams, but the robustness is unsatisfactory [17]. In [18], an improved SDIF algorithm using the clustering algorithm and PRI-Tran was proposed to improve the robustness.

The above methods have several shortcomings. First, when radar signal pulses are dense or the target pulse loss rate is high, it is easy to find the wrong PRI. Second, when pulses belonging to different targets approach, it is difficult to distinguish them accurately. Third, when searching for potential PRI and target pulses, it is necessary to set thresholds and tolerances based on experience, making the deinterleaving effect prone to large fluctuations. Moreover, the threshold is very sensitive to the data quality, and the deinterleaving effect may fluctuate significantly when the data quality changes. Fourth, these methods require...
accuracy of time information [11], [12], [13]; second, when there is more than one pulse in a time unit, only one pulse is presented, and information about other pulses is covered [11], [12], [13]. Third, the methods based on this operation may mark a position with no pulse as having one [12], [13]. In addition, the above method needs to train a network for each category of radar signal, and only one target can be deinterleaved in each output step, as shown in Fig. 4. In other words, the existing method completes a binary classification task in each output step, so it is necessary to iterate input and output data repeatedly.

Authors of [14] introduced finite automata to treat the deinterleaving problem as symbol string recognition and concentrated on meeting the requirement of online deinterleaving. But they also require a lot of prior information.

B. NNs for Semantic Segmentation and Sequence Modeling

In recent years, the application of NNs in the field of image semantic segmentation has been thoroughly studied. In [19], a “fully convolutional network” was first used for semantic segmentation and achieved excellent effectiveness. A generic “multipath refinement network” was proposed in [20] that explicitly exploits all the information available along the downsampling process to enable high-resolution prediction using long-range residual connections. Reference [21] modified “U-Net architecture” to allow the network to propagate context information to higher resolution layers. As a consequence, it works with very few training images and yields more precise segmentation. In [3], [22], an “encoder–decoder architecture” consisting of an encoder network and a corresponding decoder network was used for segmentation. The decoder upsamples its lower resolution input feature maps. Some researchers improve the localization of object boundaries by using a “fully connected conditional random field” [23]. Reference [24] propose “atrous spatial pyramid pooling” to robustly segment objects at multiple scales. These researches have achieved good results in image semantic segmentation, but they cannot be directly used in the task of this paper and can only provide some references.

The RNN is the most popular sequence modeling architecture so far and has been considered the best architecture for a long time. In 1997, Jurgen Schmidhuber proposed the long short-term memory (LSTM) architecture [25], which uses a gated unit and memory mechanism to improve RNNs in training. In the same year, Mike Schuster proposed a bidirectional RNN model (BRNN) [26], which enables RNNs to simultaneously use sequence information in both forward and backward directions. The development of the gated recurrent unit (GRU) further improved the training problem of RNNs [27]. The application of the RNN encoder-decoder effectively solves the sequence to sequence (seq2seq) problem [28], [29]. The introduction of the attention-based models significantly improves the performance of the RNN-based model on many tasks [30], [31]. The use of transformer models pushes attention-based models to new heights while abandoning the RNN architecture [32]. In [33], the authors presented a simple regularization technique for RNNs with LSTM units and demonstrated how to accurately apply dropout to LSTMs. This substantially reduced overfitting on a
A variety of tasks. Results in [34] show that partitioning hidden layers under distinct temporal constraints enables the learning of multiple timescales. Reference [35] introduced “grid long short-term memory,” a network of LSTM cells arranged in a multidimensional grid that can be applied to vectors, sequences, or high-dimensional data such as images. The network provides a unified way of using LSTM for both deep and sequential computation. The authors of [36] introduced a novel theoretical analysis method of recurrent networks based on Gersgorin’s circle theorem and proposed “recurrent highway networks,” which extended the LSTM architecture to allow step-to-step transition depths larger than one. Reference [37] presented a novel RNN-based model that combined the remembrance ability of unitary RNNs with the ability of gated RNNs to effectively forget redundant/irrelevant information in the model’s memory.

In recent years, CNN-based models have also performed excellently in some sequence modeling tasks, including audio synthesis, word-level language modeling, and machine translation, and can achieve state-of-the-art performance in some tasks [38], [39], [40], [41], [42]. In [38], the authors developed a finite context approach through stacked convolutions, which can be more efficient because they allow parallelization over sequential tokens. Reference [39] presented a faster and simpler architecture based on a succession of convolutional layers. An architecture that is entirely convolutional and equipped with gated linear units and residual connections has been proposed for the sequence to sequence modeling [40]. The researchers of [41] introduced ByteNet, which has linear running time, decouples translation from memorization, and has short signal propagation paths for tokens in sequences. Reference [42] introduced a deep NN named WaveNet for generating raw audio waveforms. They developed new architectures on the basis of dilated causal convolutions, which exhibit very large receptive fields.

These results prompt researchers to ponder: can the CNN architecture outperform the RNN architecture in more tasks, or is it simply limited to some specific tasks? Shaojie Bai et al. conducted an empirical study on this question [43] and proved that the CNN architecture outperformed the RNN architecture in many sequence modeling tasks, while these tasks are on the RNN’s “home Turf.” The authors summarize this CNN architecture as a temporal convolutional network (TCN) [43].

III. TASK CHARACTERISTICS AND DATA MODEL

A. Input and Output Form

For the input data to have a smaller variance and facilitate NN processing, the proposed method inputs DTOA of the pulse stream into the NN instead of TOA. When the pulse stream contains only a single radar target, the DTOA of the pulse stream is the real PRI of the target. When the pulse stream contains pulses from multiple radiation sources or pulses from a single target arrive through multiple paths, the DTOA of the pulse stream is chaotic. To make DTOA and TOA equal in length, we add 0 before DTOA as the first value of DTOA. The output of the proposed method is the label information about the category of each pulse. When training the NN, the DTOA and pulse labels to input are into the network, as shown in Fig. 1.

Unlike existing deinterleaving methods based on NNs [11], [12], [13], the proposed method does not digitize time information as input to avoid adverse effects caused by it. In the output, each pulse is judged to determine the target category to which it belongs, rather than whether it belongs to the target we want. In other words, the processing result of the proposed method is multi-classification, whereas that of existing methods is typically binary classification.

B. Difference Between This Task and Image Semantic Segmentation and Other Sequence Modeling Tasks

Radar signal deinterleaving based on semantic segmentation is a problem of mapping an input sequence to an output sequence. It is different from image semantic segmentation and seq2seq tasks such as natural language processing (NLP).

1) Target Points are Unconcentrated and Throughout the Sequence: In the pulse stream, the pulses of different targets are interleaved, and information about the same target runs through the entire pulse stream. However, in image semantic segmentation, the pixels of the same object are typically concentrated in one or several regions.

2) There is a Strict Mathematical Relationship Between the Data at Each Input Point of the Sequence: Since the input is DTOA, the information loss of one data point completely changes the information in the pulse stream, so pooling is not allowed. Image semantic segmentation and seq2seq tasks do not have this feature.

3) The Input and Output are of Equal Length: The input and output of this task are equal length sequences. In some sequence modeling tasks such as machine translation, the input and output are often not of equal length.

4) The Input at Each Point is Meaningless on its own: Similar to image data, the input data of each point in this task are meaningless alone. Only when they are computed along with other input data can their information be reflected. However, in NLP tasks, each input word has a specific meaning.

5) Forward and Backward Information is Equivalent: In the deinterleaving task, the forward and reverse information about a sequence is completely equivalent, which is significantly different from many sequence modeling tasks. Therefore, in this task, using bidirectional information about the sequence simultaneously is more conducive to accurately judging the category of each pulse.

C. Limitations of the Deinterleaving Method Based on Semantic Segmentation and the Solution

The deinterleaving method based on semantic segmentation faces the same problem as image semantic segmentation, i.e., it cannot distinguish multiple objects belonging to the same category in a single input. In image processing, this problem is solved by instance segmentation [44], [45], [46], which typically consists of two contents: semantic segmentation and object detection, as shown in Fig. 6 [44]. However, this solution cannot be applied to radar signal deinterleaving because, in an image, pixels belonging to the same target are concentrated, whereas in a pulse stream, pulses belonging to the same target are not.
Fig. 6. Image instance segmentation.

Fig. 7. DTOA of radar signal with constant PRI.

Fig. 8. DTOA of radar signal with D&S PRI.

Fig. 9. DTOA of radar signal with staggered PRI.

Pulses from one target are interleaved with pulses from other targets and distributed in the entire pulse stream. The method to solve this problem is to extract more different semantics from target radar signals and divide radar signals into more categories. This will be discussed in Section IV.

D. Data Model

PRI modulation mode has an important effect on radar function and performance. In this paper, we define and use three PRI modulation modes in simulation experiments. The DTOA of radar signals with these three PRI modulation modes is shown in Figs. 7–9. The subfigures show the DTOA of radar signals under different conditions: a) DTOA of nondestructive radar signal, i.e., PRI of radar signal; b) DTOA of radar signal with pulse loss; c) DTOA of radar signal with random noise pulses; and d) DTOA of radar signal with pulse loss and random noise pulses.

1) Constant PRI: The radar PRI remains a constant, and the PRI sequence can be represented as

\[ PRI_n = PRI_0, n = 1, 2, 3, \ldots \]  

(1)

2) Dwell and Switch (D&S) PRI: Radar PRI changes in groups, with the same number of pulses in each group. The value of PRI changes periodically between groups. Its mathematical model is expressed as

\[ PRI_n = PRI_{n+j}, 0 \leq j < J, \]  

(2)

\[ PRI_n = PRI_{n+J+K}. \]  

(3)

\( PRI_n \) is the first PRI in each group, \( J \) represents the number of pulses in each group, and \( K \) represents the number of pulse groups in one period, i.e., the number of PRI values in a period.

3) Staggered PRI: Radar PRI consists of several fixed values and changes periodically. The PRI sequence can be described as follows:

\[ PRI_n = PRI_{n+M}. \]  

(4)

\( M \) represents the number of PRI values in a period.
When for DTOA data, forward and reverse information was highlighted in Part C of Section III that owing to TCN's [43] outstanding performance in BLSTM and BGRU each convolution step, both sides of the feature map are padded is that, to keep the length of the feature map unchanged in section, a DCN is generally the same as a TCN. The only difference when constructing a DCN. In terms of residual block construction, sequence modeling tasks, this study used TCN as a reference to achieve this task. The parameters of BLSTM and BGRU are RNN, i.e., bidirectional GRU (BGRU) and LSTM (BLSTM), put of each BRNN's step is connected with the fully connected layer to realize the classification of each time step, as shown in Fig. 11. This operation considers the equivalence of forward and reverse information. In this study, we set the convolution kernel size of the DCN as 3 and 8 residual blocks. This makes the receptive field of each convolution kernel in the last layer of the network sufficient to cover the length of the input data. After the residual block, we use a common convolution to reduce the number of channels to the number of target categories to achieve the classification of each pulse, as shown in Fig. 11. The parameters of DCN are shown in Table II.

### IV. NN Architecture and Deinterleaving Strategy

#### A. What Kind of NN Should be Selected for This Task

According to the concept of the proposed method, we select a NN, and it is required to have good semantic segmentation ability for sequence data with strong mathematical relations. In Section III, we analyzed the difference between this task and image semantic segmentation and seq2seq tasks such as NLP.

Therefore, the NN used in the proposed method needs to meet the following requirements: good ability at sequence modeling, full use of all information about the entire sequence (or a sufficiently large receptive field), equal length of input and output, and no pooling.

Accordingly, we select the BRNN and dilated convolutional network (DCN) but ignore the classical NNs used for image semantic segmentation [3], [19], [20], [21], [22], [23], [24], [47], and used for seq2seq tasks, [28], [29], [30], [31], [32], e.g. encoder–decoder architecture.

1) **BRNN:** For DTOA data, forward and reverse information is equivalent. To make full use of the complete information about the sequence when determining the category of each pulse, the BRNN is used in this study to process DTOA data. Then, the output of each BRNN's step is connected with the fully connected layer to realize the classification of each step, as shown in Fig. 10. This study uses the LSTM and GRU architectures of RNN, i.e., bidirectional GRU (BGRU) and LSTM (BLSTM), to achieve this task. The parameters of BLSTM and BGRU are presented in Table I.

2) **DCN:** Owing to TCN’s [43] outstanding performance in sequence modeling tasks, this study used TCN as a reference when constructing a DCN. In terms of residual block construction, a DCN is generally the same as a TCN. The only difference is that, to keep the length of the feature map unchanged in each convolution step, both sides of the feature map are padded symmetrically rather than adopting causal convolution, as shown in Fig. 11. This operation considers the equivalence of forward and reverse information. In this study, we set the convolution kernel size of the DCN as 3 and 8 residual blocks. This makes the receptive field of each convolution kernel in the last layer of the network sufficient to cover the length of the input data. After the residual block, we use a common convolution to reduce the number of channels to the number of target categories to achieve the classification of each pulse, as shown in Fig. 11. The parameters of DCN are shown in Table II.

#### B. How to use Semantic Information–Deinterleaving Strategy

Semantic segmentation is based on the characteristics of different categories of objects. Analysis in Section III indicates that sufficient different semantics need to be extracted from radar signals to achieve a good deinterleaving effect. This part mainly analyzes how to deinterleave radar signals using PRI modulation modes and PRI parameters as semantics separately and how to use both for deinterleaving comprehensively.

1) **Take PRI Modulation Modes as Semantic Information:** Different PRI modulation modes represent different information, i.e., different semantics, as shown in Table III. The target radar signals can be divided into different categories accordingly. When the intercepted radar pulse stream contains multiple targets with different PRI modulation modes, PRI modulation information can be used as semantic information, and the category of each pulse can be predicted on the basis of this to achieve radar signal deinterleaving.

2) **Take PRI Parameters as Semantic Information:** When multiple target radars adopt the same PRI modulation mode, the deinterleaving method based on semantic segmentation with PRI modulation modes is limited and cannot distinguish such multiple targets. In this case, PRI parameter information can be used as semantics to distinguish different targets. That is, the radar signal with PRI value located in (a, b) is the first subclass, the radar signal located in (b, c) is the second subclass, the radar signal located in (c, d) is the third subclass, and so on, as shown in Table III.

When the number of subclasses increases, to ensure high deinterleaving efficiency, it is necessary to increase the model size of the NN. If the PRI range is too large, multiple smaller NNs can be trained at different locations to regulate the model size, as shown in Table IV. How to set the specific value range of the subclass depends on the specific task environment.

3) **Comprehensive use of PRI Modulation Modes and PRI Parameters:** It was highlighted in Part C of Section III that we need sufficient semantics to divide radar pulses into different categories to solve the problem of having multiple targets in the same category. Here, we have proposed deinterleaving methods using PRI modulation and PRI parameters as semantic information. In this section, two methods are proposed to make comprehensive use of this information for deinterleaving. One is a parallel deinterleaving method, using PRI modulation modes and PRI parameters simultaneously, as shown in Fig. 12. The second is the serial deinterleaving method, which first uses PRI modulation and then uses PRI parameters for deinterleaving, as

---

**TABLE I**

| NN     | Input size | Hidden layer size | Number of hidden layers | Output size |
|--------|------------|-------------------|-------------------------|-------------|
| BLSTM  | 1          | 104               | 3                       | number of categories |
| BGRU   | 1          | 120               | 3                       | number of categories |

---

**Fig. 10. Deinterleaving process of BRNN.**
shown in Fig. 13. The former categorizes PRI modulation modes and PRI parameters simultaneously. It has fewer deinterleaving steps and needs only one NN. The latter first categorizes PRI modulation modes and then PRI parameters. It needs to be completed step by step and uses multiple NNs. However, the latter can adapt to more complex deinterleaving environments, e.g., when the semantic categories of radar pulses are diverse and the capacity of NNs is limited.

C. Loss Function

In this task, each sample input to the NN is the DTOA of a pulse stream. The predicted loss of each sample by the NN is the average of the predicted loss of all pulses in the pulse stream, i.e.,

\[
Loss = \frac{1}{N} \sum_{n=1}^{N} loss_n. \tag{5}
\]

\(loss_n\) is the predicted loss of the nth pulse by the NN. \(N\) represents the length of the pulse stream, which is the number of pulses in the pulse stream. We use a cross-entropy loss function to evaluate the prediction performance of each pulse of the NN, which can be described below:

\[
loss = -\sum_{c=1}^{C} P_c \log \left( \hat{P}_c \right). \tag{6}
\]

\(C\) represents the category number of radar signals in the pulse stream. \(P_c\) denotes whether the current pulse belongs to the cth radar signal category, and its value is either 0 or 1. \(\hat{P}_c\) represents
the probability that the current pulse belongs to the cth radar signal category in the NN’s prediction.

V. EXPERIMENTS

A. Data Simulation

According to the definition of PRI modulation modes in Part D of Section III, the following designs are developed for simulation data.

1) For all PRI modulation modes, the PRI value satisfies the condition $20 < PRI < 100$ unless we specify it.

2) For the D&S PRI, the number of pulses in each group satisfies the condition $4 \leq J \leq 6$, and the number of pulse groups in one period satisfies the condition $4 \leq K \leq 6$.

3) For the staggered PRI, the number of PRI values in a period satisfies the condition $3 \leq M \leq 10$ unless we specify it.

4) In this paper, the Gaussian distributed deviation is added to TOA to simulate measurement errors, and the standard deviation is 0.1. Then, the DTOA is generated on this basis, and the length of the DTOA is 1,000.

5) In this paper, the problem of target pulse loss and random noise pulses in intercepted pulse stream is considered. The pulse loss rate of the target is represented by $\rho_l$, and the ratio of the number of random noise pulses to the average number of the target radars pulses in the intercepted pulse stream is represented by $\rho_n$. The proportion of the number of random noise pulses to the total number of pulses can be calculated by $\frac{\rho_n}{\rho_n + D}$, and $D$ represents the number of target radars.

B. Design of Experiments

According to the deinterleaving strategy proposed in Section IV, five experiments are designed to verify the feasibility of the proposed deinterleaving method and compare the performance of different NNs in addressing this problem.

Experiment 1 verifies the feasibility of deinterleaving radar signals taking PRI modulation modes as semantics. Experiments 2 and 3 verify the feasibility of deinterleaving radar signals taking PRI parameters as semantics. Experiment 4 verifies the parallel deinterleaving method using PRI modulation modes and PRI parameters simultaneously. Experiment 5 is used to verify the first step of the serial deinterleaving method, deinterleaving radar signals with PRI modulation modes when there are multiple targets per PRI modulation mode. Experiments 2 and 3 verify the second step. $\rho_l$ and $\rho_n$ of each sample of the training data are randomly chosen within a certain range.

1) Experiment 1 ‘Deinterleaving Radar Signals with PRI Modulation Modes: The target settings are shown in Table V. For the training data, $0 < \rho_l < 0.25$, $0 < \rho_n < 0.25$. 

![Parallel deinterleaving method](image-url)
2) **Experiment 2–Deinterleaving Radar Signals of Constant PRI with PRI Values:** The target settings are shown in Table VI. For the training data, $0 < \rho_l < 0.5$, $0 < \rho_n < 0.5$.

3) **Experiment 3–Deinterleaving Radar Signals of Staggered PRI with PRI Values:** The target settings are shown in Table VII. For the training data, $0 < \rho_l < 0.5$, $0 < \rho_n < 0.5$.

4) **Experiment 4–Deinterleaving Radar Signals Using PRI Modulation Modes and PRI Parameters Simultaneously:** The target settings are shown in Table VIII. For the training data, $0 < \rho_l < 0.5$, $0 < \rho_n < 0.5$.

5) **Experiment 5–Deinterleaving Radar Signals with PRI Modulation Modes When There are Multiple Targets per PRI Mode:** The target settings are shown in Table IX. For the training data, $0 < \rho_l < 0.25$, $0 < \rho_n < 0.25$. 
C. Datasets, Training Methods, Control Experiments, and Performance Metrics

1) Datasets: For each experiment, there are two types of datasets. The first type of dataset is a mixed dataset, and there is only one. For each sample in the mixed dataset, $\rho_1$ and $\rho_n$ are randomly obtained within the set range, and the total number of samples is 81,920. The mixed dataset is used to train the model and test the overall performance of the NNs. The second type is the specified conditional datasets, and there are 16 specified conditional datasets. For each sample in the specified conditional datasets, $\rho_1$ and $\rho_n$ are specified, and the number of samples per dataset is 8,192. The 16 specified conditional datasets were generated under four kinds of conditions: a) there is pulse loss but no random noise pulse, i.e., $\rho_n = 0$, and there are 5 datasets; b) there is no pulse loss, i.e., $\rho_1 = 0$, but there are random noise pulses, and there are 5 datasets; c) There are both pulse loss and random noise pulses, and $\rho_1 = \rho_n$, and there are 5 datasets; d) There is no pulse loss or random noise pulse, i.e., $\rho_1 = \rho_n = 0$, and there is 1 dataset.

2) Training Methods for NNs:
   a) Training, Validation and Testing Methods: We randomly divided the mixed dataset into training, validation, and test sets in a ratio of 8:1:1. The NNs are optimized according to the results of the training and validation sets. We tested the trained model on the test set to obtain the overall performance of the network and on the specified conditional dataset to obtain the model’s performance under different conditions.
   b) Learning Rate (LR): The initial LR of BLSTM and BGRU is 0.0001, and that of DCN is 0.001. When the network performance does not improve at the current LR, the LR is reduced to 1/10 of the current LR.
   c) Optimization Algorithm: Adam optimizer.
   d) Epoch Number: We did not specify the number of epochs for training but ended the training when the performance on the validation set was no longer improving.
   e) Software Environment: Windows sever 2012 R2 standard, python 3.8.0 64 bit, torch 1.6.0.

3) Control Experiments: In this paper, we select SDIF and PRI-Tran algorithms as control experiments. When searching for PRI, we set the threshold according to the algorithm provided in the corresponding literature. When searching for pulses with the found PRI, we take the first pulse of the pulse stream as the starting point and calculate the number of pulses located at integer times of the found PRI. If the number exceeds the threshold, these pulses are considered target pulses. If the search with the first pulse as the starting point fails, the next pulse is taken as the starting point, and so on. Pulse search threshold $TH$ is calculated as follows:

$$TH = C_0 \times \frac{T_0}{PRI_0}$$  \hspace{1cm} (7)

where $C_0$ is the coefficient, $T_0$ represents the duration from the starting point to the last pulse, and $PRI_0$ represents the found PRI value. Multiple parameters need to be determined when setting thresholds for searching for PRI and pulses. These parameters need to be adjusted according to the corresponding working environment to achieve a relatively good deinterleaving efficiency. We repeated experiments 1–4 many times using mixed datasets and selected the optimal parameters according to the experimental results. For experiments using the specified conditional datasets, the same parameters are used as in the experiments using the corresponding mixed datasets.

4) Performance Metrics: In this paper, we use accuracy, mean recall (MR), mean precision (MP), and mean intersection over union (MIOU) as performance metrics. For the cth category of pulses in pulse streams, according to the corresponding relationship between their true labels and predicted labels, we can divide them into four cases:

a) The true label is $c$, the prediction label is $c$, and the number of corresponding pulses is represented by $TP_c$.

b) The true label is $c$, the prediction label is not $c$, and the number of corresponding pulses is represented by $FN_c$.

c) The true label is not $c$, the prediction label is $c$, and the number of corresponding pulses is represented by $FP_c$.

d) The true label is not $c$, the prediction label is not $c$, and the number of corresponding pulses is represented by $TN_c$.

The confusion matrix formed by these four cases is shown in the Table X.

Accuracy is used to measure the proportion of correctly predicted pulses to all pulses, and it can be expressed as:

$$accuracy = \frac{\sum_{c=1}^{C} TP_c}{NUM}.$$ \hspace{1cm} (8)

$NUM$ represents the total number of pulses.

MR is used to measure the proportion of predicted positive cases among true positive cases, and it can be expressed as:

$$MR = \frac{1}{C} \sum_{c=1}^{C} \frac{TP_c}{TP_c + FN_c}.$$ \hspace{1cm} (9)

MP is used to measure the proportion of true positive cases among predicted positive cases, and it can be expressed as:

$$MP = \frac{1}{C} \sum_{c=1}^{C} \frac{TP_c}{TP_c + FP_c}.$$ \hspace{1cm} (10)

MIOU is used to measure the degree of overlap between true positive cases and predicted positive cases, and it can be expressed as:

$$MIOU = \frac{1}{C} \sum_{c=1}^{C} \frac{TP_c}{TP_c + FP_c + FN_c}.$$ \hspace{1cm} (11)

D. Results

1) The Overall Performance: The overall performance in the five experiments is listed in Tables XI–XIV.
The tables show that in experiments 2 and 3, the SSD method using BLSTM and BGRU is superior to SDIF and PRI-Tran in terms of all metrics; however, in experiments 1 and 4, the SSD method using BLSTM and BGRU is only superior to SDIF in terms of accuracy. To analyze this problem, we drew the confusion matrices of BLSTM and SDIF in experiments 1 and 4 for further analysis.

From the confusion matrices, in experiments 1 and 4, the prediction accuracy of BLSTM is superior or equivalent to that of SDIF for target pulses, but it is significantly lower than that of SDIF for noise pulses. This makes BLSTM inferior to SDIF in terms of MR, MP, and MIOU. BLSTM predicted wrong pulses as other categories, whereas SDIF predicted several wrong pulses as false alarm targets. This is because BLSTM adopts the mechanism of closed set recognition, whereas SDIF adopts the mechanism of “searching for PRI—searching for pulses with the found PRI”.

In experiment 1, BLSTM classified several noise pulses as D&S and staggered PRI signals. In experiment 4, BLSTM classified several noise pulses as staggered signals. This is because D&S and staggered PRI signals are more difficult to recognize than constant PRI signals.

Figs. 16 and 17 show real images of signals deinterleaved by BLSTM and SDIF in experiment 1. Fig. 16 illustrates that when data quality is good, the performance of BLSTM is almost perfect, whereas SDIF has a pulse loss problem for D&S and staggered signals. Fig. 17 shows that there are more noise pulses in the pulse stream deinterleaved by BLSTM, whereas pulse loss was more severe in the pulse stream deinterleaved by SDIF. The two images also show that BLSTM and SDIF have excellent deinterleaving efficiency for constant PRI signals.

For the deinterleaving of pulse streams by SDIF and PRI-Tran, we calculated the ratio of the number of correct detection targets to the total number of targets (DTR), the ratio of the number of missed detection targets to the total number of targets (MDTR), and the ratio of the number of false alarm targets to the total number of targets (FTR), which are listed as Table XV and XVI. SDIF and PRI-Tran pay a very high price for false alarms to obtain high deinterleaving accuracy. However, in our method, there is no missed detection and false alarm problem.

The experiments show that BLSTM and BGRU have better overall performance than DCN in terms of all metrics.

### Table XI
**Accuracy in the Five Experiments (%)**

| Experiment | BLSTM | BGRU | DCN | SDIF | PRI-Tran |
|------------|-------|------|-----|------|----------|
| 1          | 91.3  | 89.3 | 86.4| 84.5 | 60.6     |
| 2          | 98.2  | 97.6 | 87.7| 93   | 89.6     |
| 3          | 95.5  | 95.8 | 71.5| 90.5 | 69.6     |
| 4          | 91.5  | 90.7 | 84.2| 90.4 | 81.6     |
| 5          | 95.3  | 95.4 | 83.5|      |          |

### Table XII
**MR in the Five Experiments (%)**

| Experiment | BLSTM | BGRU | DCN | SDIF | PRI-Tran |
|------------|-------|------|-----|------|----------|
| 1          | 78.6  | 76.5 | 70.8| 84.3 | 67.3     |
| 2          | 97.9  | 96.3 | 79.3| 90.8 | 90.5     |
| 3          | 92.5  | 93.1 | 58.5| 90.5 | 80.5     |
| 4          | 81.1  | 79.5 | 71.0| 91.2 | 84.2     |
| 5          | 69.6  | 68.9 | 59.5|      |          |

### Table XIII
**MP in the Five Experiments (%)**

| Experiment | BLSTM | BGRU | DCN | SDIF | PRI-Tran |
|------------|-------|------|-----|------|----------|
| 1          | 83.6  | 81.9 | 77.7| 87.4 | 79.2     |
| 2          | 97.9  | 97.0 | 82.8| 92.4 | 86.6     |
| 3          | 94.1  | 94.6 | 63.3| 88.1 | 89.8     |
| 4          | 84.4  | 83.3 | 75.8| 89.8 | 88.4     |
| 5          | 82.0  | 81.2 | 70.6|      |          |

### Table XIV
**MIOU in the Five Experiments (%)**

| Experiment | BLSTM | BGRU | DCN | SDIF | PRI-Tran |
|------------|-------|------|-----|------|----------|
| 1          | 71.9  | 69.1 | 62.6| 74.8 | 48.2     |
| 2          | 95.3  | 93.6 | 69.3| 84.7 | 78.6     |
| 3          | 87.6  | 88.6 | 45.4| 79.8 | 71.8     |
| 4          | 74.5  | 72.6 | 61.2| 82.1 | 73.6     |
| 5          | 66.5  | 65.8 | 51.1|      |          |

Fig. 14. Confusion matrices of BLSTM and SDIF in experiment 1.

Fig. 15. Confusion matrices of BLSTM and SDIF in experiment 4.

### Table XV
**Missed Detection and False Alarm in Experiment 1**

| Method   | DTR     | MDTR   | FTR    |
|----------|---------|--------|--------|
| SDIF     | 0.993   | 0.007  | 5.077  |
| PRI-Tran | 0.922   | 0.078  | 10.562 |

### Table XVI
**Missed Detection and False Alarm in Experiment 4**

| Method   | DTR     | MDTR   | FTR    |
|----------|---------|--------|--------|
| SDIF     | 0.984   | 0.016  | 0.330  |
| PRI-Tran | 0.973   | 0.027  | 1.941  |

2) The Accuracy on the Specified Conditional Dataset: The accuracy of these methods on the specified conditional dataset is shown in Figs. 18–22.

a) Fig. 18 shows that the SSD method is superior to SDIF and PRI-Tran in terms of overall accuracy when PRI modulation
Fig. 16. Deinterleaving effect of BLSTM and SDIF in experiments 1, $\rho_t = \rho_n = 0$.

b) Fig. 19 shows that BLSTM and BGRU achieve almost perfect results when deinterleaving constant PRI signals with PRI values. The overall accuracy and robustness of SDIF are also excellent but inferior to those of BLSTM and BGRU. The robustness of DCN and PRI-Tran in pulse loss is not satisfactory.

c) Fig. 20 shows that BLSTM and BGRU have the best overall accuracy and robustness when using PRI values to deinterleave staggered PRI radar signals.

d) Fig. 21 shows that BLSTM and BGRU have the best overall accuracy when both PRI value and PRI modulation modes are used for deinterleaving and have the best robustness in pulse loss. However, their robustness is not as good as that of SDIF when noise pulses are very dense.

e) Figs. 18–21 show that the SSD method has better accuracy than SDIF and PRI-Tran when data quality is very good, regardless of the network structure. The robustness of BLSTM and BGRU is also better than that of SDIF and PRI-Tran when using PRI values only for deinterleaving. However, they are sometimes less robust than SDIF when the PRI modulation modes are needed for deinterleaving.

f) Fig. 22 shows that when there are multiple targets in each PRI modulation mode, BLSTM and BGRU are superior to DCN when deinterleaving on the basis of PRI modulation modes.

g) Figs. 18–22 show that the performance of BLSTM and BGRU is very similar in all cases and significantly better than that of DCN.

h) From the results of controlled experiments, the SDIF and PRI-Tran methods sometimes do not achieve the best deinterleaving accuracy when the data quality is the best. This is because the threshold is adjusted to make the overall performance of these methods the best, which also reflects the shortcomings of such methods. In addition, in such methods, pulses remaining after deinterleaving are regarded as noise pulses. Therefore, in some
cases, when the proportion of noise pulses increases, the overall accuracy increases; however, concretely, the accuracy of target pulses decreases.

3) The Computational Burden: For the SSD method, when the NN structure is determined, the computational burden of the SSD method is only related to the length of the pulse stream, and its time complexity can be expressed as $O(N)$.

The computational burden of SDIF is related to both the length of the pulse stream and the number of levels of TOA difference computed. Its time complexity can be expressed as...
$O(N) \times O(P)$, where $P$ represents the number of levels of TOA difference.

The computational burden of PRI-Tran is related to both the length of the pulse stream and the PRI range searched. Its time complexity can be expressed as $O(N) \times O(R_0)$, where $R_0$ represents the PRI range.

We randomly selected 1,000 samples from the mixed datasets of experiments 1–5 to determine the time consumed by different methods, as listed in Table XVII. As shown in Table XVII, when the data length is the same, different tasks have little effect on the computational burden of the SSD method. However, to achieve better deinterleaving efficiency, the $P$ of SDIF is different, whereas the $R_0$ of PRI-Tran is different in different tasks, which makes the computational burden of these two methods greatly different in different tasks.

| Experiment | BLSTM | BGRU | DCN | SDIF | PRI-Tran |
|------------|-------|------|-----|-------|----------|
| 1          | 358   | 351  | 51  | 3685  | 7977     |
| 2          | 358   | 352  | 52  | 61    | 90       |
| 3          | 359   | 358  | 52  | 11675 | 1988     |
| 4          | 363   | 190  | 50  | 739   | 1126     |
| 5          | 364   | 361  | 34  | \   | \        |
VI. CONCLUSION

In this paper, a radar signal deinterleaving method based on semantic segmentation is proposed. It uses semantic information contained in different radar signals to label pulses that constitute the same semantics as the same category. Two deinterleaving strategies comprehensively using PRI modulation modes and parameters are also proposed. Based on this research, we obtained the following conclusions.

1) Compared to the traditional methods [4], [5], [6], [7], [8], [9], [10], the proposed method does not need to search for the PRI or PRI period. When there are multiple pulses in a period, batching and PRI modulation mode recognition are no longer needed.

2) Compared to other deinterleaving methods based on NNs and automata [11], [12], [13], [14], this method does not require data to be digitized. There is no need to train a network for each target type and to iterate input and output data repeatedly. Very precise PRI values are not required as prior information.

3) The SSD method is easy to train and converge and still maintains ideal accuracy and good robustness in complex deinterleaving environments with a high pulse loss rate and noise to target ratio.

4) We propose two deinterleaving strategies that comprehensively use PRI modulation modes and parameters. In the future, we will investigate which method is more effective and which scenarios they are applicable.

5) Our research shows that BLSTM and BGRU have obvious advantages over DCN in this task. This may indicate that, although the performance of the CNN architecture exceeds that of the RNN architecture in dealing with representative sequence modeling tasks [43], RNNs are still superior to CNNs in dealing with sequences with strong mathematical relationships. This problem depends on further research and proof in the field of deep learning in the future.

6) In this paper, PRI modulation modes and parameters are proposed as semantic information. However, the method proposed in this paper still has limitations when PRI modulation modes of targets are the same and the PRI values or value range overlap. At this point, other parameters of FPD such as RF, PW, and PA can be used as semantic information to further divide the target into more categories. The deinterleaving method based on semantic segmentation and multi-parameter will be our future research direction. In addition, we have not studied the effect of our method on deinterleaving radar signals with other PRI modulation modes, such as jittered PRI, which will also be the subject of our next study.

REFERENCES

[1] P. Sharma, K. K. Sarma, and N. E. Mastorakis, “Artificial intelligence aided electronic warfare systems- recent trends and evolving applications,” IEEE Access, vol. 8, pp. 224761–224780, 2020.

[2] M. A. Nuhoglu, Y. K. Alp, and F. C. Akyon, “Deep learning for radar signal detection in electronic warfare systems,” in Proc. IEEE Radar Conf. (RadarConf20), 2020, pp. 1–6.

[3] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 801–818.

[4] H. K. Mardia, “New techniques for the deinterleaving of repetitive sequences,” IEEE Proceedings F: Radar and Signal Processing, vol. 136, pp. 149–154, 1989. [Online]. Available: https://ui.adsabs.harvard.edu/abs/1989IEEERP136.149M.

[5] D. J. Milojicic and B. M. Popovic, “Improved algorithm for the deinterleaving of radar pulses,” IEEE Proc. F: Radar Signal Processing, vol. 139, no. 1, pp. 98–104, 1992.

[6] K. Nishiguchi and M. Kobayashi, “Improved algorithm for estimating pulse repetition intervals,” IEEE Trans. Aerosp. Electron. Syst., vol. 36, no. 2, pp. 407–421, Apr. 2000.

[7] R. J. Orsi, J. B. Moore, and R. E. Mahony, “Spectrum estimation of interleaved pulse trains,” IEEE Trans. Signal Process., vol. 47, no. 6, pp. 1646–1653, Jun. 1999.

[8] J. Liu, H. Meng, Y. Liu, and X. Wang, “Deinterleaving pulse trains in unconventional circumstances using multiple hypothesis tracking algorithm,” Signal Process., vol. 90, no. 8, pp. 2581–2593, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168410000897.

[9] N. Visnevski, S. Haykin, V. Krishnamurthy, F. A. Dilkes, and P. Lavoie, “Hidden Markov models for radar pulse train analysis in electronic warfare,” in Proc. IEEE Int. Conf. Acoust., 2005, pp. v/597–v/600.

[10] Z. Ge, X. Sun, W. Ren, W. Chen, and G. Xu, “Improved algorithm of radar pulse repetition interval deinterleaving based on pulse correlation,” IEEE Access, vol. 7, pp. 30126–30134, 2019.

[11] Z. M. Liu and P. S. Yu, “Classification, denoising, and deinterleaving of pulse streams with recurrent neural networks,” IEEE Trans. Aerosp. Electron. Syst., vol. 55, no. 4, pp. 1624–1639, Aug. 2019.

[12] Z. Li, Z.-M. Liu, and Z. Huang, “Denoising of radar pulse streams with autoencoders,” IEEE Commun. Lett., vol. 24, no. 4, pp. 797–801, 2020.

[13] X. Li, Z. Liu, and Z. Huang, “Deinterleaving of pulse streams with denoising autoencoders,” IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 6, pp. 4767–4778, Dec. 2020.
[14] Z. M. Liu, “Online pulse deinterleaving with finite automata,” IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 2, pp. 1139–1147, Apr. 2020.

[15] R. G. Wiley, “ELINT: The interception and analysis of radar signals,” Artech House, Portland House, 2006.

[16] J. W. Tao, C. Z. Yang, and C. W. Xu, “Estimation of PRI stagger in case of missing observations,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 11, pp. 7982–8001, Nov. 2020.

[17] W. Cheng, Q. Zhang, J. Dong, C. Wang, X. Liu, and Q. Fang, “An enhanced algorithm for deinterleaving mixed radar signals,” IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 6, pp. 3927–3940, Dec. 2021.

[18] Y. Liu and Q. Zhang, “Improved method for deinterleaving radar signals and estimating PRI values,” IET Radar Sonar Navigation, vol. 12, no. 5, pp. 506–514, 2018.

[19] J. Johnson, A. Karpathy, and F. F. Li, “Fully convolutional networks for semantic segmentation,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2015, pp. 3431–3440.

[20] G. Lin, A. Milan, C. Shen, and I. Reid, “RefineNet: Multi-path refinement networks for high-resolution semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 5168–5177.

[21] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention, 2015, pp. 234–241.

[22] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[23] L.-C. Chen, G. Papandreou, I. Kokkinos, K. P. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected CRFs,” 2015, arXiv:1412.7062.

[24] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 834–848, Apr. 2018.

[25] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, pp. 1735–1780, 1997.

[26] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[27] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” 2014, arXiv:1412.3555.

[28] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in Adv. Neural Inf. Process. Syst., vol. 4, pp. 3104–3112, Jan. 2014.

[29] K. Cho et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in Proc. Conf. EMNLP, 2014, pp. 577–585.

[30] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in Proc. Int. Conf. Learn. Representations, 2015.