IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 27, 2019, accepted November 19, 2019, date of publication November 25, 2019,
date of current version December 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2955819

A Novel Radar Signals Sorting Method-Based Trajectory Features

QIANG GUO, LONG TENG, LIANGANG QI, XIAOWEI JI, (Student Member, IEEE), AND JIANHONG XIANG

College of Information and Communications Engineering, Harbin Engineering University, Harbin 150000, China

Corresponding author: Jianhong Xiang (xiangjianhong@hrbeu.edu.cn)

This work was supported in part by the Research Funds for the Central Universities under Grant 3072019CFG0802, in part by the National Natural Science Foundation of China under Grant 61371172, in part by the International Science and Technology Cooperation Program of China (ISTCP) under Grant 2015DFR10220, in part by the National Key Research and Development Program of China under Grant 2016YFC0101700, in part by the Heilongjiang Province Applied Technology Research and Development Program National Project Provincial Fund under Grant GX16A007, and in part by the State Key Laboratory Open Fund under Grant 702SKL201720.

ABSTRACT When several foreign fighters with the same type enter detection range, the electronic warfare (EW) receivers will intercept many the same type radar emitter signals. If the intercepted pulse is processed by the traditional sorting methods, the number of emitters cannot be identified. The main reason is that the same type of radar has similar parameters. It will cause a devastating influence on subsequent strategic decisions. A novel sorting method based on the trajectory features is proposed to solve the aforementioned problems. First, the trajectory features of the intercepted pulse signal are extracted. Then, the segmentation method is utilized to preprocess the signals, which enhances the computing efficiency and improves the sorting accuracy. Meanwhile, a prediction framework based on long short-term memory (LSTM) recurrent neural network is established to forecast pulses. Finally, the radar stagger pulses are sorted by forecast pulses. The simulation results show that the proposed method can recognize the number of emitters and achieve high sorting accuracy. It provides a new idea for the radar signals sorting of the same type.

INDEX TERMS Signal sorting, trajectory features, recurrent neural networks (RNNs), long short-term memory (LSTM).

I. INTRODUCTION

Radar signal sorting is the crucial technology of modern warfare and also as an important part of the electronic countermeasures (ECM). Whether or not the radar signal can be correctly sorted, is the key to electronic warfare. The traditional radar signal sorting methods are mainly divided into two categories: sorting methods based on the one-dimensional parameter, i.e., pulse repetition interval (PRI), and sorting methods based on multi-parameter. The two methods are briefly described below.

In the decades of research, researchers have proposed some PRI-based sorting methods, including cumulative difference histogram method [1], sequential difference histogram method [2], PRI spectrum [3], [4] and plane transform [5], [6]. The sorting method based on PRI is suitable for radar signals with a small variation range of PRI parameters. Moreover, with the rapid development of radar technology, modern radars mostly use modulation modes such as PRI agility and random PRI. Sorting techniques that only use PRI information are no longer applicable to modern radars. Hence, more and more researchers pay attention to the sorting methods based on multi-parameter.

The methods based on multi-parameter mainly include unsupervised clustering and supervised classification. Both unsupervised clustering and supervised classification utilize the Time of Arrival (TOA), Pulse Amplitude (PA), Pulse Width (PW), Radio Frequency (RF) and Direction of Arrival (DOA) to sort radar signals. The combination of the above parameters is called Pulse Description Word (PDW). As an unsupervised classification method, the clustering technique does not need the prior information of radar emitters, while it requires excellent similarity of the same emitter signals and differences of different emitter signals. Clustering methods include fuzzy clustering [7], [8], support vector clustering [9], [10] and dynamic clustering based on the nuclear field [11]. As a classification algorithm, supervised classification requires prior information and knowledge.

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Liu.

It mainly includes the self-organizing probabilistic neural network (PNN) sorting algorithm [12], one-dimension Kohonen neural network sorting algorithm [13], self-organized interval type-2 fuzzy neural network sorting algorithm [14].

However, in the complex electromagnetic environment, the electronic warfare (EW) receiver often intercept radar signals with the same type, and these signals work in the same mode. If the radar signals have the same DOA, these pulse signals have almost similar parameters except for the RF. In this case, the traditional sorting methods are difficult to judge the number of radar emitters and sort the radar signals. Therefore, we must explore new feature information to sort the signal. The reference [15], [16] describes the PA of the same radar signals has a certain variation rule in a typical scenario. The reference [17] proposes a Hough transform to fit the pulse amplitude envelope for pre-sorting. The reference [18] proposes to use the extended kalman filter (EKF) to track the pulse amplitude envelope to determine whether the pulses belong to the same sequence. They demonstrate that pulse amplitude information can also be used for signal sorting. Therefore, when the other dimension parameters are the same, we explore the pulse amplitude information to sort the same type radar signal. We extract the common trajectory features that are suitable for different motion scenarios. The features is input to LSTM model for off-line training. After the training is completed, the radar pulse signals are input into the forecast framework for online prediction. The final sorting result is obtained by predicting the source of the pulse.

The rest of this paper is organized as follows. The second section is the mathematical models involved in this paper. The third part introduces the proposed algorithm in detail. The fourth part discusses and analyzes the simulation results. Finally, we summarize this article in the fifth part.

II. THE MODEL

A. THE PULSE AMPLITUDE MODEL

PA is the received signal power of the pulse intercepted by the EW receiver. It is related to the relative distance between the receiver and radar emitter, antenna gain, and propagation medium [16]. It can be given by the following formula:

$$P_r(t) = \frac{P_t G_r \lambda^2}{(4\pi R)^2 L} G_t[\theta(t), \phi(t)] \tag{1}$$

where P_t and P_r are the transmitted and the received power of radar signals. G_r is the receiver antenna gain, λ is the wavelength, R is the range between the radar emitter and EW receiver. L is the loss factor. Atmospheric propagation losses are proportional to range and frequency. Polarization mismatch between antennas is another factor affecting PA. $G_t[\theta(t), \phi(t)]$ is the gain of radar transmitter antenna located at the azimuth and pitch angles of the receiver at time t. Since the change in geometry (distance and angle) between the radar and the EW receiver is negligible, the term $P_t G_r \lambda^2/(4\pi R)^2 L$ is assumed to be constant. This assumption applies to stationary engagement scenarios and scenarios where the scan rate of the antenna is much faster than the motion of the airborne platform, which is most common in EW. When the relative motion between the radar emitter and the receiver is significant, the power of the received signal depends on the range R. If the power of received signals is above the sensitivity level of the EW receiver, radar pulse signals with different amplitudes can be detected and analyzed.

B. THE LSTM MODEL

Recurrent neural networks (RNNs) architecture is a powerful deep learning classification method especially applied to sequential data. RNNs as the current state-of-the-art method is widely applied in natural language processing (NLP) and speech recognition [19]–[21]. It is widely acknowledged that there are many differences between the RNNs and the traditional feedforward neural network. RNNs builds model based on sequential data and links current information with the previous environment information. This means that decisions made by RNNs at time $t-1$ could affect decisions made at time t. Besides, RNNs are trained by backpropagation through time (BPTT) [22]. However, it is difficult for RNNs to learn long-range dependencies because of gradient vanishing or gradient exploding [23], [24]. To solve this problem, Hochreiter and Schmidhuber [25] first proposed the long short-term memory (LSTM) architecture. Subsequently, after the improvement of Gers *et al.* [26], LSTM has been widely applied in many fields. The following is a brief introduction to LSTM.

The core unit of the LSTM is the memory block. Memory block contains memory cells with self-connections and special multiplicative units called gate. The gate includes the input gate, the output gate, and the forget gate. The decisions for the forget gate, the input gate and the output gate are all dependent on the current input X_t and the previous output H_{t-1}. The input gate determines which elements of the state vector should be retained. The forget gate controls which elements in the S_{t-1} should be forget or reset. With the updating of the internal state, the output gate controls which the internal state S_t should be used as the LSTM output H_t. This process continues to iterate. The weights and bias parameters are learned by minimizing the difference between the output of the LSTM and the true training labels.

We assume the input data set is $\{X_1, X_2, \cdots, X_T\}$. Each member of the set represents a k-dimensional eigenvector at time t. The formulations of the LSTM structure are given by (2)-(7),

$$F_t = \text{sigmoid}\,(W_{FX}X_t + W_{FH}H_{t-1} + B_F) \tag{2}$$

$$I_t = \text{sigmoid}\,(W_{IX}X_t + W_{IH}H_{t-1} + B_I) \tag{3}$$

$$G_t = \tanh\,(W_{GX}X_t + W_{GH}H_{t-1} + B_G) \tag{4}$$

$$O_t = \text{sigmoid}\,(W_{OX}X_t + W_{OH}H_{t-1} + B_O) \tag{5}$$

$$S_t = G_t \odot I_t + S_{t-1} \odot F_t \tag{6}$$

$$H_t = \tanh\,(S_t) \odot O_t \tag{7}$$

where $W_{(\cdot)(\cdot)}$ are weight matrices, $B_{(\cdot)}$ is the corresponding bias. F_t, I_t, O_t and G_t represent forget gate, input gate,

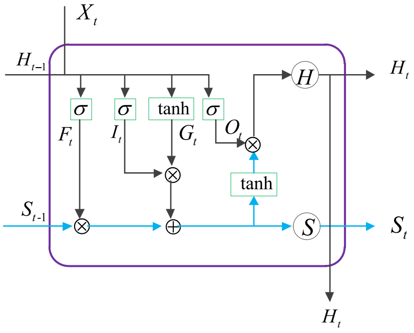


FIGURE 1. The LSTM block structure.

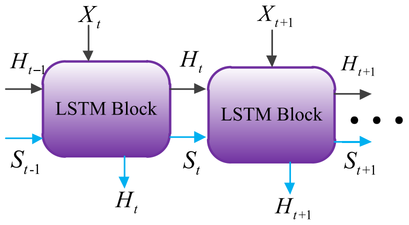


FIGURE 2. The LSTM sequential architecture.

output gate and candidate gate, respectively. H_t is the output. S_t is the internal memory state. The internal structure of the LSTM block at a single time step is shown in Fig.1. The LSTM sequential architecture of multiple time steps is shown in Fig.2. where σ represents the sigmoid function. The weight and bias parameters of each time step are shared. Through this structure, we can link the information of the current time step with the output of the future time step.

III. PROPOSED METHOD

In this paper, the TOA and PA of radar emitter signals are formed into radar pulse description train (U_i, $i = 1, 2, \cdots M$, M is the total number of pulses.) with two-dimensional information. First, the trajectory features of pulse signals are extracted, and the U_i is cut into m segments by trajectory features. Second, we polymerize pulse clusters of m segments to get subsequence 1 and then cut the subsequence 1 into n subsequence 2. Moreover, the peak value of subsequence 2 is extracted. Finally, the peak value feature is input to LSTM model for off-line training. After the training is completed, the radar pulse signals are input into the system for online prediction. We describe the sorting scheme in detail as shown in Fig.3. The following describes the various parts of the system.

A. TRAJECTORY FEATURE EXTRACTION

During the scanning process of radar, the radar signal intercepted by the EW receiver has obvious laws in the PA dimension. The PA variation is mainly affected by the relative range between the EW receiver and the emitter and the radar scanning mode. So it's relatively stable. We extract trajectory features from these signals. The trajectory features are listed in Table 1. Where i is the number of pulse clusters, $i = 1, 2, \cdots, m$. k_b is the signals sequence number in the pulse cluster, $1 \leq \mathrm{k}_b \leq \mathrm{end}$, *end* is the overall number of the signals number in a cluster. $center(i)$ is the barycenter number. l is the peak value number, $1 \leq l \leq n$, $j = 1, 2, \cdots, end$, $v = 1, 2, \cdots, end, j \neq v$.

B. PREPROCESSING

Considering a large number of pulses intercepted by the EW receiver, and these pulses appear as clusters. If all pulse signals are processed directly, it not only causes waste of computing resources and reduces the efficiency of the algorithm, but also buries the trajectory variation rules in a large number of signals. Therefore, this paper proposes the segmentation preprocessing method based on trajectory features, and the process is shown in Fig.4. The steps of segmentation preprocessing are described below.

1) PULSE CLUSTERS CLASSIFICATION

The unprocessed data is shown in Fig.5 (a). First, we take the first pulse as the reference pulse. Second, the difference value between subsequent pulses and the reference pulse is calculated in turn. Third, we define two accumulators, C_1 and C_2, and the initial values of C_1 and C_2 are zero. If the difference is less than the threshold T, the accumulator C_1 is incremented by one, otherwise, the accumulator C_2 is incremented by one. When we constantly get Q differences that are all greater than the threshold T, the reference pulses are updated and the accumulators C_1 and C_2 are compared. If $C1/C2 < 2$, the segment pulse is an overlapping pulse

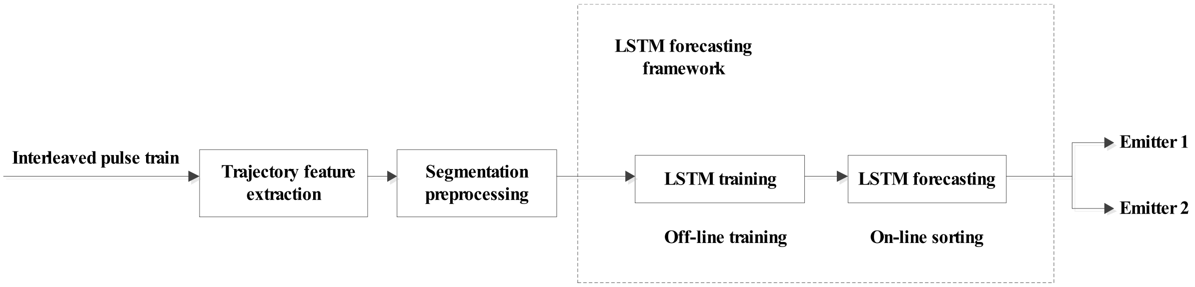


FIGURE 3. The system components.

TABLE 1. Trajectory features.

Number	Item	Formula
1	Maximum of the difference value of PA within the cluster	$\max_j (PA(i,k_j)) - \min_v (PA(i,k_v))$
2	Dwell-time of pulse cluster	$(TOA(i,k_{end})) - (TOA(i,k_1))$
3	Barycenter	$TOA_{\text{center}} = (TOA(i,k_{\text{end}}) - TOA(i,k_1))/2$ $PA_{\text{ center}} = (PA(i,k_{\text{end}}) - PA(i,k_1))/2$
4	(Difference value of PA) DPA	$\left\|PA_{\text{center}(i)} - PA_{\text{center}(i+1)}\right\|$
5	(Difference value of TOA) DTOA	$\left\|TOA_{center(i)} - TOA_{center(i+1)}\right\|$
6	Peak value	$(TOA_l, PA_l), PA_{l-1} < PA_l < PA_{l+1}$

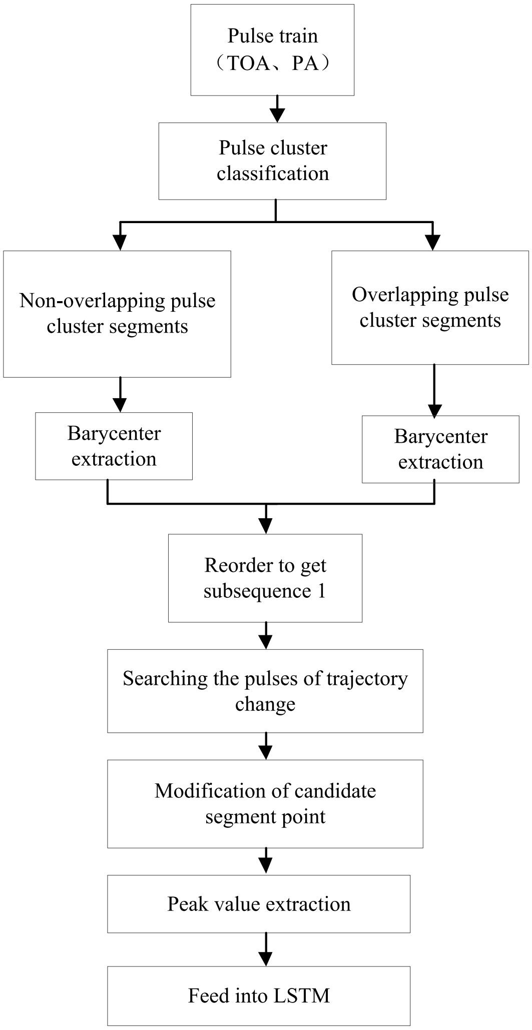

Pulse train
（TOA、PA）
Pulse cluster classification
Non-overlapping pulse cluster segments
Overlapping pulse cluster segments
Barycenter extraction
Barycenter extraction
Reorder to get subsequence 1
Searching the pulses of trajectory change
Modification of candidate segment point
Peak value extraction
Feed into LSTM

FIGURE 4. Preprocessing flow.

cluster, otherwise it is a non-overlapping pulse cluster. The accumulators C_1 and C_2 are then set to zero. We repeat the above-mentioned steps until the last pulse. Finally, the classification results of overlapping and non-overlapping pulse clusters are obtained. According to the statistical law of the data, we set Q to 10 and the threshold T to 0.1. Non-overlapping pulses can be defined as:

$$PA_T = \left\{PA_{T_1}, PA_{T_2}, \cdots, PA_{T_W}\right\} \tag{8}$$

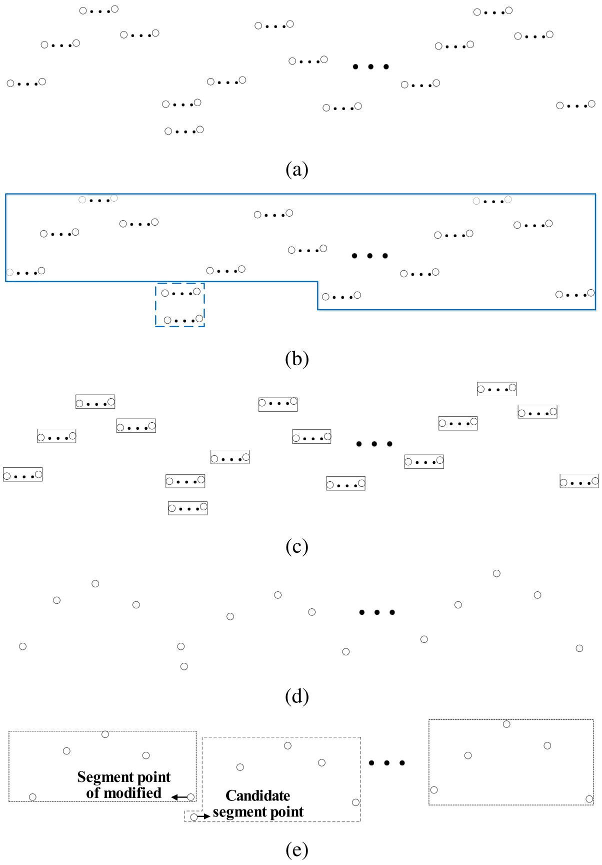


FIGURE 5. Preprocessing: (a) Unprocessed pulse clusters; (b) Pulse clusters classification; (c) Segmentation processing to obtain subsequence 1; (d) Barycenter; (e) Segmentation again to obtain subsequence 2.

where W is the total number of nonoverlapping pulses. The overlapped pulses can be defined as:

$$PA_J = \left\{PA_{J_1}, PA_{J_2}, \cdots, PA_{J_L}\right\} \tag{9}$$

where L is the total number of overlapped pulses. The solid line frame in Fig.5 (b) is the nonoverlapping pulse and the dashed line frame is the overlapped pulse.

2) NONOVERLAPPING PULSE CLUSTER PROCESSING

First, we take the first pulse as the reference pulse. Second, the difference value between all the pulses and the reference pulse is calculated in turn. If the difference value is greater

Algorithm 1 Nonoverlapping Pulse Cluster Processing

```
Input: $PA_T, F_1, i, j, PA_R, W$ in
Output: $PA_S$ out
    Initialisation:
    $PA_R = PA_{T_1}$;
    $i = 1$;
    $j = 1$;
    LOOP Process
    while ($i < W$) do
        $DP_i = PA_{T_{i+1}} - PA_R$;
        if $DP_i > F_1$ then
            $PA_S[j] = PA_{T_i}$;
            $PA_R = PA_{T_{i+1}}$;
            $j = j + 1$;
        end if
        $i = i + 1$;
    end while
    return $PA_S$
```

than feature 1, the pulse is taken as a segment point. Then the reference pulse is updated and the above operation is repeated until the last pulse. Finally, we get $Z+1$ segments pulse based on the segment points. It can be defined as:

$$PA_{sn} = \left\langle \begin{array}{l} \left\{PA_{T_1}, PA_{T_2}, \cdots, PA_{S_1}\right\}, \\ \left\{PA_{S_1+1}, PA_{S_1+2}, \cdots, PA_{S_2}\right\}, \\ \cdots, \left\{PA_{S_Z}, PA_{S_Z+1}, \cdots, PA_{T_W}\right\} \end{array} \right\rangle \tag{10}$$

where $\{\cdot\}$ is a segment pulse. The segment point set is:

$$PA_S = \left\{PA_{S_1}, PA_{S_2}, \cdots PA_{S_Z}\right\} \tag{11}$$

where Z is the total number of segment points. This process can be described by Algorithm 1. Where F_1 is the feature 1, PA_R is the reference pulse.

3) OVERLAPPED CLUSTERS PROCESSING

First, we take the first pulse as the reference pulse 1. Then the difference value between all the pulses and the reference pulse is calculated in turn. When the difference value is greater than feature 1, the pulse is taken as reference pulse 2. Second, the difference value between all pulses and the reference pulse 1 is calculated, and the difference value between all pulses and the reference pulse 2 is calculated. If one of the differences is greater than feature 1, the pulse is taken as segment point. Then we update the reference pulse 1 and 2. Moreover, we repeat the operation above until the last pulse. In the end, we get $V+1$ segments pulse based on the segmentation points. It can be defined as:

$$PA_{so} = \left\langle \begin{array}{l} \left\{PA_{J_1}, PA_{J_2}, \cdots, PA_{D_1}\right\}, \\ \left\{PA_{D_1+1}, PA_{D_1+2}, \cdots, PA_{D_2}\right\}, \\ \cdots, \left\{PA_{D_V}, PA_{D_V+1}, \cdots, PA_{J_L}\right\} \end{array} \right\rangle \tag{12}$$

where $\{\cdot\}$ is the pulse segments. The segment point set is:

$$PA_D = \left\{PA_{D_1}, PA_{D_2}, \cdots PA_{D_V}\right\} \tag{13}$$

Algorithm 2 Overlapped Clusters Processing

```
Input: $PA_J, F_1, i, j, PA_{R_1}, PA_{R_2}, windows$ in
Output: $PA_D$ out
    Initialisation:
    $PA_{R_1} = PA_{J_1}$;
    $i = 1$;
    $j = 1$;
    $windows = 5$;
    LOOP Process
    while ($i < W$) do
        $DP_i = PA_{J_{i+1}} - PA_{R_1}$;
        if $DP_i > F_1$ then
            $PA_{R_2} = PA_{J_i}$;
        end if
        $i = i + 1$
    end while
    LOOP Process
    while ($i < W$) do
        $PA_{\mathrm{windows}} = \left\{PA_{J_i}, PA_{J_{i+1}}, \cdots, PA_{J_{\mathrm{i+windows}}}\right\}$
        $DPR1_i = PA_{\mathrm{windows}} - PA_{R_1}$
        $DPR2_i = PA_{\ \mathrm{windows}} - PA_{R_2}$
        if every element of $DPR1_i > F_1$ or every element of
        $DPR2_i > F_1$ then
            $PA_D[j] = PA_{J_i}$;
            $PA_{R_1} = PA_{J_{i+1}}$;
            $PA_{R_2} = PA_{J_{i+2}}$;
            $j = j + 1$
        end if
        $i = i + 1$
    end while
    return $PA_D$
```

where V is the total number of segment points. This process can be described by Algorithm 2. Where F_1 is the feature 1, PA_{R_1} is the reference pulse 1 and PA_{R_2} is the reference pulse 2.

4) ORDERING

We order the segments pulse by TOA, and get $Z+V+2$ segments pulse, i.e., subsequence 1. Fig.5 (c) shows the subsequence 1. Where the rectangular box represents the segment pulse, i.e., the pulse cluster, which contains many pulses with small jitter.

5) BARYCENTER EXTRACTION

We extracted the barycenter of the subsequence 1, i.e., trajectory feature 3. The extraction results are shown in Fig.5 (d). Where the circle represents barycenter.

6) SEARCHING THE PULSES OF TRAJECTORY CHANGE

We calculate the DPA_{center} of adjacent pulses of barycenter in turn, starting with the first barycenter. The DPA_{center} can be defined as:

$$DPA_{center} = PA_{center(i+1)} - PA_{center(i)} \tag{14}$$

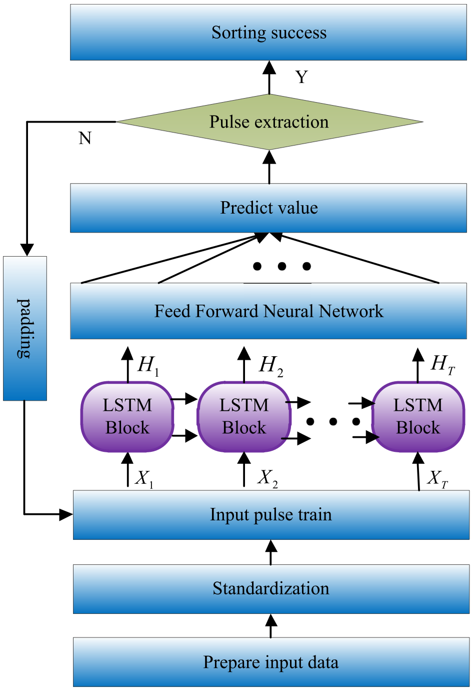


FIGURE 6. **The forecasting framework based LSTM.**

If $DPA_{\text{center}} > 0$, i.e., the trajectory shows an upward trend, the $\left(TOA_{center(i)}, PA_{center(i)}\right)$ is the candidate segment point.

7) MODIFICATION OF CANDIDATE SEGMENT POINT

We calculate the $DTOA_{center}$ of adjacent pulses of barycenter, denoted as $DTOA_1$ and $DTOA_2$, respectively. It can be written as:

$$\begin{aligned} DTOA_1 &= \left|TOA_{\text{center}(i-1)} - TOA_{\text{center}(i)}\right| \\ DTOA_2 &= \left|TOA_{\text{center}(i+1)} - TOA_{\text{center}(i)}\right| \end{aligned} \tag{15}$$

If the difference value between $DTOA_1$ and features 2 is greater than the difference value between $DTOA_2$ and features 2, the $\left(TOA_{center(i)}, PA_{center(i)}\right)$ is modified to $\left(TOA_{\text{center}(i-1)}, PA_{\text{center}(i-1)}\right)$. We get n subsequences 2 based on the segmentation points, as shown in Fig.5 (e). Where the dashed box represents the subsequence 2.

8) PEAK VALUE EXTRACTION

The peak value of each subsequence 2 is extracted, i.e., trajectory feature 6. We feed it to the LSTM model.

C. THE FORECASTING FRAMEWORK BASED LSTM

The forecasting framework based LSTM is given as Fig.6. It includes four functional modules: the input layer, the hidden layer, the output layer, and pulse extraction. The input layer is responsible for standardizing the segmented pulse train. The hidden layer is composed of LSTM layer and conventional feedforward neural networks. The forecasting value is given by the output layer. The pulse extraction adopts the iterative method to extract point by point. We define the input as $F = \{f_1, f_2, \cdots, f_l\}$. Since the LSTM is sensitive to data scale, the classic standardization method $(z - score)$ is used to standardize the data. The normalized train can be expressed as:

$$F' = \{f_1', f_2', \cdots, f_l'\} \tag{16}$$

$$f_r' = \left(f_r - \sum\nolimits_{r=1}^{l} f_r/l\right) / \sqrt{\sum\nolimits_{r=1}^{l} (f_r - l)^2 / l}$$
$$1 \le r \le l,\ r \in N \tag{17}$$

The standardized train is input into LSTM. Then the outputs of LSTM are fed forward to the conventional feed-forward neural network, which maps the output of LSTM to a single value Y, i.e., the forecasting pulse. Then we search the pulses to sort signals.

The pulse signals are searched according to the forecasting pulse and the search threshold [27]. If the pulse is found, the pulse will be extracted. If it is not found, the position of forecasting pulse will be padded. Then the search will continue. If two consecutive searches fail, the search will be stopped. Repeat the above process until the number of remaining pulses are less than 3 and the search is completed. The search process is shown in Fig.7. According to prior knowledge, we generally take the pulse maximum jitter rate as the threshold.

All frameworks in this paper are built on a PC with 4 GHz, i7 processor, and 16GB of memory using the MATLAB2018b with the Deep Learning Toolbox. After some initial tests, the results of Adam optimizer [28] with higher computational efficiency were slightly better than those of other candidates, including classic stochastic gradient descent (SGD), Adagrad [29], and Adadelta, [30]. Therefore, we use Adam to train the forecasting framework proposed in this paper.

IV. EXPERIMENT

A. RADAR PULSE DATA

To evaluate the feasibility of the proposed method, we use electromagnetic environment simulation software [31] to generate radar emitter signals for simulation verification. Radar parameters are shown in Table 2. In particular, the DOA of both radar signals are 36° and PW are 1300ns. We simulated two scenarios, one for training scenarios and the other for sorting test scenarios. The training scenario consisted of two aircraft, one for the EW receiver and the other for the radar equipment. There are three aircraft in the sorting test scenario, one is a EW receiver, and the other two are equipped with radar equipment of the same type. And the two aircraft are located in the tolerance range of the same DOA of the EW receiver. The training scene is simulated 3 times to get 3 sets of training data. The flight attitude, route, and PRI of the three sets of data are different. One of the data is shown in Fig.8 (a). The corresponding partial enlargement is shown as Fig.8 (b). Fig.9 (a) shows the radar pulse distribution of the test scenario, and the corresponding partial enlargement is shown as Fig.9 (b).

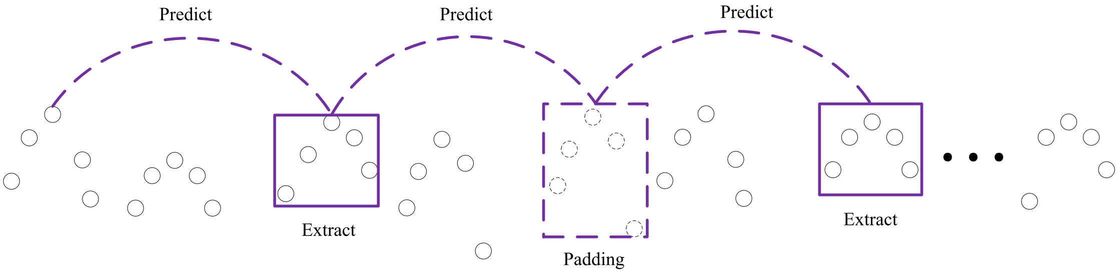


FIGURE 7. Search process.

TABLE 2. Information of radar parameters.

Item	Value
Functional State	Search
Transmitters power	$12kw$
Pattern	Sinc
Maximum gain	38dB
First sidelobe level	-34dB
Average sidelobe level	-44dB
3dB Beam Width	$2°$(azimuth) $2°$(elevation)
Antenna scanning rate	$50°/s$
Azimuth scanning	$\pm 15°$
Elevation scanning	$\pm 4°$
RF	$9400\text{MHz} < \text{RF} < 9800\text{MHz}$ agility
PRI	$(30\mu s < PRI < 50\mu s)$ group stagger
DOA	$\pm 36°$
PW	1300ns

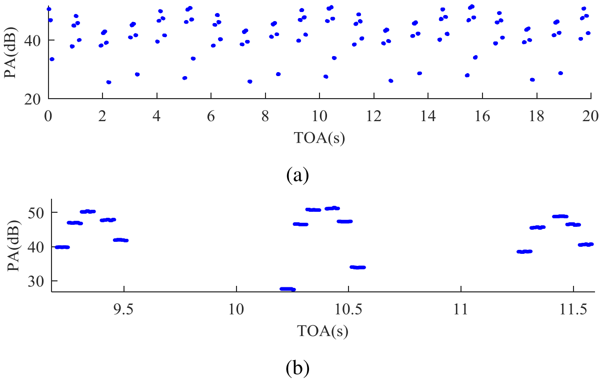


FIGURE 8. Distribution of pulse in training scenario: (a) One of the training data; (b) Partial enlargement.

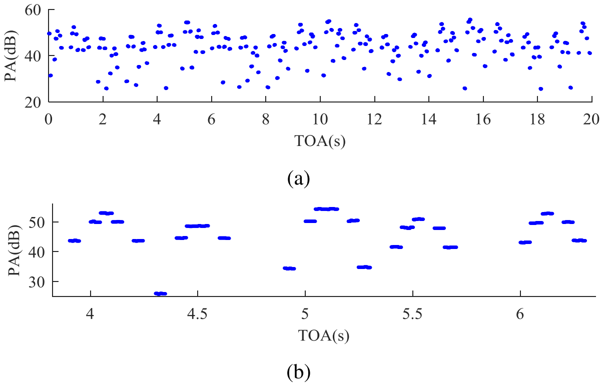


FIGURE 9. Distribution of pulse in the test scenario: (a) Test pulse data; (b) Partial enlargement.

B. PARAMETER SETTINGS

In general, hyper-parameter tuning is essential to obtain a better forecasting performance. At present, there is no universally accepted method. In this paper, we only focus on the methods for the best overall performance. Therefore, we adopt some rules of thumb for hyper-parameter tuning. Previous studies have shown that the performance of the network is relatively insensitive to any combination of some layers and the size of layers [32]. This point is also confirmed in the paper [33]. According to findings in [34], the number of hidden nodes should be sufficiently large. In this paper, we set the number of LSTM layer to be 1, the number of hidden nodes to be 128, and the number of nodes of the full connection layer to be 256. Other parameter settings: learning rate is 0.01, gradient threshold is 1. When the gradient is greater than 1, $L2$ regularization is used for gradient reduction. The number of iterations is 130. Root Mean Square Error (RMSE) was adopted as the loss function. The formula is as follows:

$$loss = \left[\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2\right]^{\frac{1}{2}} \tag{18}$$

where n is the number of samples, y^i is the expected value, and $\hat{y}^i$ is the forecasting value.

C. SIMULATION RESULTS AND DISCUSSION

In this section, we mainly analyze the results of the simulation. Traditional signal sorting methods include the methods based on PRI and the methods based on multi-parameter. Since both radar signals adopt the same PRI modulation mode of group stagger, the method based on PRI cannot sort the emitter signals. Therefore, we compare the proposed method with the method based on multi-parameter proposed in [11]. In order to verify the advantages of the LSTM model in different types of recurrent neural networks, the hidden layer cells of the LSTM model were replaced with RNN and gated recurrent unit (GRU) structures, and experiments

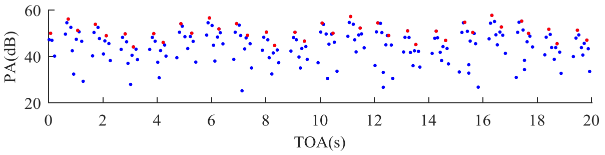
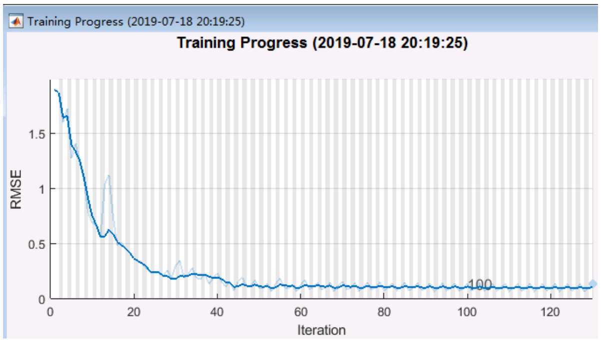


FIGURE 10. Barycenter and peak value features.


Training Progress (2019-07-18 20:19:25)

Training Progress (2019-07-18 20:19:25)

FIGURE 11. The training process of the LSTM model.

were performed with the same parameters. Besides, we also compare it with the DNN model. The DNN uses three hidden layer structures with hidden nodes of 40, 20, and 10, respectively. The extracted barycenter and peak value are shown in Fig.10. Where the blue point is the barycenter and the red point is the peak value. The training process of LSTM is shown in Fig.11. The pulse distribution of TOA and PA parameters is shown in Fig.12. The different colors represent different radar emitter signals. The sorting results are shown in Table 3.

Fig.12 (f) is the sorting result of [11]. It can be seen from the figure that the method sorts the two emitters into three emitters. The method uses pulse data for dynamic clustering. However, the pulse signals in the paper are the same type, i.e., the multidimensional parameters are the same. This has made it difficult to automatically find the boundaries using the method of [11]. Therefore, the sorting accuracy of this method is very low. Fig.12 (g) and Fig.12 (h) show the prediction results of GRU and RNN, respectively. The pulse in the black box is a prediction of failure. The reason for the prediction failure is that high prediction accuracy is required in the processing of pulse prediction, and the prediction accuracy of the three models is $LSTM > GRU > RNN$. Therefore, using GRU and RNN as hidden layer cells will cause errors in the prediction process. The predicted results of the DNN model are shown in Fig.12 (i). It can be seen from the figure that there are a large number of prediction errors, because the DNN model is not sensitive to time series data. Fig.12 (b) show results of proposed method. In order to better show the results, we selected three parts of Fig. 12 (a) and (b) to enlarge. It can be seen from Fig.12 (c) - (e) that the method can correctly sort whether they are overlapped pulses or nonoverlapping pulses. From the above, this method can not only judge the number of emitters, but also get high sorting accuracy. Compared with [11], DNN, RNN, and GRU, the accuracy is improved by 56.11%, 25.91%, 11.92%, and 7.37%, respectively.

In order to validate the stability of the method for realistic circumstances, the robustness of the proposed sorting system

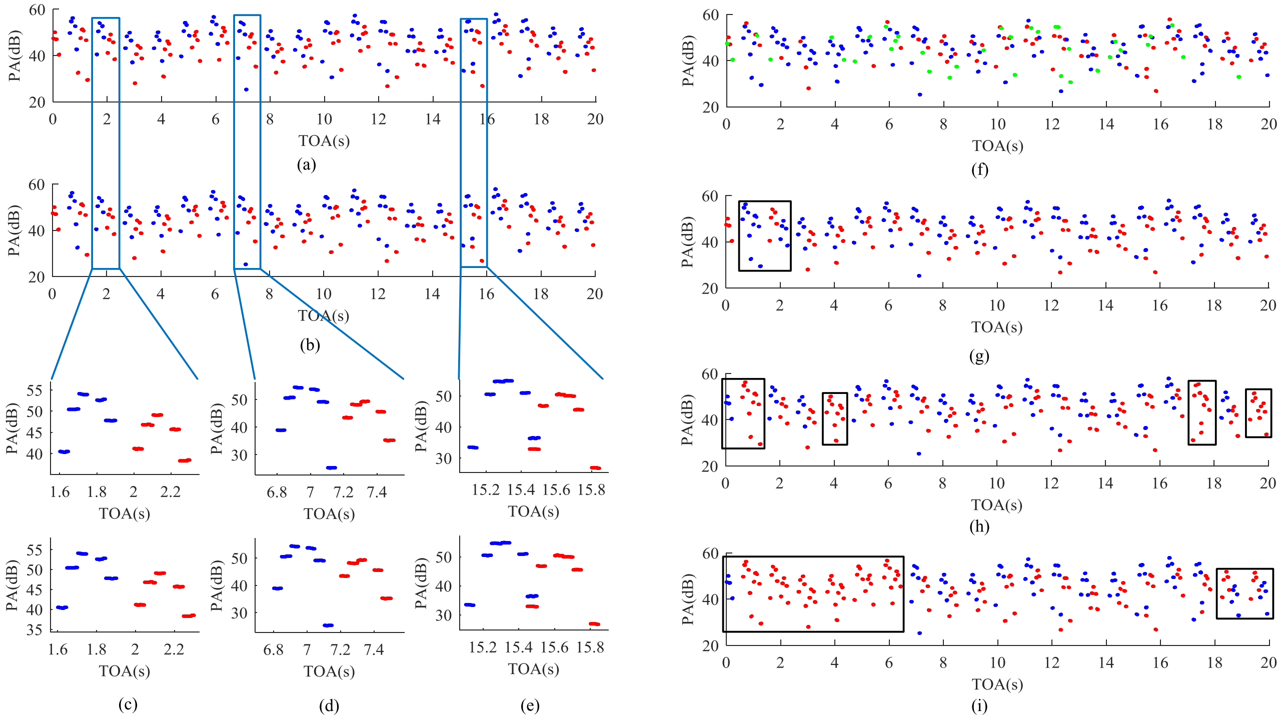


FIGURE 12. Pulse distribution of TOA and PA parameters. The proposed method compared with reference [11], GRU, RNN, and DNN: (a) True pulse data; (b) Sorting results of the proposed method; (c) Partial enlargement 1; (d) Partial enlargement 2; (e) Partial enlargement 3; (f) Sorting results of the method in [11]; (g) Sorting results of the GRU; (h) Sorting results of the RNN; (i) Sorting results of the DNN.

TABLE 3. Sorting accuracy.

Item	Proposed method	[11]	GRU	RNN	DNN
The true number of radars	2	2	2	2	2
Sorting number of radars	2	3	2	2	2
The number of total pulses	278346	278346	278346	278346	278346
The number of correct pulses	271365	115179	250845	238180	199240
The number of false pulses	6981	163167	27501	40166	79106
Sorting accuracy/%	97.49	41.38	90.12	85.57	71.58

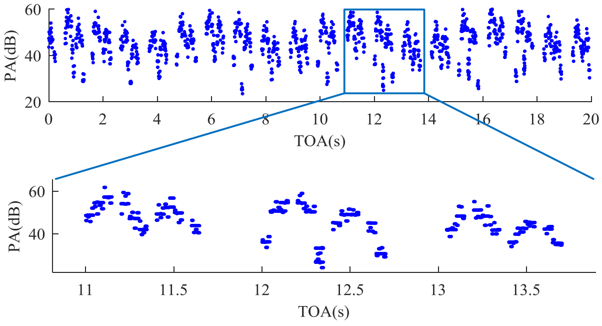


FIGURE 13. Pulse signal with jitter, missing, and false alarm.

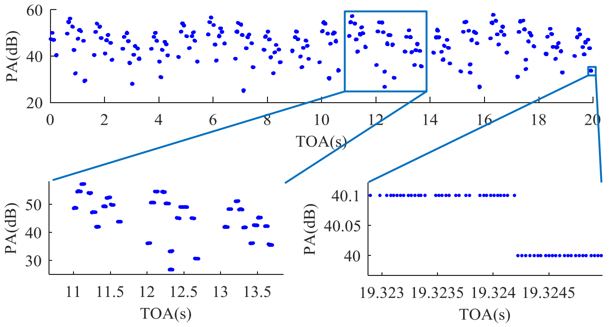


FIGURE 15. Filtered pulse signal.

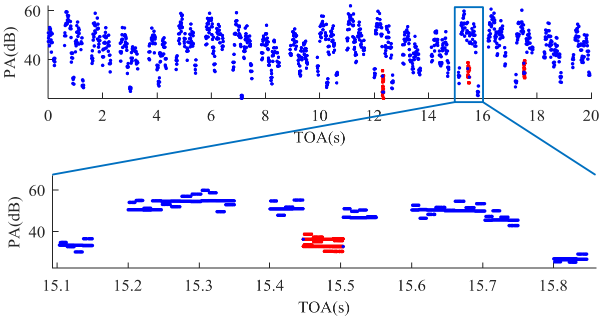


FIGURE 14. Classification results of overlapping and non-overlapping pulses with jitter, missing, and false alarm.

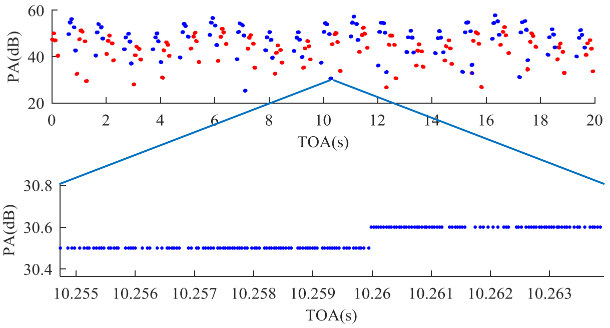


FIGURE 16. This figure shows the sorting results with jitter, missing, and false alarm.

of this paper is explored with pulse jitter, missing, and false alarm. In this section, we set the jitter of the pulse is 20%, and the missing pulses are 20%, and the false alarms are 10%. The pulse data is shown in Fig.13. First, the pulses are classified. The classification results of the overlapping pulses and the non-overlapping pulses are shown in Fig.14, in which the blue points are nonoverlapping pulse clusters and the red points represent overlapping pulse clusters. It can be seen from the figure that the two types of pulse clusters can be distinguished. Because the basis of the classification is the change between the PA. Even if there are pulse jitter, missing, and false alarm, it will not affect the overall law of amplitude variation within the pulse cluster. Therefore, two types of pulse clusters can be well distinguished. Then we filter out the false alarm signal.

Non-overlapping pulse filtering: we take the first pulse as the reference pulse. Then, the difference value between subsequent pulses and the reference pulse is calculated in turn. If the difference is greater than the threshold T, we filter out the corresponding pulse. When we constantly get Q differences that are all greater than the threshold T, the reference pulses are updated. We repeat the above-mentioned steps until the last pulse. This progress is called $p_{nooverlop}$.

Overlapping pulse filtering: on the basis of the above, the second filtering of the false alarm signal is performed. First, we execute $p_{nooverlop}$. The deleted signal and the reserved signal can be obtained. Then, the deleted pulse signal is processed. The first pulse of the deleted pulse signal is used as a reference pulse. Execute $p_{nooverlop}$ again. The pulse signal after secondary filtering is obtained. Finally, we combine the two remaining pulses to get the filtered overlapping pulses. The result after filtering is as shown in Fig.15. Comparing Fig.13 with Fig.15, we can see that the proposed method can eliminate the false alarm signal very well. Fig.16 presents the results for sorting the pulse signal with jitter, missing and false alarm. The sorting accuracy rate is 96.76%. One of the reasons why the sorting accuracy is still high when there is pulse jitter, missing and the false alarm is that the proposed

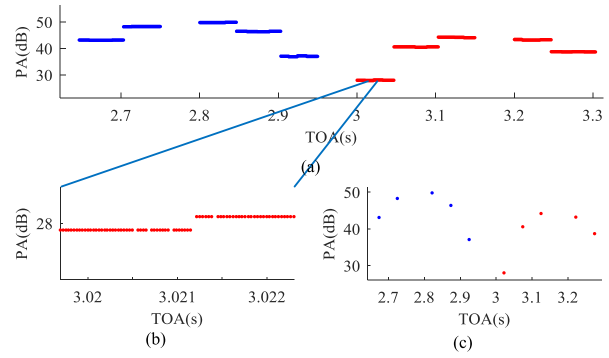


FIGURE 17. **Pulse clusters: (a) Pulse clusters with jitter and missing; (b) Partial enlargement; (c) Feature point.**

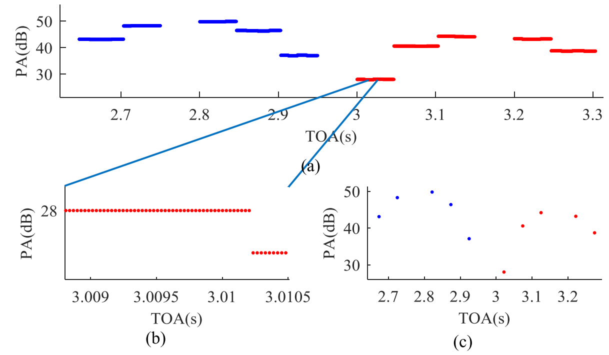


FIGURE 18. **Pulse clusters: (a) Pulse clusters of no jitter and missing; (b) Partial enlargement; (c) Feature point.**

method sorts signal in units of pulse clusters. As shown in Fig.17 and Fig.18, we have a cluster pulse as a feature point, whether there is jitter or missing. Another reason is that we have filtered out the false alarm signal. Therefore, it can be demonstrated that the sorting system is appropriate for pulse jitter, missing, and false alarm and has certain generalization ability.

V. CONCLUSION

In this paper, a novel method is proposed to address the problem of the same type radar emitter signals sorting. This method mines the information of the pulse train trajectory from the PA dimension and established a pulse sequence forecasting framework based LSTM. Then, we uses LSTM forecasting framework to predict the subsequent pulse of the current pulse. Compared with the traditional clustering and classification methods, the biggest advantage of the method is to use the inertia of the pulse sequence data itself, i.e. the trajectory feature. The method achieves a good sorting effect even when the multidimensional parameters are the same. Compared with [11], DNN, RNN, and GRU, the correct rate of sorting has been improved by 56.11%, 25.91%, 11.92%, and 7.37%, respectively. According to the robustness experiment, when the pulse sequence has jitter, missing, and false alarm, it does not affect the sorting performance of the method. Therefore, it has been proved that the proposed method can sort the same type radar signals with high accuracy and has good robustness. When the scenario is very complex, the proposed method sorting performance may be reduced. In future work, we will evaluate the method in more complex scenarios and optimize the algorithm model proposed.

REFERENCES

[1] H. K. Mardia, "New techniques for the deinterleaving of repetitive sequences," *IEE Proc. F Radar Signal Process.*, vol. 136, no. 4, pp. 149–154, Aug. 1989.

[2] D. J. Milojević and B. M. Popović, "Improved algorithm for the deinterleaving of radar pulses," *IEE Proc. F-Radar Signal Process.*, vol. 139, no. 1, pp. 98–104, 1992.

[3] D. Nelson, "Special purpose correlation functions for improved signal detection and parameter estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1993, pp. 73–76.

[4] K. Nishiguchi and K. Masaaki, "Improved algorithm for estimating pulse repetition intervals," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 2, pp. 407–421, Apr. 2000.

[5] R. Zhao, D. Long, X. Ping, and Y. Chen, "Plane transformation for signal deinterleaving in dense signal environment," (in Chinese), *Acta Electron. Sinica*, vol. 26, no. 1, pp. 77–82, 1998.

[6] W. H. Yang and M. G. Gao, "The deinterleaving of pulse signal based on plane transformation," (in Chinese), *J. Beijing Inst. Technol.*, vol. 25, no. 2, pp. 151–154, 2005.

[7] L. Yin, P. Jifei, and J. Qiuxi, "A study on sorting of radar-signals based on fuzzy clustering," (in Chinese), *Fire Control Command Control*, vol. 39, no. 2, pp. 52–54, 2014.

[8] X. Liu and X. Si, "Sorting of radar-signals based on modified fuzzy clustering," (in Chinese), *J. Projectiles Rockets Missiles Guid.*, vol. 29, no. 5, pp. 278–282, 2009.

[9] Q. Guo, C. Wang, and Z. Li, "Support vector clustering and type-entropy based radar signal sorting method," (in Chinese), *J. Xian Jiaotong Univ.*, vol. 44, no. 8, pp. 63–67, 2010.

[10] Q. Guo, C. H. Wang, L. M. Guo, Z. Li, Z.-W. Zhu, and Y. Hua, "Application of segment clustering in radar signal sorting," (in Chinese), *J. Beijing Univ. Posts Telecommun.*, vol. 31, no. 2, pp. 132–136, 2008.

[11] Q. Guo, P. Nan, and J. Wan, "Signal classification method based on data mining for multi-mode radar," *J. Syst. Eng. Electron.*, vol. 27, no. 5, pp. 1010–1017, 2016.

[12] J. Wan, S. Xiaoquan, H. Fukan, and Z. Liangzhu, "The application of neural network for sorting signal in radar," (in Chinese), *Syst. Eng. Electron.*, no. 7, pp. 29–35, 1996.

[13] Z.-Y. Lin, G. Liu, and G.-X. Dai, "Application of kohonen neural network to sorting radar multi-target," (in Chinese), *J. Air Force Eng. Univ. (Natural Sci. Ed.)*, vol. 4, no. 5, pp. 56–59, 2003.

[14] C.-M. Lin, Y.-M. Chen, and C.-S. Hsueh, "A self-organizing interval type-2 fuzzy neural network for radar emitter identification," *Int. J. Fuzzy Syst.*, vol. 16, no. 1, pp. 20–30, 2014.

[15] Z.-X. Shi, H. Gu, and S.-L. Zhu, "Analysis and feature extraction of amplitude information in full pulse data," (in Chinese), *Command Control Simul.*, vol. 31, no. 3, pp. 43–45, 2009.

[16] B. Barshan and B. Eravci, "Automatic radar antenna scan type recognition in electronic warfare," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 4, pp. 2908–2931, Oct. 2012.

[17] D. Zeng, J. Zhang, Z. Zheng, X. Chen, and F. Xu, "A radar signal sorting method using parabola Hough transform," (in Chinese), *Aerosp. Electron. Warfare*, vol. 33, no. 4, pp. 22–25, 2017.

[18] Q. Guo, "Signal sorting methods for unknown radar emitters in complex environments," (in Chinese), Ph.D. dissertation, College Inf. Commun. Eng., Harbin Eng. Univ., Harbin, China, 2007.

[19] Y. Goldberg, "Neural network methods for natural language processing," *Synth. Lect. Hum. Lang. Technol.*, vol. 10, no. 1, pp. 1–309, 2017.

[20] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. 11th Annual Conf. Int. Speech Commun. Assoc.*, Chiba, Japan, Sep. 2010, pp. 1045–1048.

[21] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proc. IEEE Int. Conf. Acoust.*, May 2011, pp. 5528–5531.

[22] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

[23] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[24] J. F. Kolen and S. C. Kremer, "Gradient flow in recurrent nets: The difficulty of learning longterm dependencies," in *A Field Guide to Dynamical Recurrent Networks*. Piscataway, NJ, USA: IEEE Press, 2001.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, 2000.

[27] T. Chen, T. H. Wang, and L. M. Guo, "Sequence searching methods of radar signal pulses based on pri transform algorithm," (in Chinese), *Syst. Eng. Electron.*, vol. 39, no. 6, pp. 1261–1267, 2017.

[28] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015.

[29] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 257–269, 2010.

[30] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012. [Online]. Available: http://arxiv.org/abs/1212.5701

[31] H. An and L. Yang, *Modeling and Simulation of Rardar and Electronic Warfare Systems*. Peking, China: National Defense Industry Press (in Chinese), 2017.

[32] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[33] D. L. Marino, K. Amarasinghe, and M. Manic, "Building energy load forecasting using deep neural networks," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2016, pp. 7046–7051.

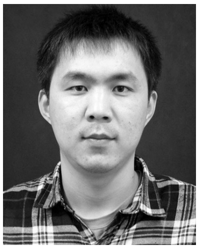
[34] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

QIANG GUO received the B.S., M.S., and Ph.D. degrees in information and communication engineering from Harbin Engineering University, China, in 1994, 2003, and 2007, respectively.

In 2009, he joined the Faculty with the School of Information and Communication Engineering, Harbin Engineering University, where he is currently a Full Professor. His research interests include electronic countermeasure, machine learning, radar signals sorting, and recognition.

He is also a Review Expert of the Science and Technology Commission of the Military Commission, the Degree and Postgraduate Education Center of the Ministry of Education, the National Natural Science Foundation of China, and the National Science and Technology Award. In 2009, he received the National 100 Excellent Doctoral Degree Dissertation Candidate Nomination.

LONG TENG received the B.E. degree in electronic information engineering from the Shaanxi University of Technology, Shaanxi, China, in 2016. He is currently pursuing the Ph.D. degree in information and communication engineering with Harbin Engineering University. His research interests include radar signals sorting and deep learning.



LIANGANG QI received the B.S. and Ph.D. degrees in information and communication engineering from Harbin Engineering University, China, in 2013 and 2018, respectively. He has been a Lecturer with Harbin Engineering University, since 2018. His research interests include electronic countermeasure, machine learning, radar target recognition, and radar signal processing. He is currently a Reviewer of IEEE ACCESS.

XIAOWEI JI received the B.E. degree in electronic information engineering from Northeast Forestry University, Harbin, China, in 2017. She is currently pursuing the M.S. degree in information and communication engineering with Harbin Engineering University. Her research interests include hyperspectral image processing, machine learning, and radar signal processing. She is currently a Reviewer of IEEE ACCESS.

JIANHONG XIANG was born in Harbin, China, in 1977. He received the M.S. degree and the Ph.D. degree in information and communication engineering from Harbin Engineering University, Harbin, in 2009. Since 2010, he has been an Associate Professor with the College of Information and Communication Engineering, Harbin Engineering University. He is also the author of one book and more than 30 articles. He holds three patents. His research interests include image processing, adaptive signal processing, communication signal processing, massive MIMO, satellite communication, and smart antenna. He is also a Reviewer of the *Journal of Systems Engineering and Electronic Technology*. He received the Heilongjiang Province Science and Technology Progress Prize, in 2014.