Radar Emitter Identification Based on Stacked Long and Short Term Memory

Lei Meng¹,²,*, Wei Qu¹, and Kai Cai²
¹Department of Electronic and Optical Engineering, Space Engineering University, Beijing, China
²Beijing Institute of Remote Sensing, Beijing, China

Abstract. With the increasing complexity of electromagnetic environment and the rising of operating patterns of new radars, emitter identification is becoming more and more difficult. This paper presents a radar emitter identification method based on stacked long and short term memory (SLSTM). Radar pulse train can be directly used as input without extracting other features, which greatly simplifies the data preprocessing and realizes the "end-to-end" recognition of radar emitter signal. The timing characteristics of the pulses are automatically extracted by SLSTM, and the optimal network parameters are trained to complete radar signal identification. Compared experiments with conventional methods are conducted, and the results show that the proposed model outperforms other existing techniques. Moreover, simulation experiments in different noise and loss pulse environment show that the method is effective and robust in solving problems of radar emitter recognition.

1 Introduction

Radar emitter recognition (REI) is an essential part of radar reconnaissance system, whose results will have a major impact on modern electronic countermeasure. With the development and progress of radar technology, multifunctional radar represented by phased array radar is widely used, whose signal modulation is complex and functions are various, which brings challenges to REI.

Traditional REI methods generally include data preprocessing, feature extraction and classification recognition. First, the signals are sorted to obtain different single radar signals, then the identification characteristics of radar signal are extracted, and finally the classification and identification are carried out using machine learning algorithms such as K-nearest neighbor, decision tree and support vector machine [1-3]. The current challenges of traditional REI methods mainly include two aspects. First, it requires manual design and selection of recognition features, and the quality of the features will directly affect the recognition effect. Second, the recognition rate and noise resistance of the traditional machine learning classification algorithm for complex radar signals need to be improved.

With the widely application of the deep learning techniques, the recurrent neural networks (RNNs) have shown an amazing ability in speech recognition, machine translation and other sequential data processing. In particular, the long and short term memory (LSTM) [4] and other improved models which overcome the problem such as gradient disappeared, have been widely used in engineering [5-7].
(PDW) is a common representation of radar pulse signal. The radar pulse train is essentially a kind of time series data, which is suitable for processing with RNN. In this paper, a REI method based on stacked long and short term memory (SLSTM) is proposed. The method uses the sorted radar pulse train as input, whose dimension is highly matched with the network input dimension. The timing characteristics of the pulses are automatically extracted by SLSTM, and the optimal network parameters are trained to complete radar signal identification. Radar pulse train can be directly used as network input without extracting other features, which greatly simplifies the data preprocessing and realizes the “end-to-end” recognition of radar emitter signal. This method is more concise in flow structure, easy to be realized in engineering, and its high timeliness is more in line with the military application requirements of REI.

2 Long and short term memory

RNNs are the type of deep NN with a “feedback” module, which are “deep” in temporal dimension and have been extensively applied in time sequence issues. In traditional RNN, the output of previous time is looped back and used as additional input of the current state. It has the capability to learn contextual information and keep “memory” of the whole sequence.

However, traditional RNN cannot handle temporal sequence with long period since there will be vanishing gradient problem [8]. In order to overcome the aforementioned issues, LSTM network is developed to introduce memory blocks to store sequential information of long-term flow.

These memory blocks are recurrently connected, which are combined with multiplication gates and memory cells. Three multiplicative gate units, which are input, output, and forget gates, determine the amount of dynamic information entering, leaving, and updating the memory cell, respectively. The input gate and the output gate measure the input and the output of the cell, while the forget gate measures the internal state (the hidden activation of the memory cell) [9]. The structure is depicted in figure 1, where the t-th state in LSTM can be written as follows.

\[
i_t = \sigma(x_t \cdot W_{si} + h_{t-1} \cdot W_{hi} + b_i) \quad (1)
\]

\[
f_t = \sigma(x_t \cdot W_{sf} + h_{t-1} \cdot W_{hf} + b_f) \quad (2)
\]

\[
o_t = \sigma(x_t \cdot W_{so} + h_{t-1} \cdot W_{ho} + b_o) \quad (3)
\]
\[ g_t = \tanh(x_t \cdot W_{sg} + h_{t-1} \cdot W_{hg} + b_g) \] (4)

\[ c_t = f_t \otimes c_{t-1} + i_t \otimes g_t \] (5)

\[ y_t = h_t = o_t \otimes \tanh(c_t) \] (6)

Where \( W_{sx} \) and \( W_{hx} \) (\( \ast \) denotes \( g, i, f \), or \( o \)) are coefficient matrices with respect to the current input \( x_t \) and previous hidden activation \( h_{t-1} \), and \( b \) is the bias vector. \( \sigma(\cdot) \) stands for a logistic sigmoid function, \( \tanh(\cdot) \) denotes a tangent function, and \( \otimes \) stands for point-wise multiplication.

### 3 Emitter recognition framework based on SLSTM

In this section, we come up with a radar emitter recognition architecture based on stacked LSTM, where radar pulse train is the input of network without extracting other features. SLSTM was used to extract the deep features of the pulse train automatically. The framework is depicted in Fig. 2. It consists of input layer, hidden layer, output layer and training module. The input layer mainly preprocesses the sorted radar pulse train to create the data set available in the network. The hidden layer consists of \( k \) LSTM layers, and the output of the \( k \)-th LSTM layer’s last time step is connected to the output layer as input. The output layer is composed of the full connection layer whose activation function is softmax. The output of the hidden layer is connected to the full connection layer. The softmax function will output the classified results. The training module uses back propagation through time (BPTT) algorithm [10] to train the network.

![Radar emitter identification model based on stacked LSTM](image)

**Figure 2.** Radar emitter identification model based on stacked LSTM.

Compared with the algorithm structure shown in Figure 2, the identification method includes 4 steps: data preprocessing, network building, model training and prediction.
3.1 Data preprocessing

The input of the network is the radar pulse train, and the output is the radar type corresponding to the pulse train. Each pulse train is taken as a training sample. The pulse train contains an unequal number of pulses, which is characterized by radio frequency (RF), pulse repetition interval (PRI), and pulse wide (PW).

The pulse train is denoted as

\[ P = \{p_1, p_2, \ldots, p_i, \ldots, p_k\}, \quad i \in [1, k] \]  \hspace{1cm} (7)

Where \( k \) is the length of the pulse train. The \( i \)-th pulse can be denoted as

\[ p_i = \{rf_i, pri_i, pw_i\} \]  \hspace{1cm} (8)

Data preprocessing is the main task of the input layer. The radar pulse train with different lengths should be converted into the data set available in the network according to the model structure. Data preprocessing includes sample length uniform, feature normalization and converting the category label of the sample to one-hot encoding.

3.1.1 Sample length uniform

Suppose the time steps of the hidden layer is \( N \), then the sample length of the model input should also be \( N \). Due to the influence of various factors during the receiving process, the length of the pulse train is not fixed, so the length of the sample data needs to be uniformly transformed to \( N \). Take the sample shown in equation (7) to illustrate.

When \( k < N \), the sample is copied and repeatedly spliced. The method is as follows: copy the sample into \( l \), \( l \) satisfies \( k \cdot (l - 1) < N < k \cdot l \), then join \( l \) samples together to make the length of pulse train expand to \( k \cdot l \), and cut the first \( N \) pulses to get the sample suitable for model input

\[ P' = \{p_1, \ldots, p_k, p_1, \ldots, p_k, \ldots, p_{N-k(l-1)}\} \]  \hspace{1cm} (9)

When \( k > N \), the sample should be segmented with a step size of \( N \). The sample length was expanded to \( m \cdot N \) by copying and splicing method, \( m \) satisfies \( N \cdot (m - 1) < k < N \cdot m \). Then divide the sample into \( m \) pulse train with length \( N \) in accordance with time order. Finally, \( m \) samples are obtained.

\[ P_j' = \{p_{N(j-1)+1}, \ldots, p_{N(j-1)+N}\}, \quad 0 < j < m \]  \hspace{1cm} (10)

\[ P_m' = \{p_{N(m-1)+1}, \ldots, p_k, p_1, \ldots, p_{N-m-k}\} \]  \hspace{1cm} (11)

3.1.2 Feature normalization

Feature normalization is a necessary operation before the training of machine learning and deep learning, mainly to eliminate the influence of too large difference in the proportion of numerical attributes of input data. Normalization is the transformation of a class of features so that their mean value is 0 and their standard deviation is 1. The specific method is to subtract the mean value of all elements in the feature from each element and divide it by the
standard deviation. Therefore, the three characteristics of RF, PRI and PW of all samples are standardized respectively.

3.2 Network building

In this Step, the main parameters of the network should be set, including time steps, network hierarchy, number of neurons, loss function, optimizer configuration, etc.

The number of time steps is the number of time steps after the RNN is expanded according to time. It is one of the key parameters that affect the network performance. It also determines the network size and sample input length. The larger the number of time steps, the more complex the network structure, and the more comprehensive the timing characteristics of the pulse train can be learned by the model. The time steps setting should refer to the overall distribution of sample length to minimize the amount of preprocessing calculation. Also, avoid setting up too large, making the network too complex and difficult to train.

The hidden layer is the main part of the network. The number of LSTM layers and LSTM cells directly determine the capability and complexity of the model. The larger the number of LSTM layers and LSTM cells, the stronger the model capability and the faster the convergence rate of the loss function. The cost is that the training speed decreases and the risk of overfitting increases. How to set the number of hidden layers and the number of neurons has not been determined theoretically, and it often needs repeated attempts.

REI is a multi-classification, single-label problem. The neuron number of the full connection layer is equal to the number of radar categories. The activation function for the full connection layer is softmax function. It converts the input into the probability of each category, and outputs the category with the highest probability as the prediction category.

The loss function is multi-classification cross entropy, and its mathematical expression is:

\[
L = -\sum_{i=1}^{n} y_i \log \hat{y}_i
\]  

(14)

Where \( \hat{y} \) is the prediction of network, \( y \) is the label, \( i \) is the category ordinal number corresponding to classification, and \( n \) is the number of categories.

The optimizer configuration refers to which optimizer to use and the learning rate. There are many gradient-based optimization algorithms, such as Nesterov [11], AdaGrad [12], RMSProp [13], Adam [14] and so on. In this paper, Adam is selected as the optimizer. Adam algorithm can calculate the adaptive learning rate for different parameters, and take up less storage resources, thus achieving faster maturity and better learning effect. The learning rate parameter generally uses the default value to obtain the good performance, the specific adjustment according to the training situation.

3.3 Model training

After data preprocessing to obtain the sample set meeting the requirements of the network, the sample set is divided into training set, verification set and test set according to 6:2:2. The training set is used for the training model, the verification set is used for the evaluation model, and finally the trained model is tested with the test set.

The training set are input to the hidden layer and calculated step by step forward according to the time sequence, as shown in equations (1) ~ (6). The network learns the data characteristics layer by layer. The last time step of the \( k \)-th LSTM layer outputs the connected full layer, and the softmax activation function outputs the classification results.
The goal of model training is to minimize the loss function. BPTT algorithm is used to continuously iterate and update network parameters, and Adam optimization algorithm is used to accelerate and obtain the final model weights.

To reduce the risk of overfitting, dropout is the most commonly used and effective method. Yarin Gal's doctoral thesis proposed the correct method of using dropout in RNN. The same dropout mask should be used for each time step, and the dropout mask that does not change with time should be applied to the layer's internal loop activation [15]. It is important to note that dropout is only used during model training, not model testing.

3.4 Model prediction

After the optimal network parameters are obtained, the model can be used to predict and classify the new radar pulse train. According to the method in step 1 of the process, the data of the signal to be recognized is preprocessed. Then input it to the model, the model can output the classification results. It should be noted that the same transformation coefficient as the training data should be used when the characteristics of the predicted data are standardized.

4 Simulation

4.1 Simulation settings

The data set was generated by Python simulation. With radar model as the label, there were 13 types of radar. The specific parameters were shown in Table I. The radar signal characteristic parameters are RF, PRI, PW and its modulation mode. Each type of radar generates 20,000 pulse train samples, including uniform distribution of working mode. The sample set includes 80,000 samples totally. For simplicity of processing, the generated pulse train length is unified to 128.

We define the concept of error deviation level (EDL) [16] and ratio of dropped pulses (RDP) as follows.

$$\text{EDL} = \left| \frac{\xi}{x_i} \right| \times 100\%$$  \hspace{1cm} (15)

Where $x_i$ is the real value of parameter, $\xi_i$ is the measuring error of parameter.

$$\text{RDP} = \frac{n_i}{N_i} \times 100\%$$ \hspace{1cm} (16)

Where $N_i$ is the number of pulses contained in the pulse train, $n_i$ is the number of pulses that are missing from the pulse train.

The sample set is divided into training set, verification set and test set according to 6:2:2. To meet the requirement of algorithm robustness, 20% EDL and 20% RDP were added to the test data set.

4.2 The effect of network parameters

We should set up the key parameters of the network first, including the number of stacked LSTM layers and the nodes of each LSTM layer. We choose the number of LSTM layers from one to four layers. For simplicity, the number of nodes in each layer is set to be the
same. It is chosen from \{64, 128, 192, 256\}. The models with different LSTM layers and LSTM nodes were used for training and testing. The Monte Carlo experiment was repeated 50 times for each parameter combination, the average recognition accuracy was taken as the result. The results are shown in table 2, wherein the vertical axis is the number of LSTM layers and the horizontal axis is the neuron number of each LSTM layer.

**Table 1.** Parameters settings of radar emitter.

<table>
<thead>
<tr>
<th>Radar Type</th>
<th>RF(MHz)</th>
<th>PRI(μs)</th>
<th>PW(μs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[5450,5800] constant</td>
<td>379/401/431/453 staggered</td>
<td>0.1 constant</td>
</tr>
<tr>
<td>2</td>
<td>[5450,5800] constant</td>
<td>770/882/808/906 staggered</td>
<td>0.25 constant</td>
</tr>
<tr>
<td>3</td>
<td>[5450,5800] constant</td>
<td>1293/1373/1255/1411 staggered</td>
<td>1.0 constant</td>
</tr>
<tr>
<td>4</td>
<td>[5300,5500] constant</td>
<td>244/399/431/362/379/278/317/232</td>
<td>3 constant</td>
</tr>
<tr>
<td>5</td>
<td>[5300,5500] constant</td>
<td>504/538/622/560/664/711 Group change(8)</td>
<td>7 constant</td>
</tr>
<tr>
<td>6</td>
<td>[5300,5500] constant</td>
<td>1360/1046/1234/1181/1287 Group change(5)</td>
<td>11 constant</td>
</tr>
<tr>
<td>7</td>
<td>[3150,3400] agility</td>
<td>746/683/623;880/850/940;1060/1150/1100;1300/1350/1410 staggered</td>
<td>6/12/29/50 Group change(3) constant</td>
</tr>
<tr>
<td>8</td>
<td>[3100,3500] constant</td>
<td>5000/10000/15000/20000 jitter (5%)</td>
<td>55 constant</td>
</tr>
<tr>
<td>9</td>
<td>3130/3220/3170/3300/3260 Group change(8)</td>
<td>510/685/858/1022/1301 Group change(8)</td>
<td>3/6/12/29/50 Group change(8) constant</td>
</tr>
<tr>
<td>10</td>
<td>[3100,3500] agility</td>
<td>[1150,1450] constant</td>
<td>3/6/12/29 optional Group change(8)</td>
</tr>
<tr>
<td>11</td>
<td>[3000,3400] constant</td>
<td>248/324/392/316/373/425 Group change(8)</td>
<td>15/20/25 Group change(8)</td>
</tr>
<tr>
<td>12</td>
<td>3120/3200/3270 Group change(10)</td>
<td>302/378/432 Group change(10)</td>
<td>15/20/25 Group change(10)</td>
</tr>
<tr>
<td>13</td>
<td>[3100,3300] agility</td>
<td>248/324/392/316/373/425 staggered</td>
<td>15/20/25 optional</td>
</tr>
</tbody>
</table>

Note: in “Group change(m)”, m is the number of pulses in the group.

**Table 2.** The overall recognition rate at different LSTM layers and LSTM neurons.

<table>
<thead>
<tr>
<th>parameter</th>
<th>64</th>
<th>128</th>
<th>192</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9322</td>
<td>0.9660</td>
<td>0.9507</td>
<td>0.9256</td>
</tr>
<tr>
<td>2</td>
<td>0.9596</td>
<td><strong>0.9871</strong></td>
<td>0.9664</td>
<td>0.9456</td>
</tr>
<tr>
<td>3</td>
<td>0.9403</td>
<td>0.9447</td>
<td>0.9026</td>
<td>0.8869</td>
</tr>
<tr>
<td>4</td>
<td>0.9384</td>
<td>0.9485</td>
<td>0.8241</td>
<td>0.7709</td>
</tr>
</tbody>
</table>

We can get the best parameter from Table II when the number of stacked LSTM layers is 2 and each neuron number is 128. These parameters will be used in the next experiments. When the LSTM layers and LSTM neuron are too large, the network is prone to overfitting.
4.3 The effect of Interference

In this section, we discuss the influence of interference including EDL and RDP on the recognition rate. Aimed to test the performance of the proposed SLSTM method, the comparison with decision tree (DT) and BP neural network (BPNN) models were conducted. The DT is based on information gain. The BPNN consists of two hidden layers, the neuron number of hidden layer is 200 and 100 respectively, and the optimization algorithm is Adam. We make the average value as our result from 100 random runs using the highest recognition rate.

We used the training set to train the three classifiers and obtained the trained model, and then added different interference to the test set for testing.

4.3.1 The effect of EDL

In this experiment, we added EDL from 0% to 50% to the test set respectively. Figures 3 represent the different methods of recognition accuracy varying with EDL conditions. As can be seen from Figures 3, the performance of DT is worse than the other methods, and the overall effect of our proposed method is the best of all methods with recognition accuracy, it is still able to reach more than 90% in the case of EDL=50%. BPNN has good recognition effect when the EDL is low, and when the EDL gradually increases to 50%, the recognition rate of BPNN gradually decreases to about 80%.

![Figure 3. Recognition accuracy varying with EDL.](image)

4.3.2 The effect of RDP

In this experiment, the EDL is fixed at 10%. Because the case of missing is similar to false pulse, here we only discuss the case of missing pulses. We observe the effect of RDP on the recognition rate of three methods. Other parameters of three methods here are the same as the above experiment. The results shown in Figure 4. We didn’t take any treatment for the loss of pulse, but directly enter the data with missing pulse in the classifier.

![Figure 4. Recognition accuracy varying with RDP.](image)
The proposed method has the best recognition effect among the three methods. When the RDP is large, even up to 50%, the method we proposed still has more than 90% recognition accuracy. The recognition accuracy of DT decreases slowly with the increase of RDP, and the overall recognition effect is good. While the recognition accuracy of BPNN drops sharply when the RDP is more than 25%, even worse than the DT.

From the figures which compared three methods, we can conclude that BPNN is greatly affected by pulse loss and DT is greatly affected by measurement error. The proposed method is superior to the other two methods under various interference conditions, it is not sensitive to the missing pulse or measurement error. Especially when EDL or RDP reaches 50%, the recognition accuracy is still above 90%. Compared to the other two algorithms, this paper shows that the proposed method is robust and can be able to adapt to the harsh radiation signal environment.

5 Conclusion
This paper presents a radar emitter identification method based on stacked long and short term memory(SLSTM), which greatly simplifies the data preprocessing and realizes the "end-to-end" recognition of radar emitter. The method flow is introduced in detail. Simulations show the validated effectiveness compared with other two traditional methods. Then we also learned the effect of measurement error and missing pulse to the recognition accuracy. The result of our approach provide better identification performance than DT and BPNN. In summary, the proposed method has advantages of high accuracy and excellent robustness.

References


