remote sensing

MDPI

Communication

Temporal Feature Learning and Pulse Prediction for Radars with Variable Parameters

Shuo Yuan [1,*] and Zhang-Meng Liu [1,2]

[1] State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System, National University of Defense Technology, Changsha 410073, China
[2] Tianjin Institute of Advanced Technology, Tianjin 300450, China
* Correspondence: ysyuanshuo@nudt.edu.cn

Abstract: Many modern radars use variable pulse repetition intervals (PRI) to improve anti-reconnaissance and anti-jamming performance. Their PRI features are probably software-defined, but the PRI values at different time instants are variable. Previous statistical pattern analyzing methods are unable to extract such undetermined PRI values and features, which greatly increases the difficulty of Electronic Support Measures (ESM) against such radars. In this communication, we first establish a model to describe the temporal patterns of software-defined radar pulse trains, then introduce the recurrent neural network (RNN) to mine high-order relationships between successive pulses, and finally exploit the temporal features to predict the time of arrival of upcoming pulses. In the simulation part, we compare different time series prediction models to verify the RNN's adaptability for pulse sequences of variable parameter radars. Moreover, behaviors of different RNN units in this task are compared, and the results show that the proposed method can learn complex PRI features in pulse trains even in the presence of significant data noises and agile PRIs.

Keywords: pulse repetition interval (PRI) feature; variable parameter radar (VPR); recurrent neural network (RNN); self-supervised learning; parameter prediction

check for updates

Citation: Yuan, S.; Liu, Z.-M. Temporal Feature Learning and Pulse Prediction for Radars with Variable Parameters. *Remote Sens.* **2022**, *14*, 5439. https://doi.org/10.3390/rs14215439

Academic Editors: Silvia Liberata Ullo, Alfonso Farina, Yu Yao, Harun Taha Hayvaci and Pia Addabbo

Received: 13 September 2022
Accepted: 27 October 2022
Published: 29 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Variable parameter radar (VPR) usually refers to radars with variable inter-pulse time intervals, which is an important means of countering reconnaissance and jamming [1,2]. For a multi-function radar with variable parameters, whenever it shifts between different work modes, its pulse repetition intervals (PRIs) will change agilely within a certain range, and the changed PRIs can hardly be predicted by previous ones. This parameter variability places a great burden on Electronic Support Measures (ESM) and Electronic Counter Measures (ECM). However, for software-defined radars, a stable functional relationship hides in multiple PRIs of the same working mode and does not change with the agility of parameters, which is called PRI feature.

PRI feature is of great value to signal processing because it is a functional law about radar working mode. First, the known PRI feature can be used to predict subsequent pulses based on already intercepted pulses, which can be applied to online pulse deinterleaving in electronic support systems [3]. Second, we can identify the working modes of radar by judging PRI feature in the intercepted pulse train [4]. Although the temporal parameters of VPR are variable, PRI feature used to generate data at the bottom still shows stability. Based on the wide applications of PRI features, effective methods should be designed to learn them from intercepted pulse trains.

In the past, PRI information was learned mainly through statistical methods such as PRI estimation [5–9] and PRI pattern extraction [10–12]. These methods mine PRI values or pulse group structures in intercepted pulse trains mainly from a statistical perspective and have a good application effect for radar with fixed parameters. However, they ignore high-dimensional PRI patterns hidden among successive PRIs and are not suitable for VPRs

having connotative PRI patterns. In recent years, neural networks have been introduced to process pulse data and have gained satisfying performance in many pulse deinterleaving and recognition tasks [13–21]. Liu recognized radars using the RNN at an early time, and he also proposed a state clustering method to identify MFR working modes [13,14]. In [15], a seq2seq LSTM was trained by fine-labeled pulse trains to identify working modes and their conversion boundaries. The Convolutional Neural Network could also be trained to recognize the signal waveform [16]. In [17], a multi-label classifier was designed using deep learning techniques for recognizing compound signals. In the deinterleaving field, Li utilized autoencoders to recover the original pulse train from its counterpart that was contaminated by data noises or interleaved with other signals, and finally realized the denoising and deinterleaving of pulse trains [18,19]. Zhu formulated temporal structures of radiation sources mathematically and abstracted the deinterleaving problem as the minimization of a cost function, which was solved by a trained Seq2seq network [20]. However, most of these neural network-based methods require sufficient manual tagged data to facilitate supervised learning, and the designed and learned model can only be used in specific application tasks, which weakens the practicability of neural networks in ESM.

In this communication, we first model the PRI feature in the software-defined pulse train as a high-dimensional function. Then, a self-supervised learning method is proposed through building an RNN structure to learn the PRI feature function in the unlabeled pulse trains. Finally, we carry out simulations to verify the performance of the proposed method in PRI feature learning and variable parameter prediction.

The main contributions of this communication are as follows:

(1) The PRI feature hidden in the pulse train is formulated with a nonlinear function;
(2) A self-supervised framework for learning PRI features is proposed based on RNN;
(3) Temporal parameter prediction within pulse groups is realized based on the PRI feature function stored in the trained RNN;
(4) The effects of many factors on PRI feature learning and prediction are investigated, including the agility range of PRIs, data noises in training pulse trains, and different recurrent units of RNNs.

2. Pulse Train Description

PRI is defined as the time interval between radar pulses, and it is the temporal element of pulse groups and pulse trains. Pulse groups constituted by different types of PRIs determine the working mode executed by a radar. For example, fire-control radars and airplane pulse-Doppler radars utilize constant PRIs that fall into different ranges to search targets at different distances and utilize stagger or jitter PRIs for ambiguity elimination and jamming deception [21,22]. Additionally, some advanced radars apply agile PRIs to escape from jamming [2]. Thus, extracting and utilizing PRI-related information has attracted great attention in the field of passive signal processing.

Each intercepted radar pulse can be represented by its parameters such as time of arrival (TOA) and radio frequency (RF) [23]. PRI features hide in the TOA sequence. In this communication, TOA is taken as the basic observation while other parameters are assumed to be fixed, so as to simplify data illustration and method description. However, relevant results can be extended directly to radars with jointly changing parameters.

PRIs of a VPR at different time instants are variable. However, since many radars are software-defined to transmit pulses, PRIs during the same working mode are not independent of each other, but with a functional relationship, [24] called the PRI feature function. The pulse generation process of the VPR is shown in Figure 1. When a VPR transmits pulses to execute a certain working mode, it first generates an initial PRI randomly from a certain agility range, and other PRIs are determined by the PRI feature function. More concretely, if $f^{(i)} = \left[x, f_1^{(i)}, f_2^{(i)}, \cdots, f_{N-1}^{(i)}\right]$ represents the PRI feature function set of the i-th working mode of a VPR, its PRI sequence is generated as:

$$p = f^{(i)}(PRI) = \left[PRI, f_1^{(i)}(PRI), f_2^{(i)}(PRI), \cdots, f_{N-1}^{(i)}(PRI)\right] \quad (1)$$

where $PRI \in [PRI_{\min}, PRI_{\max}]$ and is assumed to be uniformly distributed within this scope without loss of generality, $PRI_{\max}$ and $PRI_{\min}$ are the upper and lower bounds of the initial PRI, respectively. N represents the number of PRIs in the pulse sequence. For example, many radars use constant PRIs to search targets but have different PRIs each time they are powered on [25], so it can be regarded as a VPR. The PRI feature function of such VPR is $f_1^{(i)} = f_2^{(i)} = \cdots = f_{N-1}^{(i)} = x$.

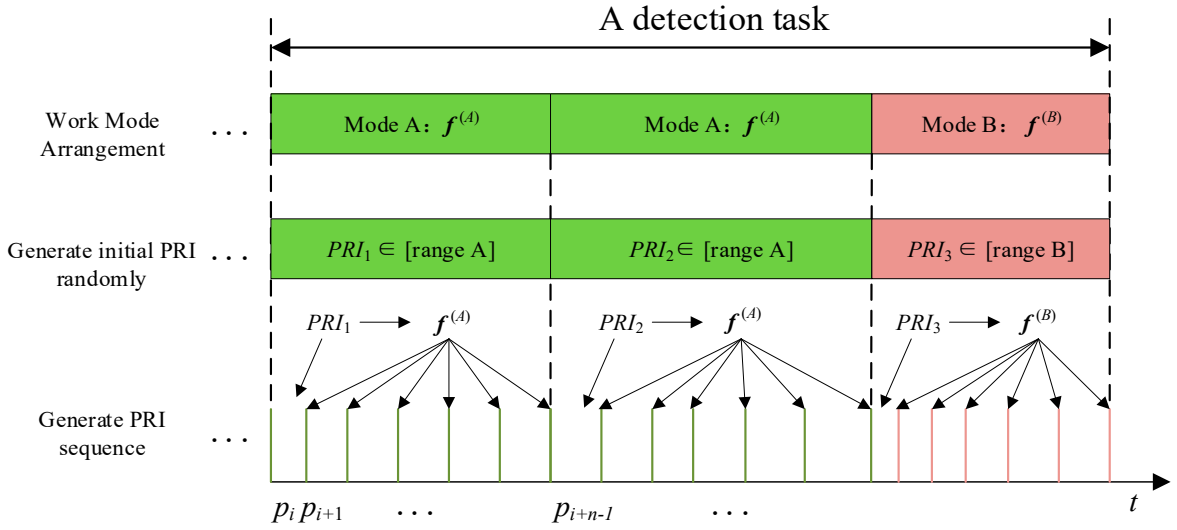


Figure 1. Schematic diagram of pulse train generated by variable parameter radar.

In non-cooperative interception systems, the pulse train is initially represented by the TOA sequence, and inter-pulse differential time of arrival (DTOA) is usually calculated for PRI information mining. Without data noise, the first-order DTOA sequence is expressed as:

$$d = p = [d_1, d_2, \cdots, d_n, \cdots, d_N] \tag{2}$$

According to the process of VPR generating pulse train, PRI feature function can be exploited to predict upcoming DTOAs based on previous ones, which can be shown as:

$$d_{n+1} = g^{(i)}(\sum_{i=0}^{n-1} d_i, d_n) \tag{3}$$

where $g^{(i)}(\cdot)$ represents the PRI feature function of the i-th working mode, which unifies the functions at different positions in $f^{(i)}$; d_n represents the n-th DTOA in Equation (2), and $d_0 = 0$ indicates the first pulse; $\sum_{i=0}^{n-1} d_i$ represents the DTOA between the $(n-1)$-th and the first pulses to indicate the time state of $(n-1)$-th pulse in pulse sequence. When $d_0 = 0$ and d_1 are input, the DTOA sequence could be predicted one by one according to the known PRI feature function $g^{(i)}(\cdot)$. We have ignored TOA measurement errors and will not consider them until the simulation part, so that we can focus on methodological contributions in the text.

In practical applications, the measured DTOA sequences are contaminated by data noises, such as missing pulses and interferential ones. As a result, the determined prediction output in Equation (3) should be modified to a probabilistic one to better describe noisy DTOA sequences as follows:

$$d_{pre} = \sum_{j=n+1}^{N} \rho^{j-n-1}(1-\rho)\delta(d - \sum_{m=n+1}^{j} d_m), d_m \in d \tag{4}$$

where $\rho \in [0,1)$ denotes the missing rate of pulses that are missed randomly and independently, δ is the impulse function, and d_{pre} represents the predicted DTOA of the next pulse.

When the number of missing pulses increases, the probability that the corresponding DTOA emerges decreases. Moreover, the prediction probability of interferential pulses is 0 due to the impulse characteristic of the prediction in Equation (4) and the random emergence of interferential pulses. In particular, the result of predicting the first DTOA without preceding pulses is:

$$d_{pre} = g(0,0) = U(PRI_{\min}, PRI_{\max}) \tag{5}$$

that is, the first predicted DTOA conforms to the uniform distribution of the initial PRI range.

The PRI feature function g known in advance can be used to predict PRI parameters and further support applications of signal deinterleaving and working mode recognition. However, it is a very demanding task to rebuild the DTOA prediction function from non-cooperatively intercepted radar pulses due to various data noises. In this communication, we will work out a way to learn the approximate PRI feature function $\hat{g}$ from intercepted pulse trains.

3. PRI Feature Learning

Due to the non-cooperation of interception, there is no tag indicating working modes, true or interferential pulses, etc. Thus, one has to realize the mining of PRI feature functions $\hat{g}$ following a guideline of self-supervised learning.

The first-order DTOA sequences of intercepted pulse trains are used as inputs for rebuilding PRI feature functions. However, as intercepted pulse trains are contaminated with interferential pulses and missing ones, the function learning process should overcome some difficulties caused by various data noises. First, the process should be statistically robust against interferential pulses, which split true DTOAs irregularly due to their random occurrence times. Second, the process should also be comprehensive for missing pulses by adapting to high-order DTOAs other than expected first-order PRIs. Actually, we have taken missing pulses into account by formulating the output of g as a probabilistic model in (4), and all the work left to us is finding out a way to approximate the PRI feature function g by feeding noisy DTOA sequences to a model learning machine.

When the DTOA sequence is input sequentially, a good approximation of the PRI feature function, denoted by $\hat{g}$, is expected to well predict the DTOA of the next pulse, i.e.,

$$DTOA_{n+1\,pre} = \hat{g}(\sum_{i=0}^{n-1} DTOA_i, DTOA_n) \tag{6}$$

However, any deviation of the estimated $\hat{g}$ from the real g leads to a bias between the predicted value and the real value of the next DTOA. Prediction loss is defined as the mean square error of each DTOA estimation to measure the difference between the estimated function and the real one, as presented below:

$$Loss = \frac{1}{N} \sum_{n=0}^{N-1} (DTOA_{n+1} - DTOA_{n+1\,pre})^2 \tag{7}$$

where N represents the number of DTOAs in the pulse sequence. The prediction loss can be used for optimizing the estimated function iteratively via back-propagation (BP) [26], as follows:

$$V^{(j+1)} = V^{(j)} + \eta \frac{\partial Loss}{\partial V} \tag{8}$$

where $V^{(j)}$ represents the parameters of the PRI feature function $\hat{g}$ in the j-th iteration. η is the learning rate.

According to Equations (3)–(6), the PRI feature function may be nonlinear, and it can hardly be modeled and rebuilt analytically. By now, recurrent neural networks (RNN) have provided good representation abilities for such complex functions. The RNN is a deep learning framework designed for pulse sequence processing and has achieved a lot in applications such as machine translation and financial prediction. The RNN deals with sequential pulses one by one to extract the temporal feature or predict the parameters of subsequent pulses based on that of the former one. The pulse parameter prediction prospect forms a great inspiration for learning PRI features in pulse trains. To achieve this goal, we introduce a deep recurrent framework, as sketched in Figure 2, to mine PRI feature functions from DTOA sequences.

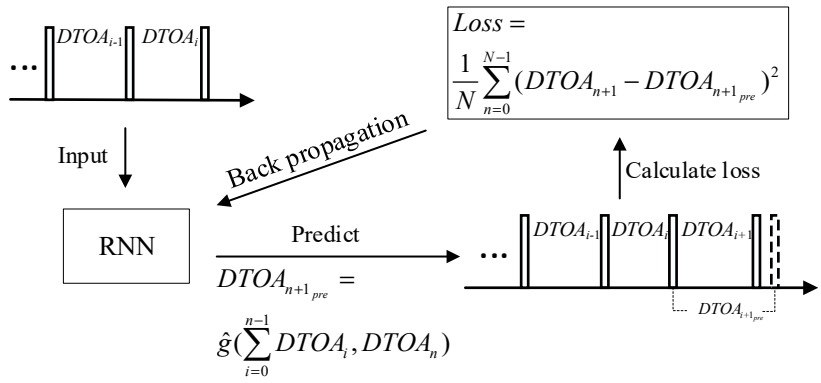


Figure 2. Flow chart of PRI feature learning using RNN.

This is a self-supervised learning process, and the RNN processes the pulse sequence sequentially. After receiving each input, there will be a single-step prediction output. The $DTOA_i$ and accumulated time of previous pulses are input into the RNN to get a predicted $DTOA_{n+1_{pre}}$ according to Equation (6). There is a deviation between the true $DTOA_{i+1}$ and the predicted one. The prediction loss is then calculated and accumulated according to Equation (7), and the BP algorithm [26] is then used to optimize the RNN parameters. The unfolded diagram of this process is shown in Figure 3.

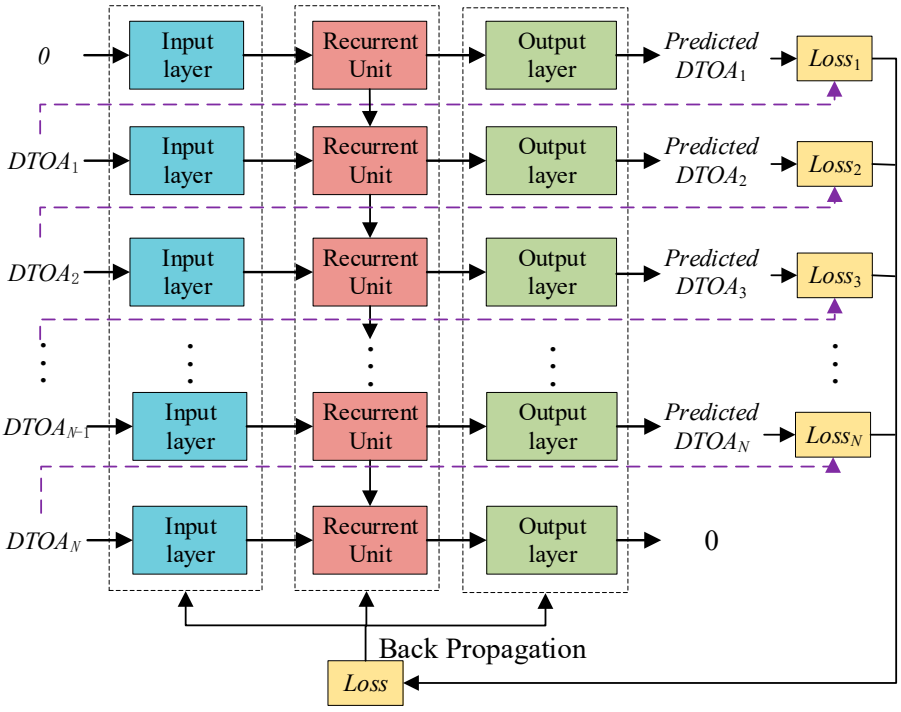


Figure 3. Unfolded diagram of self-supervised learning RNN.

In Figure 3, all layers and recurrent units could be regarded as mapping functions. The recurrent module can be realized with four types of gated units, including the simple RNN unit (SRU, no gate), minimum gated unit (MGU, a forget gate) [27], gated RNN unit (GRU, forget and input gates), and long short-term memory (LSTM, forget, input and output gates). Gated units help to mitigate the gradient disappearance and explosion caused by long-time dependence.

Through many learning iterations based on a large number of pulse trains, the learned PRI feature function is expected to approximate the true function relationship well, and the rebuilt function will be stored in the RNN in the form of a network structure and parameters.

4. Parameter Prediction Simulation

This part describes the PRI feature learning process exampled with a three-mode VPR. Then, based on the trained RNNs, a parameter prediction task is performed to verify the learning effect of PRI feature.

4.1. Data Set Simulation

To cope with different detection distances, the radar widely applies three working modes, namely low, medium, and high pulse repetition frequency (PRF) [1,28]. Thus, three types of pulse groups are included in the simulated data set. Each pulse group has an initial PRI range and a linear PRI feature function. The designed pulse group structure is shown as:

$$\begin{gathered} PRI_1 \in [220, 280]; PRI_2 \in [450, 500]; PRI_3 \in [700, 780] \\ \begin{bmatrix} [PRI_1, PRI_1 + 20, PRI_1 - 30, PRI_1 + 10] \\ [PRI_2, PRI_2 + 30, PRI_2 + 60, PRI_2 + 20] \\ [PRI_3, PRI_3 + 40, PRI_3 + 80, PRI_3 + 40] \end{bmatrix} us \end{gathered} \tag{9}$$

When generating pulse trains, the initial PRI of each pulse group is randomly determined in the variable interval range, and different kinds of pulse groups are switched with equal probability. The standard deviation of TOA is 2 us. There are 50–55 pulses in each pulse train, and 50000 pulse trains are generated in data set under same parameters. Pulses are lost randomly according to the missing pulse rate (MPR). The interferential pulses in the pulse train conform to Poisson distribution [29], and its parameter λ is regarded as the interferential pulse ratio (IPR).

4.2. Adaptation Verification to Pulse Trains with Variable Parameters

Prediction ability is an essential representation that a model understands the sequence pattern. To demonstrate that the RNN well adapts to VPR pulse trains, we compare the single-step prediction performances of four time series prediction models. The models used for performance comparison are ARIMA in Statsmodel, SVM in Sklearn, and CNN in Pytorch, and they are realized by third-party libraries in Python. We splice the unequal-length pulse trains and get a lengthy pulse train containing 10,000 pulses. The former 9900 pulses are taken as the training data and the last 100 pulses are used to test the prediction abilities of different models. The test results are shown in Figure 4 and Table 1.

As shown in Figure 4, the four prediction models can well predict trends of the PRI sequence. There would be a sharp deviation between the predicted and true values at the conversion of pulse groups due to its randomness. Actually, the prediction ability in a pulse group is the main difference between different models. In Table 1, we measure the prediction performance through commonly used indicators such as MAE, RMSE, MPE, MAPE, and R2 (Their full names are shown in Appendix A). One can conclude easily from the results in Table 1 that the proposed RNN model realizes the prediction of VPR pulse trains best among the listed methods.

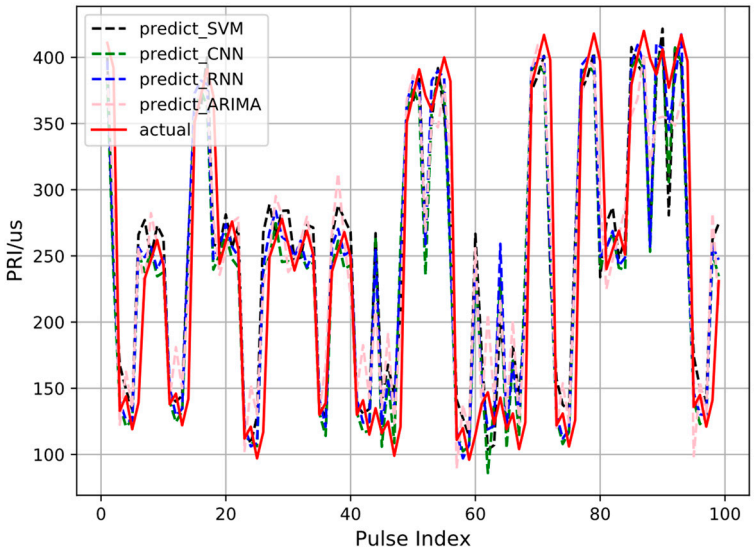


Figure 4. The prediction performances of four prediction models.

Table 1. Different prediction indicators of four prediction models.

Model	MAE↓	RMSE↓	MPE↓	MAPE↓	R^2↑
ARIMA	70.232	102.359	−6.089	35.461	−0.292
SVM	38.512	58.354	4.54	16.834	0.604
CNN	35.211	58.057	−2.049	15.893	0.645
RNN	**34.195**	**55.55**	**1.611**	**14.983**	**0.683**

↓means that smaller is better for this indicator; otherwise, it is ↑. Bold values represented the best.

4.3. Neural Network Training

The data set is trained based on the framework in Figures 2 and 3. To facilitate the loss calculation, each input DTOA is represented as a one-hot vector with 2001 dimensions [30], and the maximum DTOA that can be expressed is 2000us, and the DTOAs that exceed it are set to 0. An embedding layer compresses the inputted one-hot vector to 128 dimensions. The recurrent unit acts as a hidden layer to extract information, and its size is 128. Pytorch provides us with standard recurrent units. The output is the full connection layer with the same size as the input layer. Finally, through the activation function of SoftMax, a predicted probability spectrum of the next PRI is output. Batch training is adopted with 64 pulse trains per batch and 5000 batches are trained in all. The learning rate is set to 0.001. In different cases, the prediction loss of iterative training is shown in Figure 5.

We can see that the less data noise, the smaller the prediction loss. For different recurrent units, the models with gated units have lower prediction loss than that without gated units (SRU), and the more gated units, the better. However, the models with fewer gated units (SRU, MGU) have faster training convergence.

4.4. Parameter Prediction Task

For exhibiting the learning effect of PRI feature functions, we design a task of predicting pulse group parameters. That is, after obtaining the first DTOA (composed of two pulses) of a pulse group, we input it into the learned model to obtain a predicted DTOA, and continuously predict the remaining PRI parameters with the predicted value as the true value. An example of this process is shown in Figure 6.

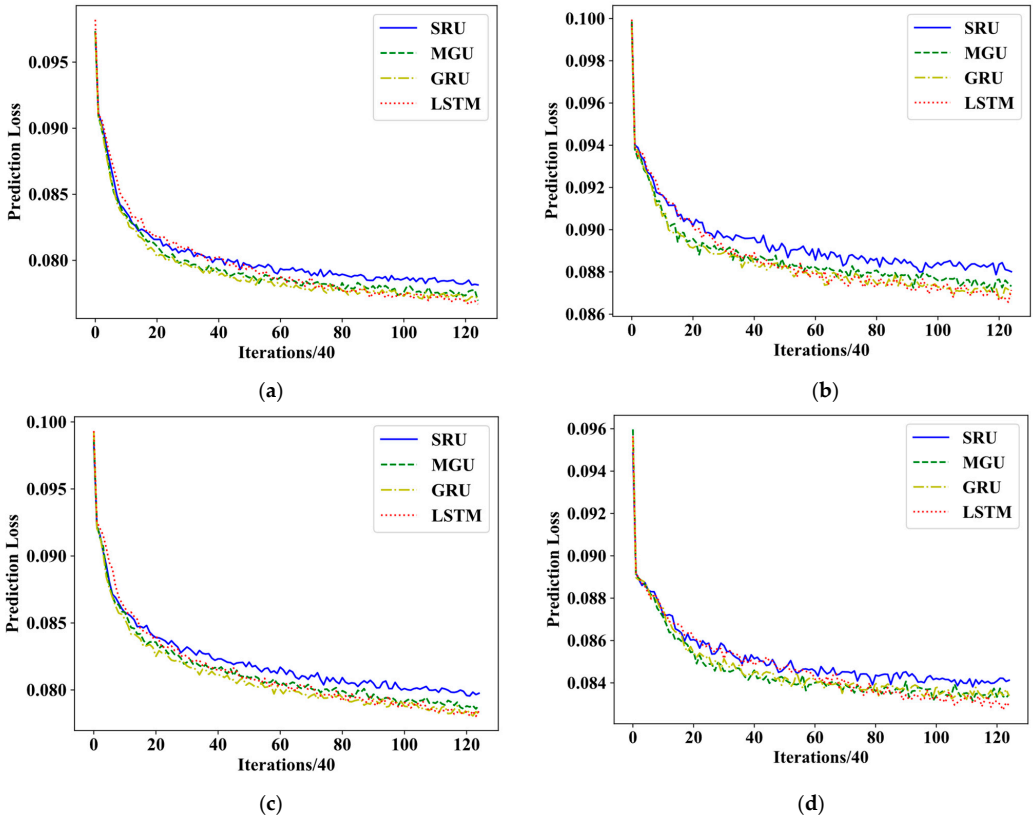


Figure 5. Prediction loss under different data noises. (**a**) MPR = 0.4, IPR = 0.2; (**b**) MPR = 0.4, IPR = 0.6; (**c**) MPR = 0.2, IPR = 0.4; (**d**) MPR = 0.6, IPR = 0.4.

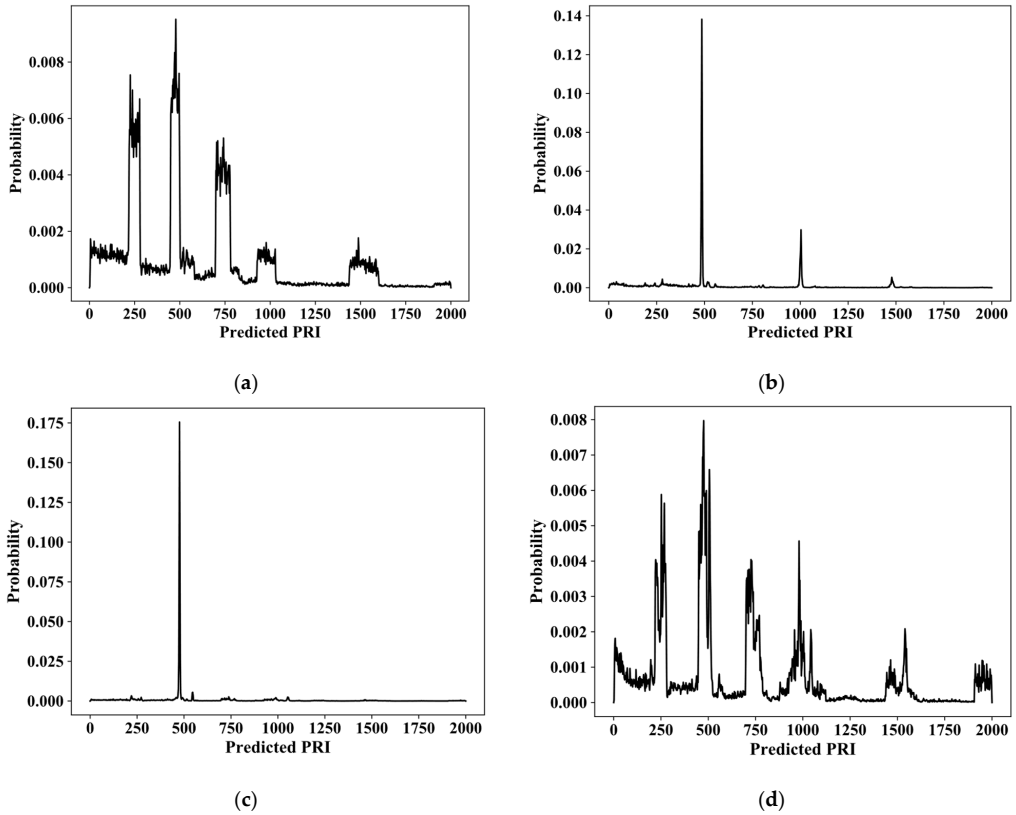


Figure 6. Prediction spectrum of LSTM under different inputs when MPR is 0.4 and IPR is 0.4. (**a**) the first input 0; (**b**) the second input 456 us; (**c**) the forth input 516 us; (**d**) the fifth input 476 us.

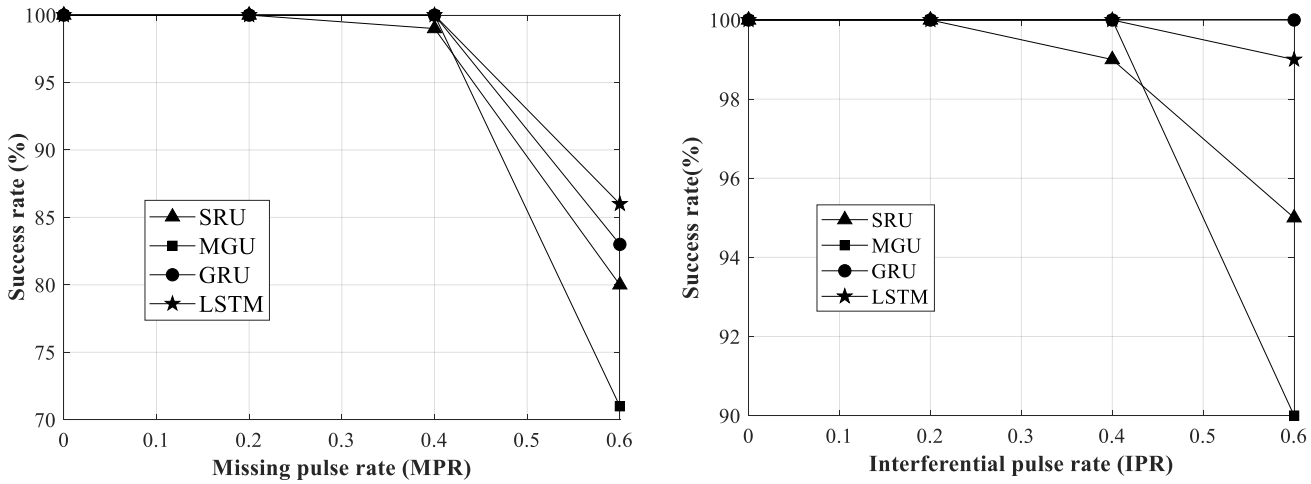
At first, only one pulse does not form a DTOA, but the model already has an input. According to Equation (3), we input the defined $d_0 = 0$ into the trained RNN, and the output indicates the possible value of the first DTOA, as shown in Figure 6a. The first three ranges show initial PRI ranges of three set pulse groups, and remaining ranges with low probability are high-order DTOAs caused by missing pulses. This result conforms to Equation (5). Then, when the second pulse is intercepted, a first DTOA is obtained as 456 us, and its further prediction spectrum is shown in Figure 6b. Its predicted PRI is 487 us with the maximum possibility. After that, we input the predicted value into the RNN model to continue iterative predictions. Until we input the PRI (476 us) with the maximum probability in Figure 6c, the predicted PRI peak in Figure 6d is significantly lower than that in the last prediction, and there are multi-range peaks. This scene is caused by the uncertainty of the next DTOA during pulse group conversion, which means this pulse group has ended and we have finished the pulse group prediction.

In this way, once the second pulse or first DTOA is intercepted, we can complete the prediction of the whole pulse group. We call it a successful pulse group prediction if the PRI feature of predicted pulse group belongs to the patterns in Equation (9). Different first PRIs are intercepted to predict their pulse groups, and we will count the success rate to measure the performance of the PRI feature learning.

This task is of great significance to improve the real-time requirements of the electronic support system due to its sequential and online prediction.

4.5. Influence of Missing Pulse Rate (MPR) and Interferential Pulse Ratio (IPR)

The performance of PRI feature learning is tested under different MPRs and IPRs. The above pulse group prediction task is executed 100 times to obtain success rates as shown in Figure 7.



(**a**) (**b**)

Figure 7. Influence of data noise on prediction task. (**a**) influence of MPR when IPR is 0.4; (**b**) influence of IPR when MPR is 0.4.

As can be seen from Figure 7a, when the IPR is fixed to 0.4, the prediction success rates of RNNs decrease with the increase in the MPR, which reflects that missing pulses increase the burden on RNNs to learn PRI features. When the IPR is 0.4 and the MPR does not exceed 0.4, RNNs can learn the PRI feature in pulse trains well, and their success rates of pulse group prediction can be close to 100%. While the MPR reaches 0.6, SRU, GRU and LSTM still achieve a success rate of more than 80%, but MGU has a significant decline of 29% in the pulse group prediction.

Figure 7b shows that interferential pulses have a similar effect on the PRI feature learning and predicting pulse groups with missing pulses. However, the influence of

interferential pulses is smaller with success rates of four RNNs reaching more than 90% even when the IPR reaches 0.6. In general, LSTM and GRU can better learn PRI features in pulse trains and have a great stability against data noises. However, MGU with only a forgetting gate is not suitable for this prediction task with noises [27].

4.6. Influence of Parameter Variable Range

The PRI agility range is a key factor of the VPR. To explore its influence on PRI feature learning, we change the initial PRI range of each pulse group in Equation (9). Then, MPR and IPR are set as 0.4, and data sets are established for training four RNNs. The pulse group prediction task is executed 100 times to obtain the prediction success rates of different agility ranges, as shown in Table 2.

Table 2. Influence of PRI agility range on PRI feature learning.

Agility Range of Initial PRIs/us	Agility Ratio	Success Rate of Pulse Group Prediction			
		SRU	MGU	GRU	LSTM
$PRI_1 \in [285,315]$; $PRI_2 \in [475,525]$; $PRI_3 \in [760,840]$	10%	100%	99%	100%	100%
$PRI_1 \in [270,330]$; $PRI_2 \in [450,550]$; $PRI_3 \in [720,880]$	20%	87%	76%	95%	97%
$PRI_1 \in [255,345]$; $PRI_2 \in [425,575]$; $PRI_3 \in [680,920]$	30%	51%	43%	60%	69%

In Table 2, when the agility ratio of initial PRIs is 10%, PRI features can be well learned by RNNs with prediction success rates of 100%. Although the agility ratio is 20%, the prediction success rates of GRU and LSTM are still over 95%, but the learning ability of RNN and MGU decreases significantly. Once the agility range reaches 30%, all four RNNs lose their ability to stably predict pulse group parameters, and LSTM has the highest success rate of 69%. Thus, RNNs can adapt to variable PRIs of the VPR to a certain context, its robustness needs to be further improved if the VPR has a much larger agility range.

5. Conclusions

In this communication, the PRI features hidden in variable parameter radar pulse trains are explored as feature functions, and a self-supervised learning framework is built to learn them using four RNNs with different recurrent units. By comparing different prediction models, we verify that RNN is capable of processing VPR pulse sequences. Then, we set a pulse group prediction task to verify the effect of PRI feature learning with RNNs. It shows that LSTM is more suitable for learning PRI features from pulse data of the variable parameter radar than other recurrent units, especially in the case of more complex data noises and larger parameter variable ranges. It is able to predict pulse parameters with a success rate of 96% when the IPR is 0.4 and MPR is 0.6, and it also can adapt to a PRI agility ratio of 20% with a success rate of 97%.

Author Contributions: S.Y.: data curation, methodology, writing—original draft.; Z.-M.L.: conceptualization, supervision, review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by Provincial Outstanding Youth Fund of Hunan, Grant/Award Number: 2020JJ2037; Hunan Youth Talent Program, Grant/Award Number: 2019RS2026.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. The description table for abbreviations.

Abbreviations	Full Names	Descriptions
DTOA	Differential time of arrival	It is the interval between two pulses.
ESM	Electronic support measurement	It supplies the timely intelligence and threat recognition to allow effective attack and protection.
ECM	Electronic Counter Measure	It is an electrical or electronic device designed to trick or deceive radar, sonar or other detection systems.
GRU	Gated RNN unit	An RNN unit with forget and input gates.
IPR	Interferential pulse ratio	The occurrence of inferential pulses follows Poisson distribution and the parameter λ is regarded as the interferential pulse ratio.
LSTM	Long short-term memory	An RNN unit with forget, input and output gates.
MAE	Mean Absolute Error	
MAPE	Mean absolute percentage error	
MGU	Minimum gated unit	An RNN unit with a forget gate.
MPE	Mean Percentage Error	
MPR	Missing pulse rate	The probability that a pulse is lost in the pulse train.
PRF	Pulse repetition frequency	The number of pulses transmitted per second.
PRI	Pulse repetition interval	
RF	Radio frequency	
RMSE	Root Mean Square Error	
RNN	Recurrent neural network	It is a deep model framework for dealing with sequence data.
R2	R-square	
SRU	Simple RNN unit	An RNN unit without gate.
TOA	Time of arrival	It is the time when the receiver intercepts the pulse.
VPR	Variable parameter radar	A radar with variable inter-pulse time intervals.

References

1. Skolnik, M.I. Electronic counter-countermeasures. In *Radar Handbook*, 3rd ed.; McGraw-Hill: New York, NY, USA, 2008; pp. 24.31–24.32.
2. Galati, G.; Pavan, G.; Savci, K.; Wasserzier, C. Counter-Interception and Counter-Exploitation Features of Noise Radar Technology. *Remote Sens.* **2021**, *13*, 4509. [CrossRef]
3. Liu, Z.-M. Online Pulse Deinterleaving with Finite Automata. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 1139–1147. [CrossRef]
4. Ou, J.; Chen, Y.; Zhao, F.; Liu, J.; Xiao, S. Method for operating mode identification of multi-function radars based on predictive state representations. *IET Radar Sonar Navig.* **2017**, *11*, 426–433. [CrossRef]
5. Mardia, H. New techniques for the deinterleaving of repetitive sequences. *IEE Proc. F Radar Signal Process.* **1989**, *136*, 149–154. [CrossRef]
6. Milojevic, D.; Popovic, B. Improved algorithm for the deinterleaving of radar pulses. *IEE Proc. F Radar Signal Process.* **1992**, *139*, 98–104. [CrossRef]

7. Nishiguchi, K.; Kobayashi, M. Improved algorithm for estimating pulse repetition intervals. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 409–421. [CrossRef]
8. Tao, J.-W.; Yang, C.Z.; Xu, C.W. Estimation of PRI Stagger in Case of Missing Observations. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7982–8001. [CrossRef]
9. Cheng, W.; Zhang, Q.; Dong, J.; Wang, C.; Liu, X.; Fang, G. An Enhanced Algorithm for Deinterleaving Mixed Radar Signals. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 3927–3940. [CrossRef]
10. Zhangmeng, L.; Shuo, Y.; Shiqian, K. Semantic coding and model reconstruction of multifunctional radar pulse train. *J. Radars* **2021**, *10*, 559–570.
11. Liu, Z.-M.; Kang, S.Q.; Chai, X.M. Automatic pulse repetition pattern reconstruction of conventional radars. *IET Radar Sonar Navig.* **2021**, *15*, 500–509. [CrossRef]
12. Yuan, S.; Kang, S.-Q.; Shang, W.-X.; Liu, Z.-M. Reconstruction of Radar Pulse Repetition Pattern via Semantic Coding of Intercepted Pulse Trains. *IEEE Trans. Aerosp. Electron. Syst.* **2022**. early access. [CrossRef]
13. Liu, Z.-M. Recognition of Multifunction Radars Via Hierarchically Mining and Exploiting Pulse Group Patterns. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 4659–4672. [CrossRef]
14. Liu, Z.; Yu, P.S. Classification, denoising, and deinterleaving of pulse streams with recurrent neural networks. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 1624–1639. [CrossRef]
15. Li, Y.; Zhu, M.; Ma, Y.; Yang, J. Work modes recognition and boundary identification of MFR pulse sequences with a hierarchical seq2seq LSTM. *IET Radar Sonar Navig.* **2020**, *14*, 1343–1353. [CrossRef]
16. Matuszewski, J.; Pietrow, D. Specific Radar Recognition Based on Characteristics of Emitted Radio Waveforms Using Convolutional Neural Networks. *Sensors* **2021**, *21*, 8237. [CrossRef] [PubMed]
17. Zhu, M.; Li, Y.; Pan, Z.; Yang, J. Automatic modulation recognition of compound signals using a deep multi-label classifier: A case study with radar jamming signals. *Signal Process.* **2020**, *169*, 107393. [CrossRef]
18. Li, X.; Liu, Z.; Huang, Z. Deinterleaving of pulse streams with denoising autoencoders. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 4767–4778. [CrossRef]
19. Li, X.; Liu, Z.-M.; Huang, Z. Denoising of Radar Pulse Streams with Autoencoders. *IEEE Commun. Lett.* **2020**, *24*, 797–801. [CrossRef]
20. Zhu, M.; Wang, S.; Li, Y. Model-Based Representation and Deinterleaving of Mixed Radar Pulse Sequences with Neural Machine Translation Network. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 1733–1752. [CrossRef]
21. Richards, M.A.; Scheer, J.A.; Holm, W.A. *Principles of Modern Radar: Basic Principles*; Scitech: Rijeka, Croatia, 2013.
22. Melvin, W.L.; Scheer, J.A. *Principles of Modern Radar: Advanced Techniques*; Scitech: Rijeka, Croatia, 2013.
23. Wiley, R.G. *ELINT: The Interception and Analysis of Radar Signals*; Artech House: Norwood, MA, USA, 2006.
24. Davies, P.G.; Hughes, E.J. Medium PRF set selection using evolutionary algorithms. *IEEE Trans. Aerosp. Electron. Syst.* **2002**, *38*, 933–939. [CrossRef]
25. Mir, H.S.; Guitouni, A. Variable Dwell Time Task Scheduling for Multifunction Radar. *IEEE Trans. Autom. Sci. Eng.* **2014**, *11*, 463–472. [CrossRef]
26. Rumelhart, D.; Hinton, G.; Williams, R. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
27. Zhou, G.B.; Wu, J.; Zhang, C.L.; Zhou, Z.H. Minimal gated unit for recurrent neural networks. *Int. J. Autom. Comput.* **2016**, *13*, 226–234. [CrossRef]
28. Li, W.; Yi, W.; Wen, M.; Orlando, D. Multi-PRF and Multi-Frame Track-before-Detect Algorithm in Multiple PRF Radar System. *Signal Process.* **2020**, *174*, 107648. [CrossRef]
29. Zhao, J.; Zhang, F.; Zhao, C.; Wu, G.; Wang, H.; Cao, X. The Properties and Application of Poisson Distribution. *J. Phys. Conf. Ser.* **2020**, *1550*, 032109. [CrossRef]
30. Alaya, M.Z.; Busy, S.; Gaïffas, S.; Guilloux, A. Binarsity: A penalization for one-hot encoded features. *J. Mach. Learn. Res.* **2017**, *20*, 1–34.