

Article

Signal Separation Method for Radiation Sources Based on a Parallel Denoising Autoencoder

Xusheng Tang^{1,2,*} and Mingfeng Wei¹¹ School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China; 220225887@seu.edu.cn² Purple Mountain Laboratories, Nanjing 211111, China

* Correspondence: tangxsh@seu.edu.cn

Abstract: Radiation source signal sorting in complex environments is currently a hot issue in the field of electronic countermeasures. The pulse repetition interval (PRI) can provide stable and obvious parametric features in radiation source identification, which is an important parameter relying on the signal sorting problem. To solve the problem linked to the difficulties in sorting the PRI in complex environments using the traditional method, a signal sorting method based on a parallel denoising autoencoder is proposed. This method implements the binarized preprocessing of known time-of-arrival (TOA) sequences and then constructs multiple parallel denoising autoencoder models using fully connected layers to achieve the simultaneous sorting of multiple signal types in the overlapping signals. The experimental results show that this method maintains high precision in scenarios prone to large error and can efficiently filter out noise and highlight the original features of the signal. In addition, the present model maintains its performance and some robustness in the sorting of different signal types. Compared with the traditional algorithm, this method improves the precision of sorting. The algorithm presented in this study still maintains above 90% precision when the pulse loss rate reaches 50%.

Keywords: autoencoder; noise reduction; PRI; signal binning



Citation: Tang, X.; Wei, M. Signal Separation Method for Radiation Sources Based on a Parallel Denoising Autoencoder. *Electronics* **2024**, *13*, 1029. <https://doi.org/10.3390/electronics13061029>

Academic Editor: Paul Leroux

Received: 14 January 2024

Revised: 8 February 2024

Accepted: 10 February 2024

Published: 9 March 2024



Copyright: © 2024 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

With the extensive use of radar signals in recent years, various types of interference and signal parameter spatial overlap phenomena have increased, creating a complex electromagnetic environment around the radar signal [1]. Signals in a complex environment are specifically characterized by certain measurement errors, more lost pulses, and false pulses, and these characteristics greatly interfere with the signals [2]. Therefore, radar radiation source sorting in complex environments has become a hot issue in the field of electronic countermeasures and other fields [3].

The traditional five-dimensional parameter method utilizes the pulse description word (PDW) of the signal, i.e., the five parameters of carrier frequency (RF), pulse amplitude (PA), pulse width (PW), direction-of-arrival (DOA), and time-of-arrival (TOA), for signal sorting and identification [4], and the errors generated in a complex environment cause the five-dimensional parameter method to be prone to parameter ambiguity and signal misjudgment, which greatly affects the sorting of the signal.

Among the five parameters of the pulse descriptor, the most stable is the TOA, and its first-level difference is called the pulse repetition interval (PRI), also known as re-frequency. The PRI parameter has a more obvious pattern [5]; so, it is a practical method with which to analyze the TOA parameter. Traditional signal sorting methods based on PRI include the difference histogram algorithm (CDIF) [6], the sequence difference histogram algorithm (SDIF) [7], the PRI transformation method, and its improved algorithm [8,9], etc. A common problem with these methods is that the accuracy of the algorithms is greatly reduced in the face of complex PRI pattern signals, as well as in scenarios with high false pulse rate or lost pulse rate, which cannot be sorted correctly.

For noise interference caused by a complex environment, the current main denoising processing methods include wavelet transform [10], singular value decomposition [11], and so on. However, due to the impulse representation of radar signals in the form of pulse description words, such traditional denoising methods cannot be used.

Aiming at the problem of there being more false pulses, lost pulses, and measurement errors that can interfere with the sorting task in real scenarios, as well as more signal overlapping, this paper proposes a parallel denoising autoencoder-based target sorting method, which can be used for multi-target parallel sorting tasks in complex environments.

The algorithm studied in this paper solves the problem presented by the currently used algorithms, which are generally sensitive to noise in the task of sorting key targets of radiation sources with known parameters; in fact, it can accurately and rapidly distinguish targets and has important applications in the fields of security, aviation, and so on. In the context of electronic countermeasures, the proposed algorithm can improve target identification accuracy, identify key signals to optimize electronic jamming strategies, and sort and summarize signal data to enhance electromagnetic situational awareness. Through these features, the proposed algorithm can improve the defense capability of electronic countermeasure systems.

The main contributions of this paper are as follows: (i) the incorporation of preprocessing operations in the form of binary encoding. The intercepted raw TOA data are preprocessed into binary sequences and then fed into the parallel denoising autoencoder network model for training. This operation facilitates the extraction and recognition of the data by the network model, highlights data features, and further improves the model recognition efficiency; (ii) the construction of a parallel denoising autoencoder network model. The described model contains multiple denoising autoencoders that can work in parallel, adapt to the strong noise environment, and realize the effective recognition and sorting of multi-target overlapping signals. The method described in this paper can realize the synchronous sorting of overlapping signals under the joint interference of lost pulse, false pulse, and measurement error, and the precision of sorting is improved compared with that of traditional algorithms.

2. Data Preprocessing

TOA (time-of-arrival) refers to the arrival time of a pulse, usually defined as the time at which the pulse front is received. In a single radiation source environment, the first-order difference of the TOA parameter is defined as the pulse repetitive interval (PRI), i.e., the re-frequency, which is the interval between two neighboring pulses [5]. According to the regular differences exhibited by the re-frequency characteristics, the inter-pulse modulation of radar signals can be classified into various types [12]. The most typical types include fixed re-frequency, periodic modulation re-frequency, and so on [13].

TOA data are a one-dimensional time series that can grow continuously, and the TOA sequence of a radar signal is usually expressed as

$$T = \{t_1, t_2, \dots, t_n\}, \quad (1)$$

where n is the total number of data contained in the sequence.

When dealing with large-scale samples, there is a situation in which the value domain range of TOA sequences is too large, and the PRI range is relatively small, resulting in PRI data being difficult to capture and learn. The fitness of such data for neural networks is low. To address this problem, this work involved carrying out preprocessing operations on the acquired TOA sequences and input them into the network after transforming them into a data form suitable for machine model learning. The preprocessed data had a clear regularity, a simple format, and high adaptability to the neural network, allowing the effective recognition of the modulation type.

The preprocessing method in this paper is binary coding. It encodes the TOA sequence data into binary sequence data consisting of zeros and ones.

Typically, the arrival time of the first pulse of the signal is designated as zero. According to Equation (1), the timeframe of the entire TOA sequence can be determined as $[0, t_n]$. Introducing the variable t_{delta} as the unit time, this timeframe is segmented into equal intervals based on the unit time duration, with the number of segments represented by

$$N = \frac{t_n}{t_{\text{delta}}}. \quad (2)$$

The mathematical model of binary coding adopted in this paper is:

$$c_j = \begin{cases} 1, & (j-1) \cdot t_{\text{delta}} < t_i \leq j \cdot t_{\text{delta}} \\ 0, & \text{otherwise} \end{cases}, j = 1, 2, \dots, N; i = 1, 2, \dots, n. \quad (3)$$

As shown in Equation (2), time segment j is denoted as $[(j-1) \cdot t_{\text{delta}}, j \cdot t_{\text{delta}}]$. If the acquired TOA data point t_i falls within time segment j , it is encoded as one, otherwise, it is encoded as zero. The resultant encoded TOA sequence is transformed into a fresh binary sequence denoted by

$$C = \{c_1, c_2, \dots, c_N\}. \quad (4)$$

With an array of re-frequency values within a cycle, each type of re-frequency value cycles through according to the array. Assuming $t_{\text{delta}} = 20 \mu\text{s}$, an example is given to illustrate the coding logic of the TOA sequence and represent the coding sequences for the four modulation types.

As shown in Table 1, complex types of re-frequency features such as slip change are difficult to quickly and clearly extract from TOA sequences. However, upon conversion to binary sequences, the PRI laws become more apparent, as machines are better equipped to handle binary sequences than decimal numbers. Furthermore, employing the unit time for encoding initially helps mitigate certain measurement errors. This preprocessing step effectively accentuates the modulation law, enhances computational efficiency, and diminishes measurement errors in the TOA data, thereby facilitating subsequent training and sorting tasks.

Table 1. Representation of coding sequences for the four modulation types.

Modulation Types	Repetition Value (μs)	TOA Fragment (μs)	Coding Sequence
Fixed	[60]	[60, 120, 180, 240]	[0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1]
Sliding	[40, 60, 80]	[40, 100, 180, 220]	[0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1]
Group	[50, 50, 70, 70]	[50, 100, 170, 240]	[0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1]
Cyclic	[27, 84, 69]	[27, 111, 180, 207]	[0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1]

3. Methods

3.1. Model Introduction

Autoencoder (AE) [14] is an unsupervised learning neural network model that automatically learns the intrinsic laws within data without requiring pre-labeling. It is defined by its ability to reconstruct the input data as faithfully as possible in the output. The autoencoder consists of two components—encoder and decoder—each for two distinct phases—encoding and decoding. During encoding, the input data are compressed, while decoding entails the precise reconstruction of the data. Currently, the autoencoders primarily serve the purposes of data compression and feature extraction. By retaining the essential features and reducing data dimensionality, they effectively accentuate data patterns, thereby enhancing operational efficiency. Consequently, autoencoders are widely applied to anomaly detection and denoising tasks.

The denoising autoencoder (DAE) [15] represents a special category within the autoencoder family. While the fundamental structure of the denoising autoencoder remains largely unchanged compared to the basic autoencoder, it introduces artificial contamination to the input data. The deliberate addition of noise disrupts some regularities within the

original data, compelling the model to acquire more robust denoising features. As a result, denoising autoencoders excel in preserving data information while effectively mitigating the impact of noise interference.

Due to the output characteristics of autoencoders, a single autoencoder model can only accommodate one type of signal. In practical scenarios, intercepted signals often comprise multiple distinct signals requiring classification. Repeatedly employing individual autoencoder networks or sequentially calling multiple autoencoder networks for training and sorting proves to be not only inefficient but also vulnerable to errors, and lacks robustness. To address these challenges, this paper introduces a parallel denoising autoencoder model aimed at enabling the simultaneous training and sorting of multiple modulation types of signals, leading to the concurrent acquisition of diverse binning results.

The workflow of the parallel denoising autoencoder is illustrated in Figure 1. Initially, the various TOA sequence data intended for training undergo preprocessing to encode them into corresponding binary sequences. During the pre-training phase, noise is added to the training binary sequences, followed by synchronous operation of denoising autoencoders to generate N network model files tailored to distinct modulation types. Subsequently, in the binning phase, the TOA sequences containing assorted radar signals are preprocessed into binary coded sequences. These binary sequences are then simultaneously inputted into the parallel denoising autoencoder network, utilizing the trained model files to yield multiple corresponding outputs. As noise presence may corrupt the PRI law of the signal and lead to errors in the output sequence during signal recovery, resulting in misclassification of radar signals, the model confirms the sorted types. Any instances of identical output types prompt further meticulous comparisons for precise identification. Ultimately, the network yields all the finalized binning results.

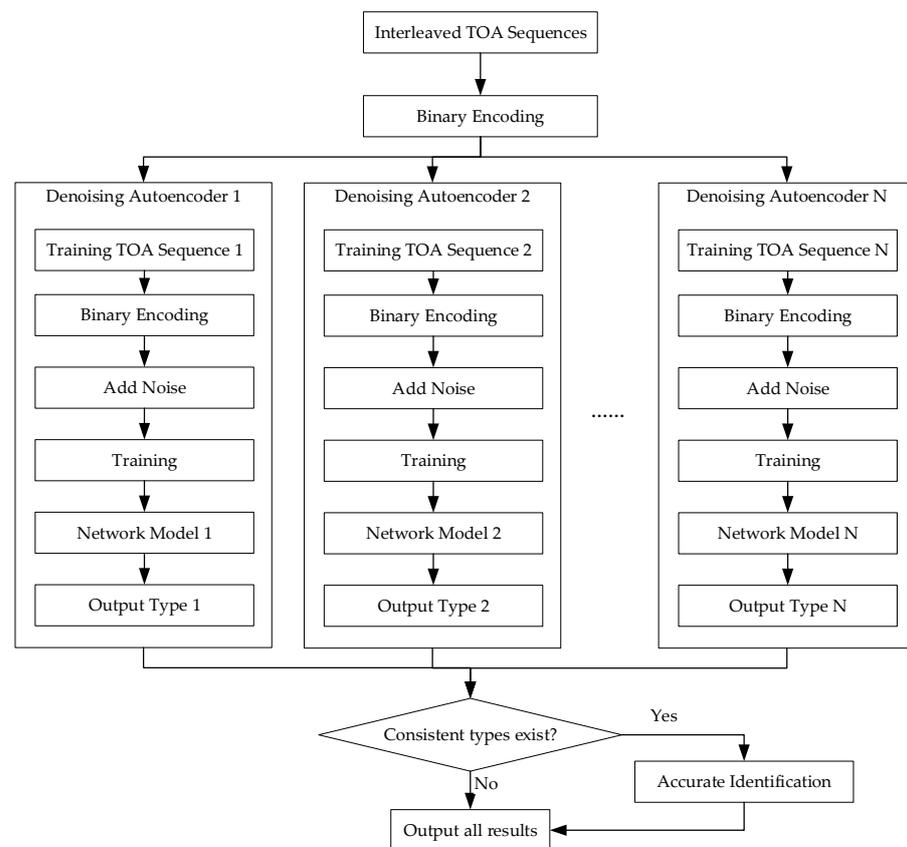


Figure 1. The workflow diagram of the parallel denoising autoencoder.

When conducting the sorting task, the parallel denoising autoencoder model synchronously activates multiple autoencoders to process the input aliased signals, generating multiple outputs simultaneously. This process incorporates specific error-correction mechanisms, thereby enhancing the efficiency and accuracy of signal sorting.

3.2. Architecture Design of the Model

The overall structure of the parallel denoising autoencoder model outlined in this paper is shown in Figure 2. In the coding sequence, solid circles represent the positions of signal pulses, with different colors denoting various signal types. Hollow circles signify the zeros within the coding sequence, while other circles mark the locations of assorted noise types. The model receives the aliased signals through a designated input port, and the signal sorting task is based on several relatively independent denoising autoencoder models operating in parallel. Each autoencoder network is responsible for sorting a specific signal type and is tasked with isolating the desired focal signal type from the binary encoded noisy aliased TOA sequence while removing the contained noise.

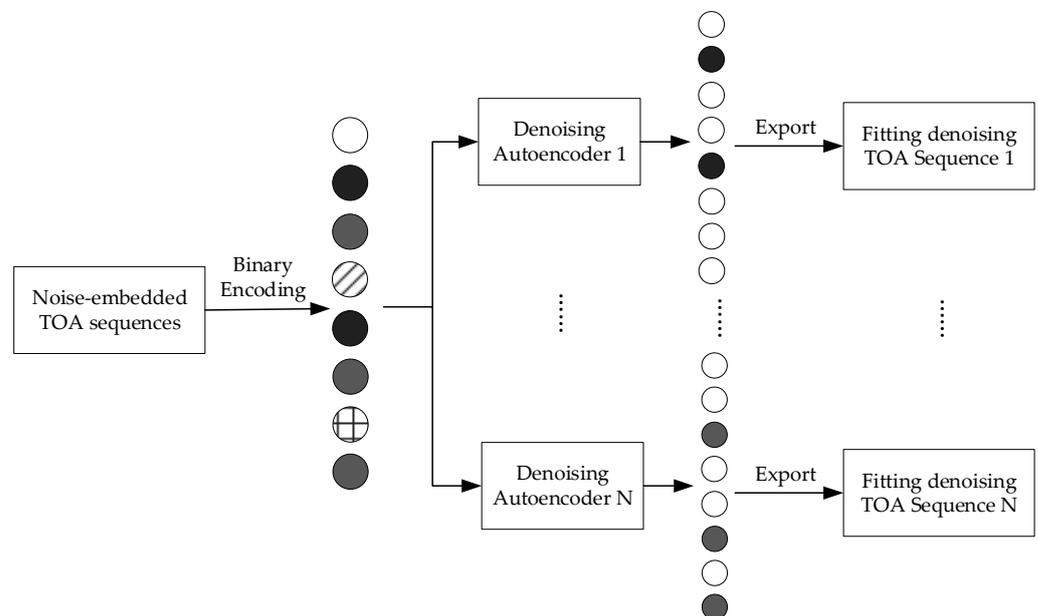


Figure 2. The model diagram of the parallel denoising autoencoder.

To ensure the generalizability of the entire model for various radar signal sorting tasks and the stability of the model itself, consistent structural design is maintained across all units. For clarity, use one unit as an example to elaborate on the network’s structural design.

The network model of the denoising autoencoder unit is shown in Figure 3. In the coding sequence, solid circles denote the positions of signal pulses, hollow circles indicate zeros, and other circles illustrate the locations of various types of noise. After preprocessing the TOA data, noise is added to the resulting binary sequence as input to the denoising autoencoder. The noise addition process is represented as

$$y = h(x), \tag{5}$$

where x represents the encoded binary sequence, and $h(\cdot)$ denotes the noise addition function. When recognizing a specific focal signal type, other types are considered as noise.

The model comprises an encoder and a decoder, both consisting of multiple fully connected layers. A single fully connected layer is mathematically represented as

$$f(x) = g(Wx + b), \tag{6}$$

where x is the input, W denotes the layer's weight, b is the offset value, and $g(\cdot)$ represents the activation function. Since neural networks themselves are linear transformations, a linear model can only solve a limited set of problems. To introduce nonlinearity and enhance the model's effectiveness, activation functions are generally used. In this paper, all fully connected layers, except for the last one, employ the rectified linear unit (ReLU) activation function defined as:

$$g_{ReLU}(x) = \max\{0, x\}. \quad (7)$$

The ReLU function outputs the greater value between zero and the input. It helps prevent the gradient vanishing problem and accelerates convergence during gradient descent. The last fully connected layer uses the Sigmoid function, which maps all output data to the range of zero to one:

$$g_S(x) = \frac{1}{1 + e^{-x}}. \quad (8)$$

Given that the final model output should be a binary sequence, a threshold ξ is set after the Sigmoid function. Numbers greater than or equal to ξ are assigned one, while numbers less than ξ are assigned zero. In this model, ξ is set to 0.5.

As seen in Figure 3, the encoding phase involves the noise-added data entering the input layer of the model and being mapped to multiple fully-connected hidden layers denoted as:

$$z = f^{(L_1)}(y) = \underbrace{f(\cdots f(y))}_{L_1}, \quad (9)$$

where L_1 represents the number of layers in the fully connected layer within the encoder, and y is the output of Equation (4), i.e., the binary sequence after adding various types of noise, which is substituted as an input variable into Equation (5) and mapped. The output z of the encoder typically has a much lower dimension than the input y , which aligns with its function of reducing data dimensionality. This compressed data z retains important features, facilitating effective network learning, reducing the risk of overfitting, and minimizing error data.

The processed data z are then fed into the decoder for reconstruction of the original data, denoted as

$$x' = f'^{(L_2)}(z) = \underbrace{f'(\cdots f'(z))}_{L_2}, \quad (10)$$

where L_2 denotes the number of fully connected layer layers within the decoder. The y in Equation (8) is the same class of signals containing different errors. Ideally, the only solution obtained by fitting y can only be the original binary sequence without noise, x . Based on this, the output of the decoder in Equation (9), x' , is the sequence fitted to x , rather than reconstructing the noisy sequence, y . x' is kept in the same dimensionality as x , and is fitted as closely as possible to the eigenstructure of x .

The trained loss function measures the difference between the predicted result and the true value, and minimizing the loss function can make the output sequence of the autoencoder closer to the actual sequence. In this paper, where the input x and output x' are binary sequences, the binary cross-entropy function is chosen as the loss function, expressed mathematically, the model is presented in Equations (11) and (12):

$$\mathcal{L}(x, x') = \{l_1, \cdots, l_n\}^T, \quad (11)$$

$$l_i = -w_i [x_i \cdot \log x'_i + (1 - x_i) \cdot \log(1 - x'_i)], \quad (12)$$

where n denotes the number of training batches, x_i and x'_i represent the raw data and model output data corresponding to batch i , respectively, and w_i denotes the weights. The training begins with the initialization of weights W and the offset value b as random decimals

between zero and one. The model is then trained in multiple batches, and the network parameters are adjusted using the error back propagation (BP) algorithm to minimize loss:

$$W_{all}, b_{all} = \operatorname{argmin} \mathcal{L}(x, x'), \tag{13}$$

$$W' = W - \mu \frac{\partial l_i}{\partial W} \tag{14}$$

$$b' = b - \mu \frac{\partial l_i}{\partial b}, \tag{15}$$

where W' and b' denote the updated weights and offset values, respectively, and μ is the learning rate set for the model.

The training process continues until the model converges, reaching a state of stability. At this point, the model's parameters remain constant and can be further tested.

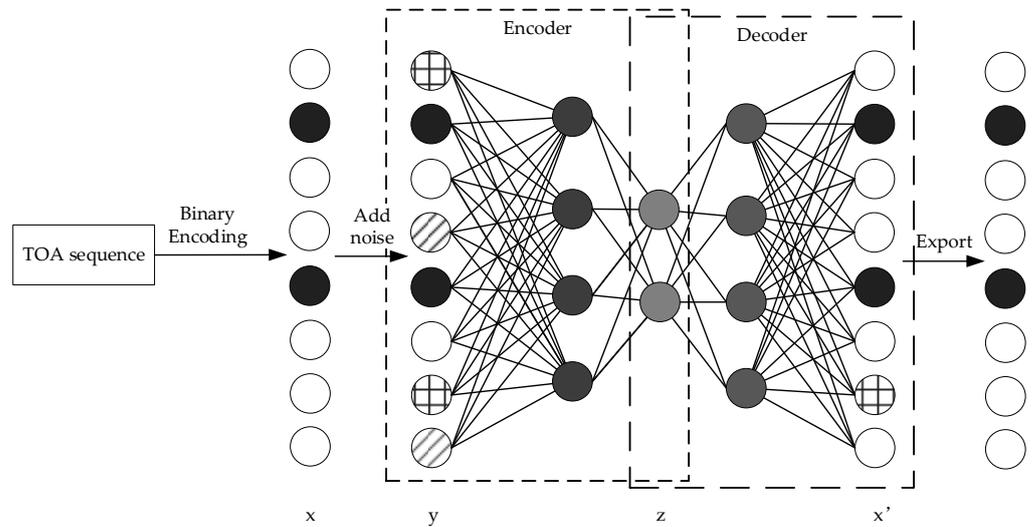


Figure 3. The model diagram of the denoising autoencoder unit.

4. Experiment

4.1. Experimental Data Simulation

4.1.1. Sample Simulation

For the sake of objectivity, this experiment selected four representative modulation types as experimental samples: fixed re-frequency, sliding re-frequency, pulse group re-frequency, and periodic modulation re-frequency. The original TOA sequences corresponding to these four types were generated using Matlab, with simulation parameters detailed in Table 2.

Table 2. The setting of sample data parameters.

Modulation Type	Cyclic Re-Frequency Value (μs)
Fixed Re-Frequency	[250]
Sliding Re-Frequency	[40, 70, 130, 160]
Pulse Group Re-Frequency	[90 × 4, 170 × 4, 240 × 4]
Periodic Modulation Re-Frequency	[$p = 50\sin(\frac{\pi}{25}t) + 100$]

Each TOA data sequence was set to a length of 10 ms, and 2000 TOA sequences were generated as training samples for each type of modulation. During the preprocessing stage, the unit time t_{delta} was defined as 10 μs for this experiment. After encoding, the length of each binary data sequence was standardized to 1000.

4.1.2. Error Simulation

In order to simulate a realistic environment, this experiment introduces various types of errors into the data. After analysis and categorization, the errors impacting the radar signal were classified into three main categories: measurement errors, lost pulses, and fake pulses.

Due to the signal or the machine itself, a small deviation in the detected actual arrival time compared to the ideal arrival time of the signal, attributed to either the signal itself or the detection system, is referred to as a measurement error. Typically, following a Gaussian distribution, measurement errors were simulated in this experiment based on an ideal PRI sequence using the equation:

$$Er_j = \frac{P_j - \mu}{\sigma}, \quad (16)$$

where P_j denotes the value at index j in the sequence, Er_j corresponds to the measurement error, μ is the mean value of the sequence, and σ denotes the standard deviation. The TOA sequence is derived from the PRI sequence and can be expressed as:

$$T' = \{t_1, t_2', \dots, t_{j+1}', \dots, t_K'\}, \quad (17)$$

$$t_{j+1}' = t_1 + \sum_1^j (P_j + Er_j), \quad (18)$$

where t_1 is the initial value of the original TOA sequence, t_{j+1}' represents the value at index $j + 1$ in the noisy TOA sequence, and K signifies the total number of data points in the sequence.

Incomplete signal fragments that are not fully received or partial pulses with similar reception times that are mistakenly interpreted as a single TOA value can result in lost pulses. Signal disturbances caused by interference and pulse aliasing from other signals contribute to the occurrence of fake pulses. By considering the specified rates of lost pulses p_l and fake pulses p_f , the number of lost pulses is determined by:

$$N_l = K \cdot p_l, \quad (19)$$

and the number of fake pulses is expressed as

$$N_f = K \cdot p_f. \quad (20)$$

The final simulated noisy TOA sequence, incorporating all three types of errors, involves updating N_l ones to zeros and N_f zeros to ones in the binary sequence derived from the preprocessing of T' .

In this experiment, the measurement error rate p_e is within the $[0, 10\%]$ of the re-frequency value, the lost pulse rate p_l is between $[0, 50\%]$, and the fake pulse rate p_f is within $[0, 70\%]$. During the training phase, these three parameters are randomly assigned values within specified ranges to generate diverse training data.

4.2. Network Parameter Setting

According to the sample setup of this experiment, a network model comprising four autoencoder units is established, with the parameter configurations of the unit model depicted in Figure 4.

The encoder within the unit model is structured with three fully connected layers containing 128, 64, and 32 neurons sequentially. Following each layer, the ReLU function is applied. Conversely, the decoder is composed of three fully connected layers with 32, 64, and 128 neurons sequentially. In this case, the ReLU function is used after the initial two layers, while the Sigmoid function is employed after the final layer.

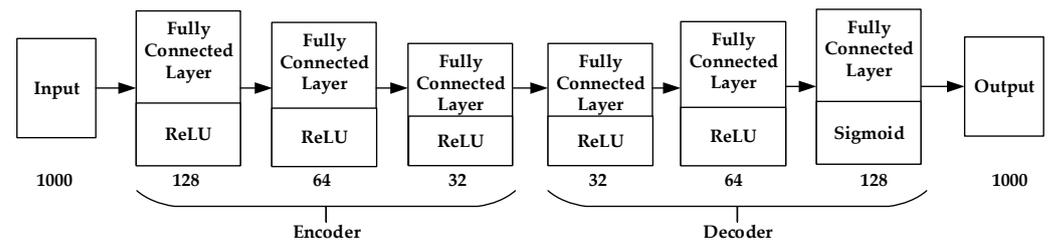


Figure 4. The parameter diagram of the autoencoder unit model.

The Pytorch framework is utilized in this experiment with a batch size of 128, a learning rate set at 0.0001, and a weight decay of 1×10^{-5} . Experimental results demonstrate that the model efficiently converges to a more optimal state within 20 cycles.

4.3. Description of The Training Process

Prior to testing, the model developed in this paper requires training. The training dataset is initially constructed by conducting preprocessing operations as detailed in Section 4.1, in order to convert the simulation data into binary form. Each of the four types in the dataset comprises 2000 binary data entries, with each piece of data having a length of 1000 and encompassing the PRI features of its corresponding type, along with randomly added errors falling within a specified range. Four denoising autoencoder units are established based on the parameters outlined in Section 4.2, and they are organized in parallel to construct the parallel denoising autoencoder model essential for this paper. During training, the model feeds the data from the training set into the denoising autoencoder units based on their respective type, with each unit processing data of only one type at a time. Throughout the training process, each unit autonomously completes the reconstruction of input data pertaining to its designated type. Employing a batch size of 128, the model achieves convergence within 20 cycles. Upon completion of training, the model file of the parallel denoising autoencoder is saved to capture the model structure after all four units have converged.

4.4. Analysis of Experimental Results

This paper comprehensively evaluates the algorithm's effectiveness using accuracy, precision, and recall. As shown in Table 3, the experimental results of the binary classification problem are classified into four types. TP represents data with positive predictions that align with the actual outcome, FP signifies data with positive predictions but contradicting actual negative outcomes, FN indicates data with negative predictions but actual positive outcomes, and TN denotes data with negative predictions matching the actual negative outcomes. Consequently, accuracy is denoted as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (21)$$

precision is denoted as

$$Precision = \frac{TP}{TP + FP} \quad (22)$$

and recall is denoted as

$$Recall = \frac{TP}{TP + FN} \quad (23)$$

Table 3. Table of binary classification results.

	Actual Positive Samples	Actual Negative Samples
Predicting positive samples	TP	FP
Predicting negative samples	FN	TN

4.4.1. Modulation Type Sorting Experiment

The test dataset still retains the modulation type and parameter configurations outlined in Table 2, with measurement errors within the [0, 10%] range and the random incorporation of fake pulses within the [0, 70%] range, added at random. The pulse loss rate was systematically increased from 0 to 50%, resulting in the generation of five distinct data sets. By overlaying the four types of TOA coding sequences at equivalent loss rates, five sets of test data comprising aliased signals with varying pulse loss rates were compiled, each set comprising 2000 binary coding sequences.

This experiment aims to assess the efficacy of the parallel denoising autoencoder model in categorizing the four types of modulated signals, with the test outcomes illustrated in Figure 5.

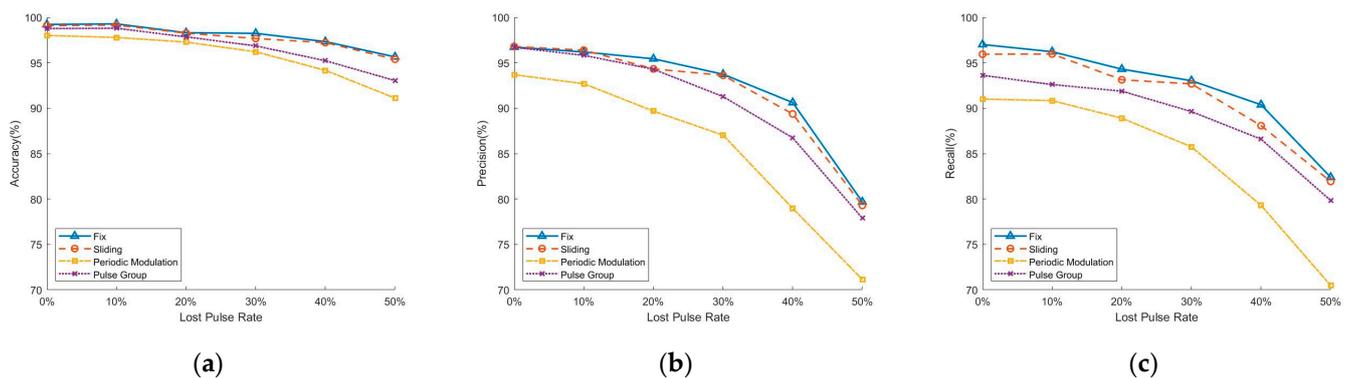


Figure 5. Sorting performance of the model for four types of signals with different pulse loss rates. Accuracy (a), precision (b), recall (c).

Figure 5 presents line plots depicting the three classification assessments at varying pulse loss rates, with (a) denoting the accuracy rate, (b) signifying the precision rate, and (c) indicating the recall rate. Observably, the model's overall performance in the classification task shows a declining trend as the lost pulse rate increases, attributed to the corruption of PRI features by lost pulses. The detrimental impact becomes more pronounced with higher lost pulse rates, and a substantial decrease in the model's classification performance is evident after the lost pulse rate exceeds 30% in Figure 5. A comparison of the plots (a), (b), and (c) reveals that the reduction in accuracy depicted in (a) is notably smaller than the trends observed in (b) and (c). This is due to the binary nature of the input network signals, which can be treated as a binary classification problem during assessment. Only the number 1 in the sequence represents the pulse, but the number 0 in the sequence also occupies a certain proportion, and the recognition results of both values are calculated into the accuracy rate, yielding a relatively high accuracy rate. Consequently, in this experiment, the accuracy rate can only partially reflect the model's performance and must be considered alongside the precision and recall rates. Notably, the model's classification performance across the four types of modulated signals exhibits a consistent trend, indicating its strong generalizability. Additionally, the representative nature of the four signal types used underscores the robustness of the parallel denoising autoencoder network model formulated in this paper, demonstrating its applicability to diverse signal classification tasks. Among these types, fixed and sliding re-frequencies consistently exhibit high values for all three criteria, while performing the poorest in the periodic modulated PRI sorting task. This is attributed to the relatively simple PRI patterns for fixed and sliding re-frequencies, which consist of repetitions of a single or a small set of digits easily captured and learned by the network. In contrast, the periodic modulation PRI type is more intricate, featuring changing PRI values over a period, posing challenges for the model to learn. Nonetheless, even in the most challenging scenario, the sorting accuracy in this experiment remains above 90%, with precision and recall rates exceeding 70%, affirming the parallel denoising autoencoder model's strong performance in signal sorting tasks.

4.4.2. Comparison Experiment

The traditional method of signal sorting, predominantly based on PRI, continues to be widely utilized. This includes enhancements to classical algorithms like SDIF and CDIF, as well as approaches that incorporate neural networks. Reference [16] proposes an enhanced SDIF algorithm that integrates the PRI transform method, representing an advancement over conventional techniques. Similarly, reference [17] presents a CRRNN algorithm, which leverages RNN networks and incorporates a residual network structure to achieve higher precision. To demonstrate the effectiveness of the method proposed in this paper, we conducted a comparative experiment. This experiment involved the improved SDIF algorithm, the CRRNN algorithm, and the parallel denoising autoencoder model developed in this study, focusing on the slip-variant PRI type as a case study. The data for this experiment were specifically configured to include lost pulses, but excluded fake pulses and measurement errors.

Figure 6 demonstrates the sorting performance of the three algorithms for slip variable PRI types at different pulse loss rates in terms of the precision rate. As the pulse loss rate increases, the precision of this paper's model remains consistently above 90%, showcasing its stability. In contrast, the precision of the enhanced SDIF algorithm shows a notable decline, dropping below 50% in the most severe case. This highlights the significant superiority of the model proposed in this paper over a traditional algorithm. Meanwhile, although the CRRNN algorithm demonstrates higher precision at a pulse loss rate below 20%, its performance noticeably deteriorates when the pulse loss rate exceeds 30%. This decline can be attributed to the autoencoder network's ability to reduce data dimensionality, which in turn enhances its capability for feature extraction in scenarios characterized by high error rates.

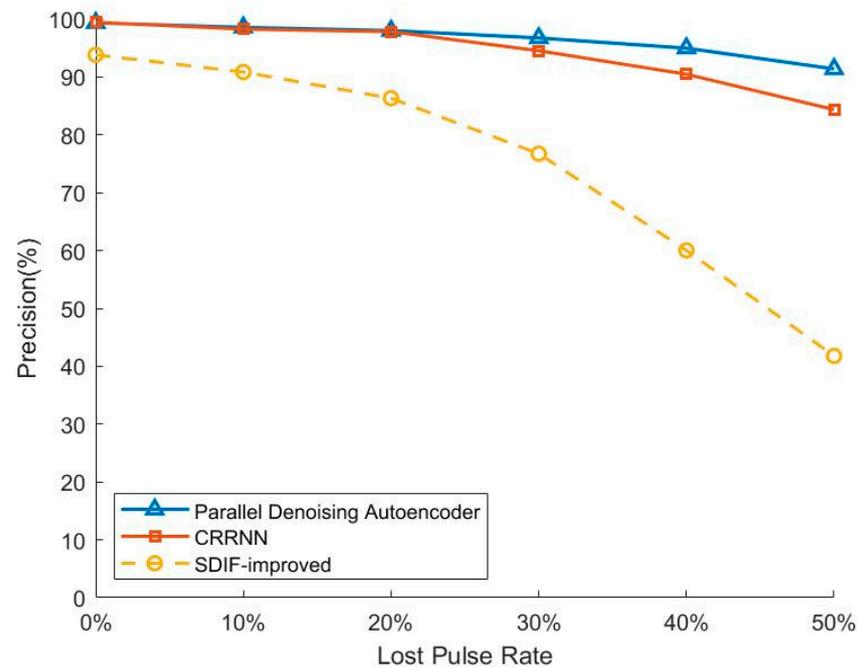


Figure 6. Comparison of algorithm performance at different pulse loss rates.

The experimental results discussed above convincingly demonstrate that the parallel denoising autoencoder network model excels in radar radiation source signal sorting, particularly when it comes to handling multiple signals in complex environments.

5. Conclusions

In this paper, we propose a parallel denoising autoencoder-based method for signal sorting. This method encodes known TOA sequences with preprocessing to amplify the PRI features of the signals. It efficiently segregates desired signal types from overlapping TOA sequences using a parallel denoising autoencoder network.

Comprehensive experimental results show that our algorithm exhibits remarkable adaptability in high error scenarios, maintaining high precision even as pulse loss rate increases. This adaptability makes it well-suited for sorting radiation source tasks in complex environments. In terms of algorithm performance, the algorithm in this paper demonstrates superior efficacy, maintaining an accuracy rate above 90% and a precision rate above 70%, even when the pulse loss rate reaches 50%. Regarding the algorithm's generalization capability, it exhibits a consistent trend and shows strong generalization capability, when facing the task of sorting various types of radiation source PRI signals.

Further analysis suggests that the advantage of the algorithm stems partly from the autoencoder's enhanced filtering ability, which effectively mitigates noise and other unfocused signals by reducing data dimensionality, thus proving to be more adaptable in complex environments. Additionally, the use of multiple repetition architectures, all built with fully connected layers, simplifies the complexity. This enables rapid sorting of extensive data and accommodates multiple sorting types.

Overall, the parallel denoising autoencoder model proficiently handles the classification of multiple signal types in challenging environments.

However, it is important to note that the algorithm performs optimally when the PRI features of the radiated source signal are distinct and the training data volume is adequate. In situations where errors, especially lost pulses, are prevalent, the autoencoder might extract incomplete PRI features, adversely affecting the sorting process. Future research will focus on enhancing the PRI features of the radiation source in extremely complex environments.

Author Contributions: Conceptualization, X.T.; methodology, X.T.; software, M.W.; validation, X.T. and M.W.; formal analysis, X.T.; investigation, M.W.; resources, M.W.; data curation, M.W.; writing—original draft preparation, M.W.; writing—review and editing, X.T.; visualization, X.T.; supervision, X.T.; project administration, X.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data underlying the results are available as part of the article and no additional source data are required.

Acknowledgments: The authors would like to thank the editors and reviewers for their efforts to help with the publication of this work.

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

References

1. Liu, Z.-M. Pulse Deinterleaving for Multifunction Radars with Hierarchical Deep Neural Networks. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 3585–3599. [[CrossRef](#)]
2. Li, X.; Liu, Z.; Huang, Z. Deinterleaving of Pulse Streams with Denoising Autoencoders. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 4767–4778. [[CrossRef](#)]
3. Dong, X.; Liang, Y.; Wang, J. Distributed Clustering Method Based on Spatial Information. *IEEE Access* **2022**, *10*, 53143–53152. [[CrossRef](#)]
4. Lang, P.; Fu, X.; Cui, Z.; Feng, C.; Chang, J. Subspace Decomposition Based Adaptive Density Peak Clustering for Radar Signals Sorting. *IEEE Signal Process. Lett.* **2022**, *29*, 424–428. [[CrossRef](#)]
5. Xiang, H.; Shen, F.; Zhao, J. Deep ToA Mask-Based Recursive Radar Pulse Deinterleaving. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 989–1006. [[CrossRef](#)]
6. He, Y.; Sang, X.; Li, Y.; Wang, J.; Guo, Q.; Wu, P. PRI-based radar signal binning algorithm and simulation implementation. In Proceedings of the 2023 IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 15–17 September 2023; pp. 706–710. [[CrossRef](#)]

7. Sang, X.; He, Y.; Li, Y. Radar signal binning based on improved SDIF algorithm. In Proceedings of the 2023 IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 15–17 September 2023; pp. 203–207. [[CrossRef](#)]
8. Liu, Y.; Zhang, Q. An improved method for deinterleaving radar signals and estimating PRI values. *IET Radar Sonar Navigat.* **2018**, *12*, 506–514. [[CrossRef](#)]
9. Xi, Y.; Wu, Y.; Wu, X.; Jiang, K. An improved SDIF algorithm for anti-radiation radar using dynamic sequence search. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 5596–5601.
10. Wu, Y.; Gao, G.; Cui, C. Improved Wavelet Denoising by Non-Convex Sparse Regularization Under Double Wavelet Domains. *IEEE Access* **2019**, *7*, 30659–30671. [[CrossRef](#)]
11. Li, Y.; Yue, X.; Wu, X.; Zhang, L.; Zhou, Q.; Yi, X.; Liu, N. A Higher-Order Singular Value Decomposition-Based Radio Frequency Interference Mitigation Method on High-Frequency Surface Wave Radar. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 2770–2781. [[CrossRef](#)]
12. Han, J.-W.; Park, C.H. A Unified Method for Deinterleaving and PRI Modulation Recognition of Radar Pulses Based on Deep Neural Networks. *IEEE Access* **2021**, *9*, 89360–89375. [[CrossRef](#)]
13. Nuhoglu, M.A.; Alp, Y.K.; Ulusoy, M.E.C.; Cirpan, H.A. Image Segmentation for Radar Signal Deinterleaving Using Deep Learning. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 541–554. [[CrossRef](#)]
14. Min, B.; Yoo, J.; Kim, S.; Shin, D.; Shin, D. Network Anomaly Detection Using Memory-Augmented Deep Autoencoder. *IEEE Access* **2021**, *9*, 104695–104706. [[CrossRef](#)]
15. Lee, W.-H.; Ozger, M.; Challita, U.; Sung, K.W. Noise Learning-Based Denoising Autoencoder. *IEEE Commun. Lett.* **2021**, *25*, 2983–2987. [[CrossRef](#)]
16. Liu, Y.; Chen, Y.; Sun, S. A radar signal sorting algorithm based on PRI. In Proceedings of the 2019 19th International Symposium on Communications and Information Technologies (ISCIT), Ho Chi Minh City, Vietnam, 25–27 September 2019; pp. 144–149. [[CrossRef](#)]
17. Al-Malahi, A.; Almaqtari, O.; Ayedh, W.; Tang, B. Radar Signal Sorting Using Combined Residual and Recurrent Neural Network (CRRNN). In Proceedings of the 2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 17–19 December 2021; pp. 21–27. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.