Deinterleaving of Pulse Streams With Denoising Autoencoders

XUEQIONG LI
ZHANGMENG LIU
ZHITAO HUANG
National University of Defense Technology, Changsha, China

Analyzing radar signals is an important task in operating electronic support measure systems. The received signals in the real electromagnetic environment often originate from multiple emitters and must be separated for further processing. Pulses from important target emitters with known parameters should be picked out first. To solve the problem, time-of-arrival (TOA) deinterleaving may be performed to extract signals from a certain emitter by learning the pulse repetition interval (PRI) modulation that makes up the signal. However, conventional deinterleaving methods only work with simple PRI modulations; their performance degrades in noisy environments. A novel approach based on denoising autoencoders for TOA deinterleaving was developed in this article. The inner patterns of pulse-of-interest sequences were learned by the proposed denoising autoencoders to generate output sequences from well-trained autoencoders. Simulation results show that the proposed method outperforms conventional methods, especially in environments with high lost and spurious pulse ratios.

Manuscript received January 17, 2019; revised November 8, 2019 and March 30, 2020; released for publication June 3, 2020. Date of publication June 24, 2020; date of current version December 4, 2020.

DOI. No. 10.1109/TAES.2020.3004208

Refereeing of this contribution was handled by P. E. Pace.

This work was supported by the Program for Innovative Research Groups of the Hunan Provincial Natural Science Foundation of China under Grant 2019JJ10004.

Authors' address: Xueqiong Li, Zhangmeng Liu, and Zhitao Huang are with Department of Electronical Science, National University of Defense Technology, Changsha 410003, China, E-mail: (lixueqiong13@nudt.edu.cn; liuzhangmeng@nudt.edu.cn; huangzhitao@nudt.edu.cn). *Corresponding author: Xueqiong Li)*

I. INTRODUCTION

The electronic support measure (ESM) system has played a significant role in electronic warfare (EW) since the 1990s [1]. ESM systems intercept and analyze sources of radiated electromagnetic energy for the purpose of immediately recognizing threat signals [2]. Deinterleaving is a vital part of ESM processing, which separates the interleaved radar pulses belonging to different emitters. Only after those intercepted pulse streams are sorted into several independent pulse trains they can be properly analyzed and processed for subsequent emitter classification tasks [3].

Advancements in communication [4], navigation [5], and radar [6] technology have made these systems increasingly complex. Crowded electromagnetic environments and complex radar modes make the deinterleaving task significantly more difficult. There may be millions of pulses intercepted by the ESM system per second in a crowded electromagnetic environment [1]. High pulse density results in high ratios of missing and spurious pulses, which drastically weakens the pulse stream regularity. Advanced radar operation modes also introduce complex pulse stream patterns containing inner connections that are very challenging to detect.

Pulse streams have been described with pulse description words (PDWs), which contain five basic parameters: pulse width (PW), carrier frequency (CF), pulse amplitude (PA), direction of arrival (DOA), and time of arrival (TOA) [2], [3], [7], [8]. A simple but effective way to solve the deinterleaving problem is to separate pulses by clustering the PW, CF, PA, and DOA. If those parameters are missing or cannot be categorized due to an overly complex environment, then the TOA values alone can be exploited for deinterleaving [9]. The first-order difference of the TOA sequence typically has inherent regularity that is referred to as the pulse repetition interval (PRI). TOA information is widely used to separate emitters.

The cumulative difference histogram (CDIF) [8] and sequential difference histogram (SDIF) [9] have been proposed to solve deinterleaving problems based on the periodic nature of pulses. The CDIF algorithm calculates the accumulation of histogram values for each difference level in one iteration. The SDIF algorithm calculates the histogram values for difference levels simultaneously. Both CDIF and SDIF perform well for pulses with simple PRI modulation modes and constant PRI values, but subharmonics can largely affect the results. The PRI transform [10] is not influenced by subharmonics and uses a complex valued autocorrelation-like integral to indicate the PRI value on the PRI spectrum, but comes with a large computational burden and is not suited to staggered PRI modulation modes.

Recently, researchers have improved various existing methods [1], [10]–[12] to separate pulse streams with other complex PRI modulation modes. However, the performance of these methods is susceptible to missing pulses. Machine learning techniques have also been utilized for deinterleaving. Fuzzy adaptive resonance theory (ART) [13], for example, has been used to cluster pulse repetition frequencies

(PRFs) to separate pulse streams. ART performs well for simple emitter patterns but is not capable of complex PRI modulation. Recurrent neural networks (RNNs) [14] can also be used to solve the deinterleaving problem. RNNs can be applied to several types of TOA sequences and are robust to some degree of noise, but RNNs have low processing efficiency due to their inherent network structure; high spurious pulse ratios can also greatly affect the RNN deinterleaving performance.

In this article, we used autoencoders to address the pulse stream deinterleaving problem. The autoencoder is a learning-based neural network that is trained on efficient data presentations by back-propagation. When it has appropriate dimensionality and sparsity constraints, the autoencoder can learn data projections that are more discriminative than principal component analysis or other basic techniques.

The autoencoder was first proposed by Ballard in 1987 [15] to pretrain artificial neural networks (ANNs). In recent years, autoencoders have been widely applied in many fields. Noh *et al.* [16], for example, used convolution and deconvolution layers as autoencoders for semantic segmentation. Yang *et al.* [17] developed an autoencoder architecture to generate images. Rasmus *et al.* [18] used a denoising autoencoder architecture for unsupervised and semisupervised feature-learning. Rematas *et al.* [19] built an autoencoder architecture for object reflectance map prediction. Zhao *et al.* [20] proposed a unified framework for supervised, unsupervised, and semisupervised learning by adding a reconstruction loss to the autoencoder. Newell *et al.* [21] suggested an autoencoder structure that relies on simple nearest neighbor upsampling for human pose estimation. Throughout the literature, data denoising and dimensionality reduction for data visualization are the two primary practical applications of autoencoders.

The main challenge yet-to-be-resolved in regards to deinterleaving is the efficient, accurate selection of the given target using known parameters. Here, we propose the use of autoencoders to extract inner patterns in pulse trains and to isolate target pulses from the interleaved pulse stream. Pulse streams containing pulses from one or more emitters are fed into the autoencoder neural network; the pulses-of-interest is the target of the model. The model can reveal the specific target pulses from within the mixed streams. A new representation of TOA sequences is introduced, and the detailed procedures and behaviors of the proposed autoencoders are presented to explain how autoencoders solve the problem of deinterleaving. The parameters of the autoencoders are tuned while training. The trained autoencoders generate pulses-of-interest when test streams are imported to them during the validation phase. This is an end-to-end process.

Simulation results show that the proposed method performs well on deinterleaving problems and is robust to lost pulses and spurious pulses. The entire process can also be paralleled to categorize several specific emitters for enhanced deinterleaving efficiency.

The rest of the article is organized as follows. Basic information for the input TOA sequences and noise in the EW environment is provided in Section II. The proposed deinterleaving model based on denoising autoencoders is presented in Section III. Section IV shows the training and validation of the autoencoder structure. Section V reports the simulations we conducted to test the proposed method. Finally, Section VI concludes this article.

II. PROBLEM FORMULATION

The main functions of ESM systems are receiving, measuring, and deinterleaving pulses to identify alternative threat emitters. Among these processes, pulse train deinterleaving is vitally important. Due to limited storage capacity and transmission speed, the original signals cannot be analyzed directly. Conventional methods of deinterleaving radar pulses involve measuring the basic parameters of individual pulses (e.g., PDWs) so that the pulses can be directly represented by PDWs. PDWs, as mentioned above, consist of the PW, PA, RF, DOA, and TOA. Other on-pulse parameters such as intrapulse modulation and polarization may also be available. The PDWs can be used to perform clustering or diagramming to identify the characteristics of different pulses.

A. TOA Sequences

In general, intrinsic sequential patterns are the most distinguishable features that separate pulse trains from other emitters. After a coarse separation of pulses with statistical features such as DOA and PW, PRI values can serve as a major parameter distinguishing target pulses from spurious pulses [8], [9]. The PRI cannot be directly measured by receivers, so TOA is often exploited as the measurable PDW parameter reflecting the regulation of PRI. In this article, we used the TOA as the parameter to deinterleave pulse trains.

A TOA sequence of an interleaved pulse stream can be written as follows:

$$T = \{t_1, t_2, \ldots, t_i, \ldots, t_N\} \tag{1}$$

where t_i is the arriving time of the ith pulse of the entire pulse stream and N is the number of intercepted pulses.

Two circumstances we simulated to test the proposed model are shown in Fig. 1. Fig. 1(a) describes the deinterleaving task of picking out pulses belonging to a certain emitter from an interleaved TOA sequence with a large number of spurious pulses. Blue columns represent the pulses-of-interest; the gap between two blue columns is the PRI of this certain emitter. Fig. 1(b) shows the separation of several groups of pulses from different emitters. Blue columns and yellow triangles represent pulses transmitted from two respective emitters; the gaps between the successive same type of pulses are their PRI values.

B. Different Types of PRI

The first-order difference of the TOA sequence typically reveals inherent regularity that can be denoted as the PRI sequence:

$$P = \{p_1, p_2, \ldots, p_{N-1}\}, p_i = t_{i+1} - t_i \tag{2}$$

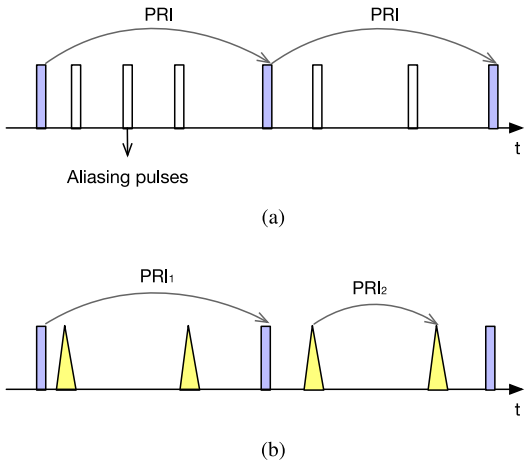


Fig. 1. Proposed model simulations. (a) Deinterleaving one emitter from other emitters. (b) Separating several groups of emitters.

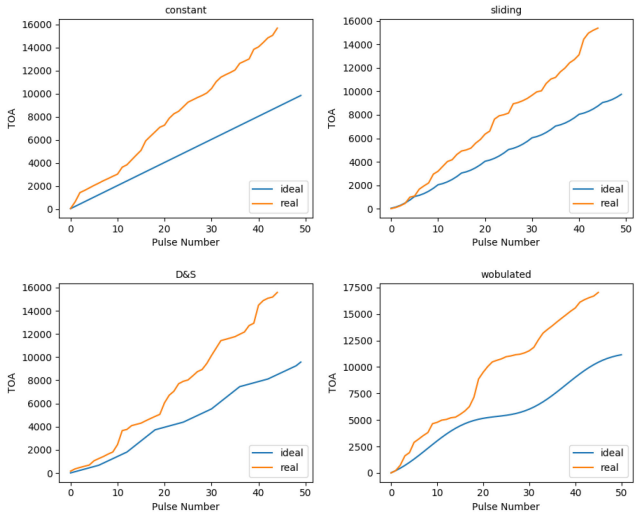


Fig. 2. Four different types of TOA sequences in the ideal environment and real EW environment.

where t_i is the time instant receiving the pulse and N is the number of pulses.

This article discusses four different PRI modulation modes with different characteristics.

Constant PRI: A constant PRI sequence always has a fixed value of k. $p_n = k, \forall n \in \mathbb{Z}+$ for some real number $k > 0$.

Sliding PRI: The values of sliding PRI sequences always monotonously increase (or decrease) to the maximum (or minimum) value, and then suddenly decrease (or increase) to the minimum (or maximum) value. $p_n = PRI_0 + \delta * (n \bmod M)$ where $PRI_0 > 0$ is the initial PRI value of a slide period, $\delta \in \mathbb{R}$ is a value indicating the rate of change in PRI during the slide period, and M is the number of pulses in each slide window.

Dwell and Switch (D&S) PRI: There are several stable values in the D&S PRI sequence, but the values may remain at the same value $y_i > 0 \forall i$ for several pulses $x_i - x_{i-1}$ before switching to another value $y_{i+1} > 0$ in this case.

Wobulated PRI: The values of the wobulated PRI sequence always have a shape similar to a sinusoidal function and also appear periodically. $p_n = PRI_0 + A \sin(\omega * x_n + \varphi)$ where $PRI_0 > 0$ is the value that PRI oscillates around. A is the amplitude of the modulation, ω is the frequency of the sine function, φ is the phase of the sine function, and x_n is a value proportional to n that defines the sampling resolution.

C. Lost Pulses

In the modern EW environment, electromagnetic crowding and advanced signal emitters make deinterleaving a difficult task due to the large number of received pulses and complex TOA patterns. Low probability of interception (LPI) applications make it even harder to intercept all the pulses. The lost pulse problem grows increasingly severe in this context; lost pulses also occur randomly and may reach 50% of the entire pulse streams under some circumstances as radar pulse signals are received.

When j pulses are lost from the ith pulse to the $(i + j - 1)$th pulse in the received TOA sequence T_n, then the new PRI sequence p'_n can be represented as follows:

$$p'_n = \begin{cases} p_n, & n = 1, 2, \ldots, i-1 \\ p_i + \cdots + p_{i+j-1}, & n = i \\ p_{n+j}, & n = i+1, \ldots, N-j-1 \end{cases}. \tag{3}$$

In an ideal EW environment without measurement noise or lost pulses, the TOA sequences can be easily identified through the TOA sequence curves. However, when the sequence is noisy, it is hard to separate those pulses belonging to the emitters with certain PRI values.

Four TOA sequences with different PRI modulation modes in an ideal environment without lost pulses are marked by blue lines in Fig. 2. The orange lines represent TOA sequences with the same PRI values in a real EW environment with Gaussian noise and 50% lost pulses. These curves are pulse streams from one emitter without any other pulses from aliasing emitters. The four PRI modulation modes investigated here, as discussed above, are constant PRI, sliding PRI, D&S PRI, and wobulated PRI. In an ideal environment, the regulations are reflected in the TOA curves; in a real EW environment, it is difficult to discern the relations among those pulses.

D. How do Autoencoders Assist in the Deinterleaving Task?

Machine learning techniques have been swiftly and extensively developed in recent years to the point that certain machine learning techniques outperform humans on certain tasks [22], [23]. The autoencoder, a typical ANN, has been widely used in denoising and dimensionality reduction applications for data visualization. Selecting pulses-of-interest from an interleaved pulse stream involves deleting those aliasing pulses, which are not needed, so the

deinterleaving problem can also be treated as a denoising problem.

Denoising autoencoders were first presented by Vincent *et al.* [24] in an effort to extract and compose robust features. Denoising autoencoders artificially corrupt input data in order to force a more robust representation to be learned. Many researchers have since investigated denoising autoencoders for use in various contexts. Scholars with opendeep.org, for example, used Theano to build a very basic denoising autoencoder and train it on the MNIST dataset. Rasmus *et al.* [18] used a denoising autoencoder architecture for unsupervised and semisupervised feature learning. Xie *et al.* [25] used deep networks pretrained with denoising autoencoders for image inpainting. Takaaki *et al.* [26] utilized a structured denoising autoencoder for fault detection and analysis. Chinese word segmentation can also performed with a text window denoising autoencoder by Ke *et al.* [27]. Denoising autoencoders have also been used for music removal in speech recognition applications [28].

The successful applications mentioned above take advantage of the extraordinary noise elimination function of denoising autoencoders. The essence of the denoising autoencoder is to add irregular noise to the regular input and learn the inner pattern of the input datasets, then extract the features to perform respective tasks. Such structures are also expected to process time series such as TOA sequences and extract distinguishing features between pulses-of-interest and spurious pulses. Denoising autoencoders also function well in environments with high lost and spurious pulse ratios (e.g., noise in speech recognition or inpainting applications). By extracting intrinsic patterns from pulses-of-interest, the trained denoising autoencoder may be capable of resolving the deinterleaving problem.

III. DEINTERLEAVING MODEL BASED ON DENOISING AUTOENCODER

A denoising autoencoder was used in this article to isolate pulses-of-interest from interleaved TOA sequences. The TOA sequences were preprocessed, the autoencoder model was established for deinterleaving, and the coded TOA sequences were processed as discussed in detail below.

A. Representation of TOA

TOA sequences are measurable parameters that are used to deinterleave pulses. This representation satisfies the requirements for conventional statistical methods [8]–[10], but is not easily understood by machines. The TOA sequences should be digitally processed before being fed into the machine learning model.

The arriving time of pulses $\{t_i\}$ continually increases, so the TOA sequence increases to an ultimately large value. To allow the neural network model to process the input as easily as possible, the TOA sequence was transformed into a binary code sequence with only 0 and 1 here as inspired by our previous work [14]. We added another binary coding method in this article with a more simple expression and higher performance. Given a small unit t_{unit}, valid feature values of the new sequence $\{t_i'\}$ within scopes of $[0, M \cdot t_{\text{unit}}]$ were digitized linearly with respect to units of t_{unit}, where M is the number of units presented in the new TOA sequence

$$t_i' = \begin{cases} 1, & i \cdot t_{\text{unit}} < t_i \le (i+1) \cdot t_{\text{unit}} \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

After digitization, the new TOA sequence $\{t_i'\}$ is represented by digits with tolerable quantization errors.

The appropriate value of t_{unit} is crucial when using this binary representation of the TOA sequence. If t_{unit} is too large, there will be more than one pulse located within the same interval resulting in computational error. If t_{unit} is too small, the dimension of $\{t_i'\}$ will create an undue computational burden. Both situations cause problems in the subsequent neural network computation procedure and affect the overall performance of the system. (A proper setting of all parameters including t_{unit} is introduced in Section V.)

Example TOA sequences for the four types of modulations mentioned above with the binary coding expression are listed in Table I, where t_{unit} is set to 50 and M to 24 to clearly illustrate the regulations. Measurement noise is not included in the table. $p_0(n)$ represents the PRI values of the first period of the sequence.

Take the constant PRI as an example. Assume that the PRI value is 200 μs, then the corresponding TOA sequence is $[200, 400, 600, 800, 1000, 1200, \cdots]$. Given $t_{\text{unit}} = 50$ and $M = 24$, the TOA sequence is [0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1] according to (4).

B. Autoencoder Model for Deinterleaving

The interleaved pulse stream consists of several groups of substreams from different emitters. The interleaved stream from each emitter can be considered a noise-contaminated substream, so the deinterleaving task can be treated as a denoising task. As per the characteristics of autoencoders discussed above, the deinterleaving procedures can be operated with a denoising autoencoder.

An autoencoder is a neural network that copies the input to the output for the purpose of dimension reduction or data denoising. It roughly consists of two parts: the encoder and the decoder. The encoder creates a hidden layer (or several hidden layers), which generates a compressed representation of the input. The decoder then reconstructs the input from the hidden layer. In general, the dimensions of the hidden layer created by the encoder are much lower than the dimensions of the input; in effect, the encoder compresses the input into a reconstruction and then decompresses that code into another representation similar to the original data in form, thus reducing the dimensions of the input. Hinton *et al.* [29] found that this characteristic of the autoencoder remits favorable performance in solving classification problems.

Architecturally, the simplest autoencoder has an input layer, an output layer, and one or more hidden layers between them. This is a feedforward neural network similar

TABLE I
Binary Coding of Four TOA Sequences With Different PRI Modulations

No.	mode	PRI values of one period	TOA binary expressions
1	Constant	$p_0(n) = 200$	[0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1]
2	Sliding	$p_0(n) = 100, 180, 260, 340$	[0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1]
3	Wobulated	$p_0(n) = 252, 273, 180, 420$	[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0]
4	D&S	$p_0(n) = 200, 200, 100, 100, 300, 300$	[0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1]

to a multilayer perceptron, but the output layer of the autoencoder has the same dimensions as its input layer as it reconstructs its inputs.

The encoder and the decoder can be defined as transitions ϕ and ψ, respectively. Given the input X, then

$$\phi : \mathcal{X} \to \mathcal{F} \tag{5}$$

$$\psi : \mathcal{F} \to \mathcal{X} \tag{6}$$

$$\phi, \psi = \arg\min_{\phi,\psi} \|X - (\psi \circ \phi)X\|^2 \tag{7}$$

where $\mathcal{X}$ and $\mathcal{F}$ represent the input space and feature space, respectively. After the transitions ϕ and ψ, the output of the autoencoder is $(\psi \circ \phi)X$. The goal of training the autoencoder is to make input X and output $(\psi \circ \phi)X$ as close as possible. Therefore, the transitions ϕ and ψ can be obtained according to (7).

In the simplest case with one hidden layer, the encoder stage takes the input $\mathbf{x} \in \mathbb{R}^d = \mathcal{X}$ and maps it to $\mathbf{z} \in \mathbb{R}^p = \mathcal{F}$:

$$\mathbf{z} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{8}$$

where σ is an elementwise activation function [e.g., a sigmoid function or a rectified linear unit (ReLU)], $\mathbf{W}$ is a weight matrix, and $\mathbf{b}$ is a bias vector. This encoder's output $\mathbf{z}$ is usually referred to as a code, latent variable, or latent representation. The decoder stage of the autoencoder maps $\mathbf{z}$ to the reconstruction $\mathbf{y}$ of the same shape as $\mathbf{x}$

$$\mathbf{y} = \sigma'(\mathbf{W}'\mathbf{z} + \mathbf{b}') \tag{9}$$

where σ', $\mathbf{W}'$, and $\mathbf{b}'$ for the decoder may differ in general from the corresponding $\boldsymbol{\sigma}$, $\mathbf{W}$, and $\mathbf{b}$ for the encoder depending on the autoencoder design.

Autoencoders are also trained to minimize the reconstruction errors (e.g., squared errors) in order to tune the parameters of the neural network

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2 = \|\mathbf{x} - \sigma'(\mathbf{W}'(\sigma(\mathbf{W}\mathbf{x} + \mathbf{b})) + \mathbf{b}')\|^2. \tag{10}$$

The main purpose of the proposed model is to learn the inner pattern of pulses-of-interest with high noise and lost pulse ratios. Vincent *et al.* [24] proved that an autoencoder can extract and compose a robust representation from a corrupted input, which is useful for recovering the corresponding undistorted input. The structure of our denoising autoencoder model for deinterleaving is shown in Fig. 3.

As shown in Fig. 3, the model operation has three steps: noise addition, encoding, and decoding. In the first step, the coded TOA sequences $\mathbf{x}$ from one emitter are given a large proportion of noise as spurious pulses. The interleaved

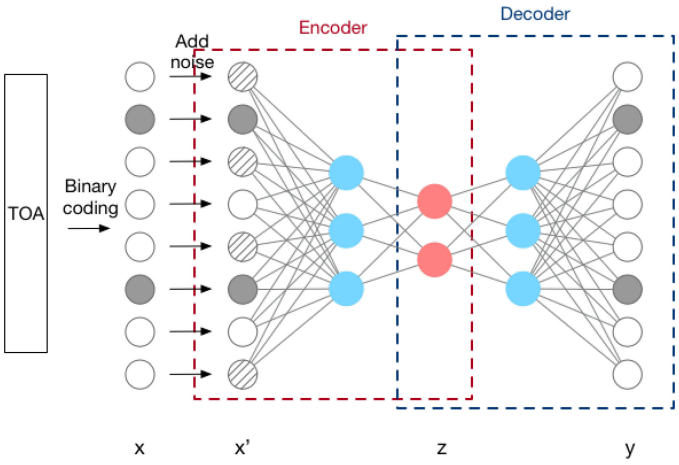


Fig. 3. Structure of proposed denoising autoencoder model for deinterleaving. Coded TOA sequences are supplied with noise and compressed to a shorter representation via the encoder; the decoder decompresses it into a reconstruction similar to the input coded TOA sequence.

sequence $\mathbf{x}'$ is then fed into the autoencoder neural network. Detailed procedures of the proposed model are as follows:

$$\mathbf{z} = \sigma(\mathbf{W}\mathbf{x}' + \mathbf{b}) \tag{11}$$

$$\mathbf{y} = \sigma(\mathbf{W}'\mathbf{z} + \mathbf{b}') \tag{12}$$

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = \{l_1, \ldots, l_N\}^T$$

$$l_n = -w_n[y_n \cdot \log x_n + (1 - y_n) \cdot \log(1 - x_n)]. \tag{13}$$

$\mathbf{x}'$ in (11) represents the interleaved TOA sequence, which is transformed linearly with respect to weight matrix $\mathbf{W}$ and then given a bias vector $\mathbf{b}$ to obtain the compressed vector $\mathbf{z}$ via a logistic sigmoid function $\sigma(\cdot)$. Similarly, the output $\mathbf{y}$ is obtained via $\sigma(\cdot)$ with the sum of bias vector $\mathbf{b}'$ and the product of the weight matrix $\mathbf{W}'$ and $\mathbf{z}$. The purpose is to minimize $\mathcal{L}(\mathbf{x}, \mathbf{y})$ at the loss of $\mathbf{x}$ and $\mathbf{y}$. A binary cross entropy is used here, which is a loss function that measures the error of input and target with numbers between 0 and 1. N in (13) is the batch size of the training phase.

IV. USING AUTOENCODERS FOR DEINTERLEAVING PROBLEM

To effectively learn the inner pattern of TOA sequences and ignore the noise for distinguishing between pulses-of-interest and spurious pulses accurately, the denoising autoencoder should be trained until convergence prior to applying it. Below, the deinterleaving problem with the proposed denoising autoencoder model is presented in detail.

A. Training the Proposed Autoencoder Model

The proposed autoencoder structure has several weight matrices and bias vectors, which must be tuned in order to obtain correct output vectors. In the proposed structure, the interleaved TOA sequences are fed into the network and the targets are the pure TOA sequences of pulses-of-interest (lost pulses are still missing). The output sequences are computed through the autoencoder with the current parameters, then compared with the given targets to compute the binary cross-entropy loss which describes the difference between the input TOA sequence and estimated TOA sequence. Finally, the parameters of the neural network are tuned to minimize any loss and ensure high accuracy. After several tuning iterations, the trained autoencoder is capable of selecting pulses with the determined PRI values.

Before the training starts, all the weight matrices and bias vectors are initialized randomly between the values of 0 and 1. When new interleaved TOA data sample $T = (t_1, t_2, \ldots, t_i, \ldots, t_N)$ is observed, they are digitized and transformed into binary coding vectors as input vectors $\mathbf{y}$, then fed into the autoencoder. The network processes the input vector according to (11)–(13) and outputs a pulse-of-interest vector $\mathbf{x}'$, which is updated recurrently each time a new input vector arrives.

The values of the output vectors are decimals between 0 and 1 after the sigmoid function. The target values are either 0 or 1 representing whether the pulses are of interest. A threshold τ is set to determine the accuracy and measure the precise loss of the network. The values of output x_i' are forced to become 0 or 1 as follows:

$$x_i' = \begin{cases} 0, & x_i' < \tau \\ 1, & \text{otherwise} \end{cases} \tag{14}$$

where τ is always set to 0.5 to ensure a balanced distribution of spurious pulses and pulses-of-interest.

The purpose of the training phase is to tune all the parameters of the autoencoders to minimize the binary cross-entropy loss. The proposed autoencoder thus can gradually and precisely select the pulses-of-interest. The binary cross-entropy loss can reach 0 only when $\mathbf{y} = \mathbf{x}$, which means the output TOA vector is completely the target pulse-of-interest sequence.

Back-propagation is imposed in the training phase to minimize loss and tune the parameters of the autoencoder. Back-propagation is a technique used in ANNs to calculate a gradient that is needed in the calculation of the weights to be used in the network; it is commonly used to train deep neural networks. In the context of machine learning, back-propagation is commonly adopted via gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss function:

$$\alpha_{\text{new}} = \alpha_{\text{old}} - \eta \frac{\partial \text{loss}}{\partial \alpha} \tag{15}$$

where α represents a specific tunable parameter and η is a self-defined positive learning rate smaller than **1**.

B. Deinterleaving via Proposed Autoencoder Model

Test data is supplied to the autoencoder once it has converged. The validation phase serves to reveal the output of the corresponding input via the trained autoencoder. These test data have no labels, so the parameters of the autoencoder do not change in this phase. The output of the autoencoder $\mathbf{y}$ is obtained via (11), (12). The trained autoencoder indicates which elements of the vector should be the pulses-of-interest.

The final output vector $\mathbf{y}$ of the autoencoder is presented here as a binary code with the same shape as the input. The location of the nonzero element indicating pulses-of-interest is searched, so when the last sample of the TOA sequence has been processed, the final outputs are used to calculate the corresponding TOA sequences $T' = (t_1,' t_2,' \ldots, t_i,' \ldots, t_N')$.

A range of TOA values is calculated first according to output $\mathbf{y}$. For each nonzero element y_i of $\mathbf{y}$, the corresponding range of TOA value t_{is} to t_{ie} is calculated as follows:

$$\begin{cases} t_{is} = i \cdot t_{\text{unit}} \\ t_{ie} = (i+1) \cdot t_{\text{unit}} \end{cases} \tag{16}$$

where t_{unit} is the unit number used in (4). For the output $\mathbf{y} = \{y_i\}$, the corresponding range of TOA sequences is $T_r = \{(t_{is}, t_{ie})\}$.

As discussed above, the TOA sequence of pulses-of-interest T' belongs to the interleaved TOA sequence T. Therefore, if the element of T is within the range of T_r, then this element location indicates the pulse-of-interest, i.e., it is an element of T'.

C. Parallel Processing

All the procedures mentioned above comprise the integrated deinterleaving process. Operating these procedures only one time reveals only the pulses-of-interest for one certain PRI modulation. In the real EW environment, certain emitters have different PRI values that need to be found. Parallel processing is crucial to increase the efficiency of this entire deinterleaving process (see Fig. 4).

The TOA sequences are first binary-coded into several vectors as spurious pulses are added into those vectors at random. Those vectors become the prospective inputs of several denoising autoencoders. The networks are trained based on the given TOA sequences and several output vectors are obtained through the trained networks. Those output vectors are then used to calculate the precise location indicating which pulses have the same PRI modulation modes as the given input vectors.

Several denoising autoencoders work simultaneously in the parallel process. Those networks reveal certain pulses according to given PRI regulations and together resolve the deinterleaving problem. The parallel process is highly efficient and allows for certain errors to be tested and corrected during its operation.

TABLE II
Attributes of Simulated PRI Modulation Modes

No.	mode	PRI values (μs)	STD variance	ratio of lost pulses	proportion of spurious pulses
1	Constant	200	$0 \sim 5\%$	$0 \sim 50\%$	$0 \sim 90\%$
2	Sliding	$\{100, 150, 200, 250, 300\}$	$0 \sim 5\%$	$0 \sim 50\%$	$0 \sim 90\%$
3	D&S	$\{110 * 6, 190 * 6, 320 * 6\}$	$0 \sim 5\%$	$0 \sim 50\%$	$0 \sim 90\%$
4	Wobulated	$p = 150 \sin(\frac{\pi}{15} t) + 200$	$0 \sim 5\%$	$0 \sim 50\%$	$0 \sim 90\%$

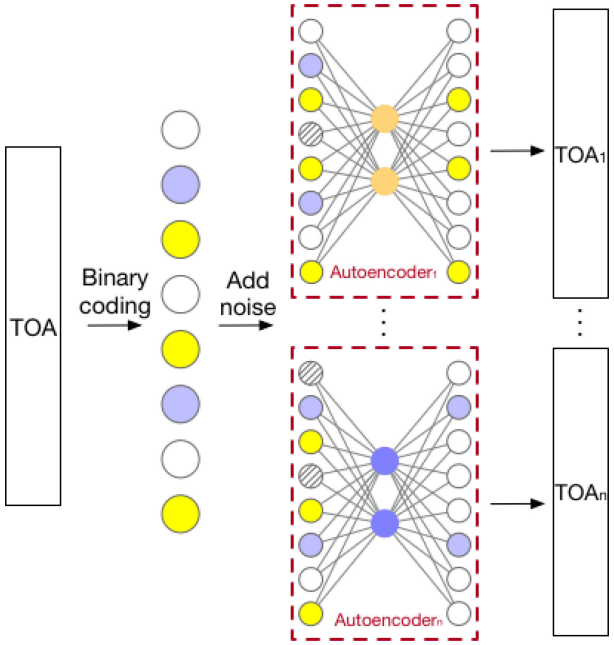


Fig. 4. Parallel processing for deinterleaving tasks. The interleaved TOA sequences are placed into several well-trained autoencoders; the outputs are groups of pulses-of-interest from different emitters.

V. SIMULATIONS

We conducted a series of simulations to test the performance of the proposed autoencoder model for deinterleaving. The TOA sequences considered here are noise-contaminated and have high lost pulse ratios, which makes them hard to separate by the conventional methods described in Section I. Below, the simulation settings are introduced; the results and comparisons among the proposed method and conventional sequence search methods are then given followed by a detailed analysis.

A. Simulation Settings

A number of parameters were used in the simulations mainly involving the PRI sequences and proposed autoencoder. As TOA sequences are the input to the entire model, various TOA sequence settings result in different autoencoder performance indicators. The TOA sequences were designed here to reflect typical examples and modulation types to show that the proposed autoencoder would function well in a real-world deinterleaving environment. The autoencoder parameters were set via cross validation to make for a high-performance deinterleaving task.

1) *Parameters of PRI Sequences:* Four TOA sequences with different PRI modulation modes were considered in this article (see Table II). The PRI values are only a typical example of their modulation modes, while the noise ranges cover most plausible real-world EW scenarios to show the proposed autoencoder's suitability for a wide range of deinterleaving tasks. The attributes of the TOA sequences are listed in Table II.

We generated 10 000 sets of TOA sequences with the four respective PRI modulation modes. As for the binary coding phase, the unit value is set to $t_{\text{unit}} = 10\,\mu$s. The value of t_{unit} is basically selected by the average PRI values and the results of several experiments. We must ensure that as few pulses as possible are located within the same interval, and meanwhile avoiding unnecessary computational burden. Therefore, we have performed several experiments to obtain the proper t_{unit} value. Each PRI sequence first had 200 pulses but only 3000 units remained for each binary TOA sequence to give the inputs of the neural networks the same shape; only about 150 pulses remained for deinterleaving in this case. Gaussian-distributed deviations were also added to the TOA sequences to simulate measurement noise with the mean PRI value and variance ranging from 0 to 5%. Up to 50% of pulses-of-interest may be dropped in the simulation. The proportion of pulses from aliasing emitters was up to 90%. Aliasing pulses were randomly added to the TOA sequences.

2) *Parameters of Denoising Autoencoders:* The proposed denoising autoencoder model for deinterleaving was trained for selecting pulses-of-interest from the interleaved TOA sequences. Four networks were established for TOA sequences with four different PRI modulations and 10 000 pulses were generated to train the network. The entire network structure consists of two basic parts: the encoder and the decoder. For each training or validation sample, the detailed parameter settings were as follows.

a) *Encoder:* Two fully connected layers were included with 128 and 32 as the number of neurons for each layer. ReLUs followed by every layer. The resulting encoder output shape was 32×1.
b) *Decoder:* Two fully connected layers were included, one with 32 and one with 128 neurons. The ReLU followed every layer. The shape of the decoder's output in this case was the same as the encoder's input, which reflects the TOA sequence dimensions 1500×1. To complete the process, a sigmoid function was performed.

The proposed autoencoder deinterleaving model was trained on the Pytorch platform with a batch size of 64 and a learning rate of $\eta = 0.001$. The weight decay was

TABLE III
Training Accuracy of TOA Sequences With Different PRI Modulations

No.	PRI Modulation	Training accuracy (%)
1	Constant	96.33
2	Sliding	96.19
3	Dwell and Switch	96.17
4	Wobulated	95.83

set to 1×10^{-5}. Each batch was selected randomly from the corresponding datasets; note that 10 epochs is usually sufficient for such a network to converge.

B. Results

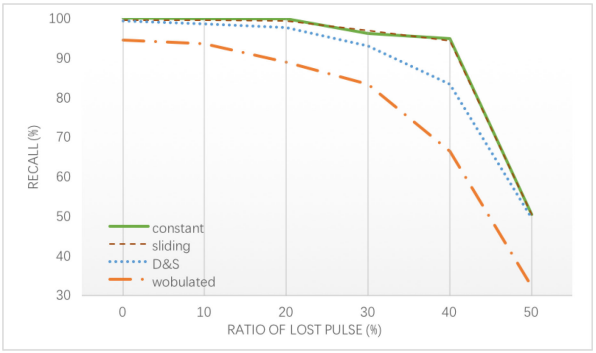
There are two main steps to the deinterleaving process: the training phase and the validation phase. Training the proposed autoencoder serves to make each network capable of extracting the inner pattern from the given TOA sequences under various circumstances. Validation serves to test the performance under different certain environmental settings. The training datasets must include all possible situations, while validation datasets include different situations with several known noise ratios.

During the training phase of our test, TOA sequences under all possible scenarios with a wide range of noise ratios were fed into the proposed model. Once the neural networks converged, the training accuracy of TOA sequences with different PRI modulations were as listed in Table III.

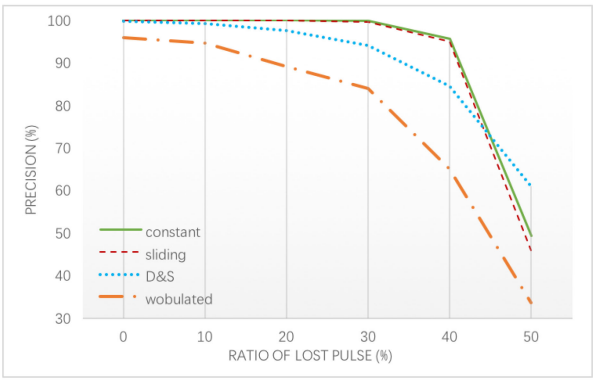
As shown in the table above, the training accuracies of different TOA sequences were all above 95%. The PRI modulations did not significantly affect the model's performance, which suggests that the proposed autoencoder is robust to all manner of PRI modulation modes.

In the validation phase, samples with different proportions of lost pulses were fed into the trained autoencoders for four independent PRI modulation modes. We generated 1000 samples for each circumstance to test the performance with 0–90% pulses from other emitters of the neural network.

The goal of deinterleaving is to isolate the pulses-of-interest from all the intercepted pulses, so we counted the numbers of four cases for performance evaluation: 1) True Positive (TP), where pulses-of-interest are detected as pulses-of-interest; 2) True Negative (TN), where spurious pulses are detected as spurious pulses; 3) False Positive (FP), where spurious pulses are detected as pulses-of-interest; and 4) False Negative (FN), where pulses-of-interest are detected as spurious pulses. Deinterleaving performance was evaluated in two criteria, namely, the Recall and Precision [30]. Recall is the ratio of correctly predicted positive observations to all observations in the actual class and precision is the ratio of correctly predicted positive observations to the total predicted positive observations. In simple terms, Recall indicates how many of the pulses-of-interest are correctly detected by (17). Precision indicates the ratio of actual pulses-of-interest in the collection of samples being detected as pulses-of-interest in (18)



(a)



(b)

Fig. 5. Proposed autoencoder in first scenario with increasing ratios of lost pulses measured by (a) Recall, (b) Precision.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{17}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{18}$$

The deinterleaving results can be divided into two scenarios as mentioned in Section II. The first scenario involves choosing pulses from one emitter from all the random spurious pulses. The other involves separating pulses from several emitters.

Scenario 1: For the first situation, TOA sequences with one known PRI values were first simulated with pulses from other emitters and noise. The obtained statistical Recall and Precision are shown in Fig. 5.

In Fig. 5, the ratio of lost pulses varies from 0 to 50%. Recall and Precision both decrease due to an increase in missing information on TOA sequences. The four lines representing four different PRI modulation modes show a similar trend with good performance when the lost pulse ratio is lower than 50%, indicating that the proposed autoencoder can distinguish pulses from noise with all manner of PRI modulation modes. TOA sequences with constant PRI and sliding PRI show the best results because of their stable regularity. Relatively speaking, TOA sequences with wobulated PRI modulation perform worse than the other three because the PRI values change with time instant t, but

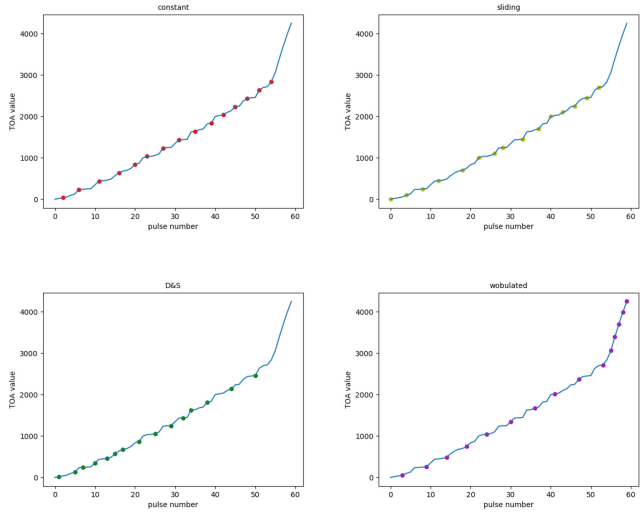


Fig. 6. Results of deinterleaving four PRI modulation modes in the ideal environment.

still show around 70% of Recall and Precision when 40% of the pulses are lost.

Scenario 2: We next simulated TOA sequences with multiple PRI modulations and attempted to isolate the pulses with a given PRI value. Using this trained model to test the second scenario is actually very similar to the first scenario. Whether these pulses were randomly generated or with certain PRI modulations, they are ideally considered to be noise and eliminated in the proposed model. This is because the target of the autoencoder is the pulse-of-interest, so the model only learns the pulses-of-interest features. (The detailed statistical results are thus very similar and are not fully listed here.)

The TOA sequences including four PRI modulation modes (see Table II) were tested in an ideal circumstance to observe how the pulses-of-interest would be reflected in the TOA curves. The results are shown in Fig. 6. The interleaved TOA sequences are marked by blue lines. Dots of different colors represent the pulses-of-interest in four independent TOA sequences. The four TOA sequences appear to be isolated from the system with almost 100% accuracy. These experiments altogether indicated that well-trained autoencoders can accommodate situations with several known emitters.

A unique scenario emerged as we tested the second scenario that was even harder to deinterleave, where streams consisted of two groups of pulses with the same PRI values. In the real EW environment, there may be two emitters with the same parameters transmitting signals simultaneously while one is much farther away, thus causing a large number of lost pulses. This situation is very complicated due to the same pattern that must be extracted at the same time while the other signal is superfluous. In this situation, there are two emitters with the same PRI values in one TOA sequence; one is intercepted as normal with the four kinds of PRI modulation modes mentioned in Table II while the other has different ratios of lost pulses. We simulated this scenario while also ensuring that the first pulse received came from

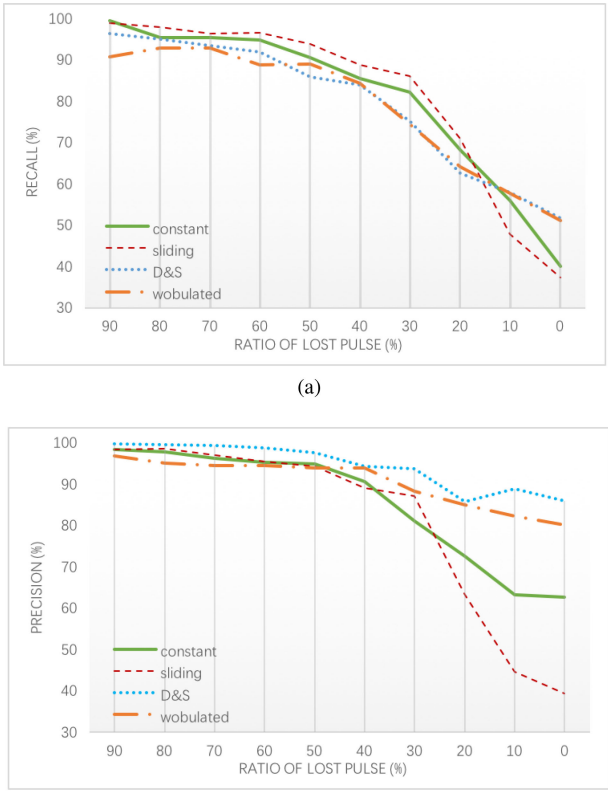
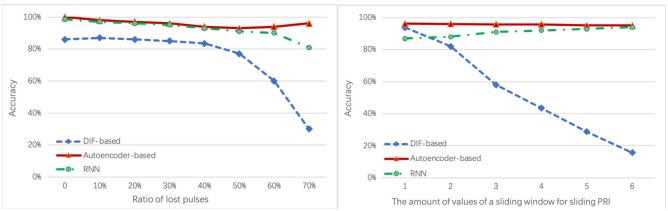


Fig. 7. Performance of proposed autoencoders dealing with pulses with same-PRI-value pulses measured by (a) Recall, (b) Precision.

the emitter-of-interest. The results reflect how the proposed autoencoder can select pulses belonging to the first emitter as per the resulting Recall and Precision values (see Fig. 7).

Fig. 7 shows the performance of the proposed model with decreasing lost pulse ratios of the pulses of a secondary ("disturbing") emitter which has the same PRI values as the target pulses. In most cases, the performance is favorable with around 80% Recall and 90% Precision when the lost pulse ratio of the disturbing emitter is larger than 30%. When the two emitters have the same number of pulses being intercepted, the Precision and Recall of the sliding PRI modulation is relatively low. This is because the sliding TOA has repeatable PRI values and the two sliding TOA sequences are quite easily confused. However, the majority of problems in the real-world EW environment can be well resolved by the proposed method—the exceptions being extreme and fairly rare situations.

We also tested a DIF-based method and RNNs for comparison against the proposed model. The DIF-based method uses a sequential search to identify pulses-of-interest. The results are shown in Fig. 8. The criterion *accuracy* measures how many pulse-of-interest and noise elements are correctly recognized. As shown in Fig. 8(a), the deinterleaving accuracy of DIF-based method decreased dramatically as the lost pulse ratio increased while the proposed autoencoder method fluctuated only slightly down to 90%. The DIF-based method, as mentioned above, uses a sequential search to isolate the pulses with PRI values of certain measurement



(a) (b)

Fig. 8. Deinterleaving accuracies using DIF-based method, RNNs, and autoencoder-based method with (a) different lost pulse ratios, (b) different sliding PRI vector element quantities.

errors; if there are missing pulses, the next pulse cannot be detected as a pulse-of-interest.

Fig. 8(b) shows the changes in accuracy as different amounts of values of a sliding window for sliding PRI were simulated. As the figure shows, the DIF-based method did not correctly deinterleave the TOA sequences as sliding PRI values increased while the proposed autoencoder method stayed robust between sliding PRI of 2-7. The TOA sequence grows more complex as more elements are contained in the PRI vector, which makes it difficult to identify correct pulses using conventional sequence search methods. The proposed autoencoder model outperformed the RNNs in this test. The main reason we did not use RNNs here is that they have a very high computational burden.

Accordingly, our proposed autoencoder model for deinterleaving can not only pick out pulses of-interest, but also is robust to the environment with a high ratio of lost pulses. TOA sequences with complex PRI modulation modes can be deinterleaved as well.

C. Analysis and Discussion

There are four main reasons that the autoencoder performs well in solving the deinterleaving problem.

1) *Purpose:* Deinterleaving is, essentially, a denoising problem. The purpose of deinterleaving is to identify pulses-of-interest—other pulses (e.g., noise or aliasing emitters) can all be considered noise. Denoising is the basic function of the autoencoder, making it well-suited to the deinterleaving problem.
2) *Shape uniformity:* The input and output to the autoencoder must have the same shape. The deinterleaving input and output are both pulses in the same time period with the same length of data, which makes them uniform in shape and thus well suited to the autoencoder.
3) *Lost adaption:* Autoencoders can adapt to high ratios of missing data (corrupted data) due to their low-dimensional hidden layers. The most problematic aspect of deinterleaving is the inherently high ratio of lost pulses; this makes the autoencoder particularly advantageous.
4) *Simplicity:* The basic autoencoder has a simple, straightforward structure and can be operated at a relatively low computational burden. It is able to complete the necessary processing swiftly and efficiently. This is vital for deinterleaving problems, as they require a large amount of data to be processed in real time.

VI. CONCLUSION

A denoising autoencoder model was established in this study to resolve deinterleaving problems using TOA sequences as inputs. The model was trained, then tested in various environments which may markedly diverge from the training datasets. The model proved capable of learning the inner connections of pulses-of-interest and isolating them from interleaved pulses on the basis of their PRI modulation modes. Simulation results show that this model can mine and abstract statistical and local patterns of TOA sequences with different PRI modulation modes; the mined patterns can be easily utilized to select pulses-of-interest from the interleaved TOA sequences. The proposed method was found to solve deinterleaving problems even in complex environments with a high lost pulse ratios, which demonstrates its robustness as well as its efficiency.

In the future, we plan to consider two other, more extensive deinterleaving problems to further test the proposed method.

1) *Blind deinterleaving problem:* In this article, the proposed method performed well on a deinterleaving problem with known PRI values. We assert that our method is also adaptable to more complex environments with parameters ranging beyond the scope tested here. In the real EW environment, emitters have several patterns within a certain range and can also be deinterleaved by our structure. Certain emitters can be trained before the deinterleaving process, then the pulses-of-interests can be isolated with the trained neural network. The proposed autoencoder can learn the inner pattern of given TOA samples and its robustness to high lost pulse ratios would prove advantageous in this context as well.

2) *Application in measured data:* The deinterleaving problem solved in this article was based entirely on simulation data, which allowed us to conveniently evaluate the performance of the proposed method. It is yet necessary to test the proposed method in resolving deinterleaving problems in a real EW environment with real measurements. Autoencoders may perform tasks differently on simulated versus measured data. Transfer learning, which centers on storing knowledge gained while solving one problem and applying it to different but related problems, may be a workable approach to this. The deinterleaving of simulated data and measured data are two related problems with different inputs; combining transfer learning and the proposed autoencoder model may help to resolve problems in the real EW environment.

REFERENCES

[1] Y. Liu and Q. Zhang
Improved method for deinterleaving radar signals and estimating pri values
IET Radar, Sonar Navigati., vol. 12, no. 5, pp. 506–514, 2018.

[2] R. G. Wiley
ELINT: The Interception and Analysis of Radar Signals. Norwood, MA, USA: Artech House, 2006.

[3] D. K. Barton
Radar System Analysis and Modeling. vol. 1. Norwood, MA, USA: Artech House, 2004.

[4] H. Arslan
Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems. Berlin, Germany: Springer, 2007.

[5] D. C. Robbins, R. K. Sarin, E. J. Horvitz, and E. B. Cutrell
Advanced navigation techniques for portable devices
U.S. Patent 7 327 349, Feb. 2008.

[6] M. A. Richards, J. Scheer, W. A. Holm, and W. L. Melvin
Princ. of Modern Radar. Institution of Engineering & Technology, 2010.

[7] R. G. Wiley
Electronic Intelligence: The Analysis of Radar Signals. Norwood, MA, USA: Artech House, 1982, p. 250, 1982.

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

[9] D. Milojevic and B. Popovic
Improved algorithm for the deinterleaving of radar pulses
IEE Proc. F-Radar Signal Process., vol. 139, no. 1, pp. 98–104, Feb. 1992.

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

[11] A. Mahdavi and A. M. Pezeshk
A fast enhanced algorithm of PRI transform
In *Proc. IEEE 6th Int. Symp. Parallel Comput. Elect. Eng.*, 2011, pp. 179–184.

[12] Y. Mao, J. Han, G. Guo, and X. Qing
An improved algorithm of PRI transform
In *Proc. Intell. Syst. WRI Global Congr.*, 2009, vol. 3, pp. 145–149.

[13] A. Ata'a and S. Abdullah
Deinterleaving of radar signals and PRF identification algorithms
IET Radar, Sonar Navigat., vol. 1, no. 5, pp. 340–347, 2007.

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

[15] D. H. Ballard
Modular learning in neural networks
In *Proc. 6th Nat. Conf. Artif. Intell.*, 1987, pp. 279–284.

[16] H. Noh, S. Hong, and B. Han
Learning deconvolution network for semantic segmentation
In *Proc. Int. Conf. Comput. Vision*, 2015, pp. 1520–1528.

[17] J. Yang, S. E. Reed, M. Yang, and H. Lee
Weakly-supervised disentangling with recurrent transformations for 3D view synthesis
In *Proc. Neural Inf. Process. Syst.*, 2015, pp. 1099–1107.

[18] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, and T. Raiko
Semi-supervised learning with ladder networks
In *Proc. Neural Inf. Process. Syst.*, 2015, pp. 3546–3554.

[19] K. Rematas, T. Ritschel, M. Fritz, E. Gavves, and T. Tuytelaars
Deep reflectance maps
In *Proc. Comput. Vision Pattern Recognit.*, 2016, pp. 4508–4516.

[20] J. J. Zhao *et al.*,
Stacked what-where auto-encoders
Mach. Learn., 2015.

[21] A. Newell, K. Yang, and J. Deng
Stacked hourglass networks for human pose estimation
In *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 483–499.

[22] D. Silver
et al., Mastering the game of go with deep neural networks and tree search
Nature, vol. 529, no. 7587, pp. 484–489, 2016.

[23] D. Silver
et al., Mastering the game of go without human knowledge
Nature, vol. 550, no. 7676, pp. 354–359, 2017.

[24] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol
Extracting and composing robust features with denoising autoencoders
In *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.

[25] J. Xie, L. Xu, and E. Chen
Image denoising and inpainting with deep neural networks
Adv. Neural Inf. Process. Syst., vol. 25, pp. 341–349, 2012.

[26] T. Tagawa, Y. Tadokoro, and T. Yairi
Structured denoising autoencoder for fault detection and analysis
In *Proc. 6th Asian Conf. Mach. Learn.*, 2015, vol. 39, pp. 96–111.

[27] K. Wu, Z. Gao, C. Peng, and X. Wen
Text window denoising autoencoder: Building deep architecture for Chinese word segmentation
In *Proc. Natural Lang. Process. Chin. Comput.*, 2013, pp. 1–12.

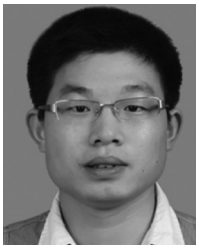
[28] M. Zhao, D. Wang, Z. Zhang, and X. Zhang
Music removal by convolutional denoising autoencoder in speech recognition
In *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2015, pp. 338–341.

[29] G. E. Hinton and R. R. Salakhutdinov
Reducing the dimensionality of data with neural networks
Sci., vol. 313, no. 5786, pp. 504–507, 2006.

[30] D. M. Powers
Evaluation: From precision, recall and f-measure to ROC, informedness, markedness and correlation
J. Mach. Learn. Technol., vol. 2, no. 1, pp. 37–63, 2011.

Xueqiong Li received the B.S. degree in communication and information engineering from the College of Transportation, Southeast University, Nanjing, China, in 2013, and the M.S. degree in communication and information engineering in 2015 from the College of Electronic Science and Engineering, National University of Defense Technology, Changsha, China, where she is currently working toward the Ph.D. degree in communication and information engineering.

Her current research interests include signal processing and machine learning.



Zhangmeng Liu received the Ph.D. degree in statistical signal processing from National University of Defense Technology, Hunan, China, in 2012.

He is currently an Associate Professor with NUDT working in the interdiscipline of electronics engineering and computer science, especially electronic data mining.

Zhitao Huang received the B.S. and Ph.D. degrees in information and communication engineering from the College of Electronic Science and Engineering, National University of Defense Technology, Changsha, Hunan, China, in 1998 and 2003, respectively.

He is currently a Professor with the College of Electronic Science and Engineering, National University of Defense Technology. His research interests include radar and communication signal processing, array signal processing.