

Deinterleaving of Intercepted Radar Pulse Streams via Temporal Convolutional Attention Network

Haojian Wang, Zhenji Tao, Jin He, Ting Shu

Abstract—In the domain of electronic warfare, radar signal deinterleaving emerges as the foundational and indispensable phase of electronic reconnaissance. The ever-increasing complexity of electromagnetic environments, further compounded by technological advancements such as multi-function radars (MFRs), has led to the inadequacy of traditional deinterleaving techniques. To tackle these challenges, this paper introduces the Temporal Convolutional Attention Network (TCAN) framework. This framework harmoniously combines a temporal convolutional network (TCN) with an advanced attention mechanism, thereby significantly enhancing the system's signal sorting proficiency. Through rigorous experimental validation, we demonstrate that TCAN consistently outperforms existing baseline methods. This superiority is particularly pronounced under conditions of signal sparsity and in non-ideal environments, which are typified by pulse loss, spurious pulses, and measurement errors. Furthermore, we conduct a thorough analysis to elucidate the profound impact of various input formats and multi-parameter features on deinterleaving performance. By meticulously examining these factors, we establish TCAN as a robust and versatile solution capable of effectively navigating the heightened complexity of modern radar signal environments. Our findings highlight TCAN's potential as a potent instrument for augmenting electronic reconnaissance capabilities amidst evolving electromagnetic challenges.

Index Terms—Electronic warfare, radar signal processing, radar signal sorting, temporal convolutional attention network (TCAN).

I. INTRODUCTION

Electronic Support Measures (ESM) systems play a crucial role in electronic reconnaissance, undertaking the vital tasks of intercepting, identifying, and analyzing signals from hostile radar systems. The importance of ESM systems has surged in modern electronic warfare, as evidenced by recent scholarly works (e.g., [1]) and vividly demonstrated in Fig. 1. A fundamental technology within ESM systems is Radar Signal Sorting (RSS), also known as deinterleaving, which separates intertwined pulse streams [2]. This initial sorting is essential for subsequent accurate and effective signal source classification, identification, and tracking [3].

However, the increasing complexity of the electromagnetic environment, marked by radio frequency spectrum congestion [4], poses significant challenges for ESM systems in deinterleaving densely mixed pulse streams. This challenge

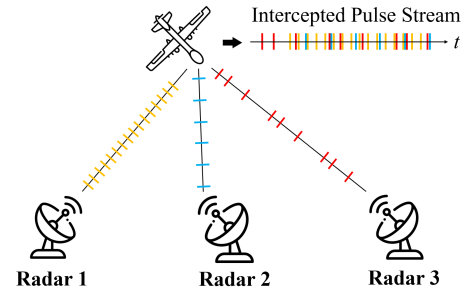


Fig. 1. An intercepted pulse stream from radar emitters.

is further compounded by the proliferation of radar emitters. Radar pulses are characterized by the Pulse Description Word (PDW), which includes attributes such as Time of Arrival (TOA), Pulse Width (PW), Radio Frequency (RF), and Direction of Arrival (DOA). Among these attributes, TOA stands out as the most critical parameter. It not only pinpoints the exact instances when different radar pulses are captured by the signal interception receiver but also acts as a vital feature for unveiling the underlying, manually configured parameters of the emitter. In the case of a pulse stream originating from a single radar source, the first-order difference of its TOA sequence, termed the Difference of TOA (DTOA), often exhibits a distinctive pattern. This regularity, known as the Pulse Repetition Interval (PRI), encapsulates vital classified information, including radar detection range and range resolution. As a fundamental indicator of radar operation modes and a cornerstone for threat assessment, PRI holds a pivotal position in electronic reconnaissance and situational awareness.

Existing radar signal sorting methods can be broadly categorized into TOA-based and multi-parameter-based approaches. TOA-based methods, which primarily leverage TOA for signal separation, include algorithms like Cumulative Difference Histogram (CDIF) [5], Sequential Difference Histogram (SDIF) [6], and PRI Transform [7]. These methods extract potential PRIs through histogram analysis and search for corresponding pulses in the sequence. Despite various improvements [8]–[10], they grapple with issues such as PRI jitter, environmental noise, and pulse loss. On the other hand, multi-parameter-based methods utilize all features of the PDW for RSS, employing a subset or the entirety of PDW components as inputs. These methods, often employing clustering techniques [11]–[20], differentiate signal sources based on similarities among parameters like RF, DOA, and PW. However, they presuppose similar parameter distributions for pulses originating from the same radar, which may be disrupted by Multi-Function

Manuscript received . This work was supported by the Natural Science Foundation of Shanghai under Grant 24ZR1431700. Haojian Wang, Jin He, and Ting Shu are with the Shanghai Key Laboratory of Intelligent Sensing and Recognition, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, 200240, P. R. China. Zhenji Tao is with the Shanghai Institute of Mechanical and Electrical Engineering, Shanghai, China.

Radars (MFRs) with wide-range agile parameters and multiple operation mode.

To address these challenges, recent research has pivoted towards deep learning techniques, aiming to enhance the adaptability and efficacy of signal sorting. Convolutional Neural Networks (CNNs) have been applied to deinterleaving by transforming PDWs into images [21], [22]. However, this approach is sensitive to image resolution, whereby lower resolutions can degrade sorting accuracy, while higher resolutions can increase computational complexity. Additionally, the process of transforming PDWs into images introduces extra latency, which further compromises real-time performance. Temporal deep learning models, such as Recurrent Neural Networks (RNNs), have also been explored [23]–[26]. These methods frame deinterleaving as a prediction task, utilizing part or all of the PDW data to train an RNN. Advanced algorithms enable sorting multiple signals using a single trained model, reducing training costs. Researchers have also approached RSS from alternative perspectives, such as framing deinterleaving as a denoising challenge [27] or employing blind signal separation principles [28]. These innovative approaches underscore the versatility of deep learning techniques in tackling RSS challenges.

Despite the advancements, two pivotal challenges persist: the selection of an optimal input format and the mitigation of signal sparsity in realistic scenarios. A comprehensive analysis of input format selection is absent from the existing literature, with most studies limiting their discussions to qualitative assessments [25]. Furthermore, the literature frequently models pulse streams as contiguous sequences, overlooking the inherent signal sparsity encountered in practical reconnaissance situations. This sparsity arises from the constrained beamwidth of the main lobe, leading to discontinuous yet periodic pulse segments. Addressing these challenges is imperative for the progression of radar signal sorting in increasingly complex electromagnetic environments.

In this paper, we embark on an exploration rooted in end-to-end prediction methodologies, introducing a novel deinterleaving algorithm that integrates the Temporal Convolutional Network (TCN) [29] with the attention mechanism [30]. This synthesis is termed the Temporal Convolutional Attention Network (TCAN). Our investigation focuses on the RSS task, leveraging TOA data alongside multi-parameter inputs. To achieve a comprehensive understanding, we examine two prevailing input formats within the context of this task. Through a rigorous experimental framework, we systematically assess various combinations of neural networks and input formats, meticulously analyzing the distinctive attributes, strengths, and constraints associated with each configuration. Furthermore, we delve into the impact of signal sparsity on sorting performance via targeted experiments, aiming to elucidate the intricate relationship between these factors.

The principal contributions of this research are outlined as follows:

- 1) **Novel Framework:** We propose TCAN, a novel framework that seamlessly integrates the foundational TCN model with an attention mechanism. By embedding self-attention within the TCN architecture, our model

demonstrates enhanced proficiency in capturing pivotal temporal features. Experimental outcomes affirm that TCAN consistently outperforms other baseline methodologies, establishing its superiority.

- 2) **Signal Sparsity Analysis:** To our knowledge, this study is the first to examine the phenomenon of signal sparsity arising from restricted main lobe illumination. While existing literature often models pulse streams as continuous entities, addressing challenges such as measurement inaccuracies, pulse omissions, and spurious pulses, our research delves into the nuances of radar detection in reconnaissance contexts. Specifically, the constraints imposed by the main lobe's beamwidth necessitate scanning for comprehensive coverage, resulting in intercepted signals that contain pulse sequences exclusively during main lobe illumination. This leads to discontinuous segments, which we term as signal sparsity.
- 3) **Input Format Evaluation:** Previous research has typically relied on qualitative analysis to select a singular input format, overlooking the impact of different input forms on sorting performance. In contrast, this study builds upon qualitative analysis by incorporating quantitative experimental research. We systematically evaluate two primary input formats: DTOA input and binary input. Our experimental findings elucidate the respective strengths, limitations, and applicability of each format across various scenarios, providing valuable insights for future research endeavors.

The remainder of this paper unfolds as follows. Section II offers a comprehensive introduction to the deinterleaving task, elucidating the mathematical underpinnings and characteristic features of the signal. Furthermore, it enumerates the intricate challenges confronted by interception systems in this context. Section III delineates the methodology adopted in this research, encompassing data preprocessing techniques and delving into the intricate details of their implementation. In Section IV, we present the experimental design and setup, followed by an elaborate analysis of the experimental results, offering insights into the efficacy and performance of our proposed method. Finally, Section V summarizes the key findings and contributions of this research.

II. PROBLEM FORMULATION

In the domain of electronic warfare, radar reconnaissance systems employ signal interception receivers with expansive apertures that cover both spatial and frequency domains. This enables the capture of signals from multiple enemy radar emitters. Consequently, the intercepted signal sequences often consist of interleaved pulses originating from various radar sources, as illustrated in Fig. 1. The primary objective of RSS is to disentangle these interleaved signals by leveraging PDWs extracted from the intercepted sequences. This allows for the subsequent identification and comprehensive analysis of diverse radar emitters.

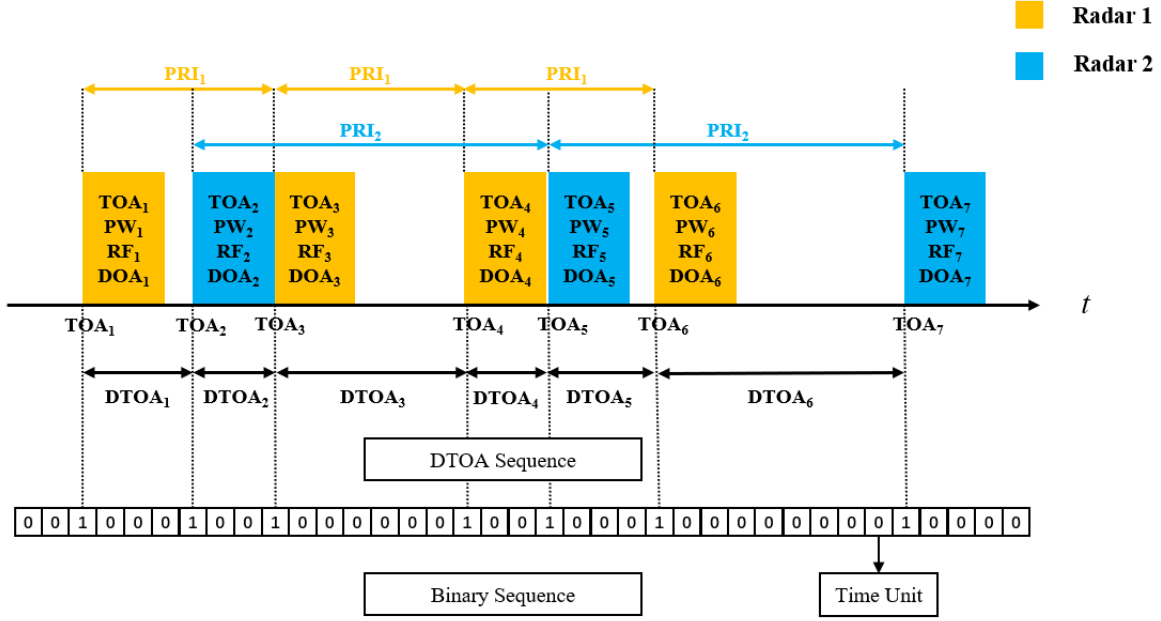


Fig. 2. Two representations of an interleaved sequence.

A. Representation of Pulse Sequences

Before delving into the mathematical formulation of signal sorting, it is crucial to establish a solid mathematical representation for pulse sequences. An intercepted pulse sequence obtained from an interception receiver typically manifests as a discrete sequence of PDWs. This sequence amalgamates several key parameters: TOA, Pulse Width (PW), Radio Frequency (RF), and DOA. As illustrated in Fig. 2, the PDWs of a pulse sequence, which encompasses mixed radar pulses from multiple radar sources, are ordered in ascending TOA. The PDW of the i -th single pulse p_i and the entire PDW sequence P can be mathematically represented as follows:

$$p_i = \{\text{TOA}_i, \text{PW}_i, \text{RF}_i, \text{DOA}_i\} \quad (1)$$

$$P = \{p_1, p_2, \dots, p_N\} \quad (2)$$

where N signifies the total number of pulses captured within the reception window. The primary objective of RSS is to discern a mapping relationship that accurately assigns each pulse to its respective radar source based on the PDW sequence.

Assuming the presence of M radar sources, the i -th pulse is assigned a unique label y_i , $1 \leq y_i \leq M$, and the corresponding label sequence is denoted as $Y = \{y_1, y_2, \dots, y_N\}$. The mathematical formulation of RSS can be succinctly expressed as:

$$\text{Determine a function } f \text{ such that } f(P) = Y \quad (3)$$

Among the various parameters, TOA stands out as an implicit and intrinsic feature of the PDW sequence, containing a wealth of information compared to other parameters. Within the temporal attributes of a radar signal, the PRI is paramount, and it can be extracted from the TOA. Considering an ideal scenario where a single radar transmits a pulse stream intercepted by a perfect interception system, devoid of any pulse loss or measurement error, the PRI can be determined through

the first-order difference of the TOA sequence, commonly referred to as the DTOA. This PRI encapsulates crucial information such as the radar's operating mode, detection range, and range resolution, all of which are intrinsic to the radar and frequently classified as confidential parameters by numerous nations.

Due to the existence of interleaved pulse streams and non-ideal conditions that alter the original PRI characteristics, it is not possible to directly extract the PRI from intercepted TOA or DTOA data. As depicted in Fig. 2, prior to successful signal sorting, the DTOA merely represents the temporal difference between consecutive pulse arrivals. The impact of non-ideal conditions on this degradation will be further analyzed in Section II-C. Moreover, to reduce variance and ensure the stationarity of input data without compromising any information, DTOA is often employed as a proxy for TOA in various algorithms. This substitution significantly enhances the efficiency and stability of most of these algorithms. Consequently, the actual input x_i to these algorithms can be expressed as follows for TOA-based and multi-parameter-based approaches, respectively:

$$x_i = \{\text{DTOA}_i\} \quad (4)$$

$$x_i = \{\text{DTOA}_i, \text{PW}_i, \text{RF}_i, \text{DOA}_i\} \quad (5)$$

B. PRI Modulation

In this paper, we consider four prevalent PRI modulation types that are encountered in realistic electromagnetic environments. These include fixed PRI, jitter PRI, stagger PRI, and dwell & switch PRI. For each of these typical PRI patterns, both a precise definition and a corresponding schematic diagram (Fig. 3) are provided as follows.

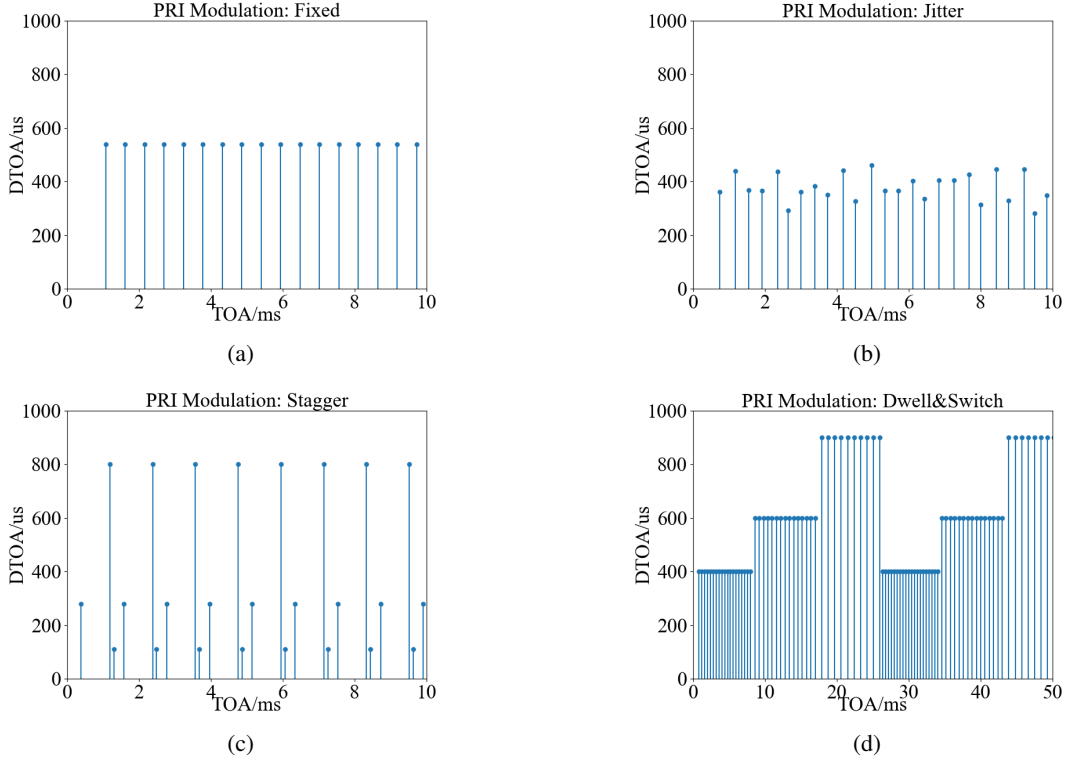


Fig. 3. DTOA patterns of different PRI modulations. (a) Fixed PRI. (b) Jitter PRI. (c) Stagger PRI. (d) Dwell&Switch PRI.

- 1) **Fixed PRI:** This modulation type is marked by a consistent PRI value throughout the entire transmission process. Mathematically, this can be expressed as:

$$\text{PRI}(i) = \text{PRI}_0, \quad i \in \mathbb{N} \quad (6)$$

where PRI_0 denotes the constant value of PRI.

- 2) **Jitter PRI:** Jitter PRI signals exhibit variations around a central value. These fluctuations often adhere to a Gaussian or uniform distribution, and the mathematical representation is given by:

$$\text{PRI}(i) = \text{PRI}_0 + J \cdot \delta, \quad \delta \sim \mathcal{U}(-\text{PRI}_0, \text{PRI}_0) \quad (7)$$

Here, PRI_0 represents the center value of the jitter PRI, while J denotes the jitter rate, indicating the amplitude of the fluctuations.

- 3) **Stagger PRI:** Stagger PRI signals are characterized by a sequence of multiple, distinct PRI values that appear periodically in a specific order. The mathematical model for this pattern is:

$$\text{PRI}(i) = \text{PRI}_k, \quad k = i \bmod m. \quad (8)$$

Here, m denotes the number of unique PRI values, specifically $\text{PRI}_0, \text{PRI}_1, \dots, \text{PRI}_{m-1}$.

- 4) **Dwell & Switch PRI:** This pattern consists of several pulse groups that repeat in a predefined order, akin to Stagger PRI. However, a key difference lies in the varying number of pulses within each group, all maintaining the same PRI value within a given group. The PRI sequence for this pattern can be described as:

$$\text{PRI}(i) = \text{PRI}_k, \quad L_{k-1} \leq i \bmod L < L_k \quad (9)$$

In this context, m also represents the number of different PRI values in the pulse groups, denoted as $\text{PRI}_0, \text{PRI}_1, \dots, \text{PRI}_{m-1}$. Additionally, l_j signifies the length of each pulse group, where $j = 0, 1, \dots, m-1$. We further define $L_k = \sum_{j=0}^k l_j$ and $L = L_{m-1}$ to delineate the boundaries of the pulse groups.

The characteristic distinct DTOA pattern corresponding to each PRI modulation type is graphically represented in Fig. 3. While other parameters also undergo modulation processes akin to this, a comprehensive elaboration on those aspects is omitted from the present discussion.

C. Non-ideal Conditions

Operating radar signal interception receivers in realistic electromagnetic environments presents significant challenges. Often, the high density of pulse streams gives rise to pulse overlapping, impeding the detection of certain pulses by the receiver and ultimately causing pulse loss within the received pulse train. Moreover, spurious pulses, resulting from erroneous detections due to pulse splitting and overlapping, present a persistent challenge. Furthermore, measurement errors, stemming from the inherent imprecision of sensors in receivers and typically following a Gaussian distribution, introduce further imperfections during the sorting process. Consequently, deinterleaving in practical electronic warfare scenarios becomes exceptionally challenging due to these three types of noise, as they disrupt the patterns of PDWs and disturb the regularity of the intercepted pulse flow.

It is important to highlight that deinterleaving is performed on pulse sequences detected by the receiver. Consequently, the

effects of various factors that impair receiver pulse detection performance, such as Signal-to-Noise Ratio (SNR) or Signal-to-Interference-plus-Noise Ratio (SINR), pulse overlapping, and pulse splitting, are inherently manifested in pulse loss and spurious pulses. Therefore, this study does not engage in further modeling or analysis of these factors.

Moreover, to the best of our knowledge, existing published research primarily simulates pulse streams as continuous sequences without significant intervals. These studies typically focus on measurement errors, pulse loss, and spurious pulses, either individually or in combination. However, in actual reconnaissance scenarios, radars are constrained by the limited width of their main lobes. To detect or search within a specific range, mechanical or electronic scanning is often employed to cover all relevant directions. As a result, the signals intercepted by the receiver only encompass pulse sequences during the illumination of the enemy radar's main lobe. Consequently, the received signals exhibit periodic changes corresponding to the scan cycle of the enemy radar, manifesting as segments of discontinuous yet periodic pulse sequences within the pulse stream, as illustrated in Fig. 4.

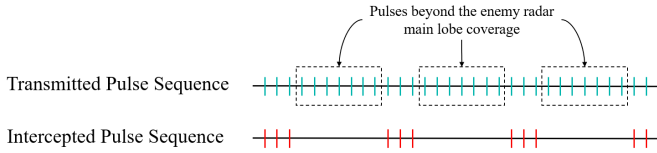


Fig. 4. Discontinuous but periodic intercepted sequences.

To address this gap, this study introduces a novel non-ideal condition, while retaining the aforementioned three types of non-ideality: the intercepted pulse sequence undergoes significant and continuous pulse loss over a duration much longer than its PRI – a phenomenon we henceforth term signal sparsity, due to the resulting sparsity of pulses within the sequence, as illustrated in Fig. 4. This condition further complicates the already challenging task of signal processing in electronic warfare scenarios.

D. MFR Effects

Multi-function radar represents a pivotal advancement in modern radar technology [31]. In contrast to conventional radars that operate within a single working mode, MFRs demonstrate remarkable versatility by switching among multiple operational modes tailored to diverse mission requirements. These radars encompass a wide range of capabilities, including target search, tracking, moving target indication, missile guidance, and target imaging.

MFRs are equipped with phased array antennas, enabling precise directional control over the radiated signal energy. Ideally, the radiation pattern of such antennas boasts a narrow beam with high gain and minimal sidelobes. The main lobe width typically spans only 1-2 degrees, and the gain disparity between the main lobe and sidelobes can attain levels of 30-40 dB.

While signal sorting for conventional radars is primarily influenced by factors such as pulse loss, spurious pulses, and

measurement errors, the signal sorting process for MFRs is significantly more complex. On one hand, to achieve multiple objectives like target search and tracking, MFRs emit specially structured pulse groups in various spatial directions, leveraging agile parameters to enhance anti-reconnaissance and anti-jamming capabilities. These parameters include frequency agility and PRI agility, among others. In the presence of pulse loss and spurious pulses, successful signal sorting requires a deeper understanding of the temporal relationships between pulse groups corresponding to different functions, enabling more effective separation from other radar signals.

On the other hand, the strong directionality and beam agility of MFRs significantly reduce the probability of signal interception by traditional electronic reconnaissance systems. Unlike conventional radars, which are primarily affected by sporadic pulse loss, airborne radar-based reconnaissance systems often encounter the loss of entire pulse groups when intercepting MFR signals. This results in a substantial decrease in the temporal regularity between pulse groups, posing significant challenges for subsequent signal sorting and the analysis of MFRs' diverse operational modes.

III. THE PROPOSED METHOD

A. Overall Structure

The architecture of the proposed end-to-end framework is depicted in Fig. 5. Initially, the PDW sequence P undergoes a preprocessing stage, where typically two options are available for refinement. Following preprocessing, the sequence is fed into the TCAN sorting network.

TCAN is an acronym representing a neural network architecture that primarily comprises several cascaded TCN Residual Blocks (Res Block, indicated in blue), a Self-Attention Block (SA Block, highlighted in lavender), and a Fully-Connected layer (FC layer, denoted in green).

The TCAN network processes the interleaved PDW sequence in an end-to-end manner. This processing involves the sequential application of the TCN Residual Blocks, which leverage dilated convolutions to capture temporal dependencies at various scales. The Self-Attention Block further enhances the model's ability to focus on relevant parts of the input sequence, thereby improving its capacity to discern between different signal types and their respective features. Finally, the FC layer maps the processed sequence to the predicted label sequence Y , which corresponds to the identified operational modes and signal characteristics of the MFR.

B. Data Preprocessing

In addition to the DTOA representation discussed in Section II-A [24], [25], [28], the binary sequence representation constitutes another prevalent input format in the relevant research domain. As illustrated in Fig. 2, this approach involves digitizing the TOA by assigning each pulse to a specific time unit based on its arrival time [23], [27], [31]. This digitization process is analogous to sampling in digital signal processing, resulting in

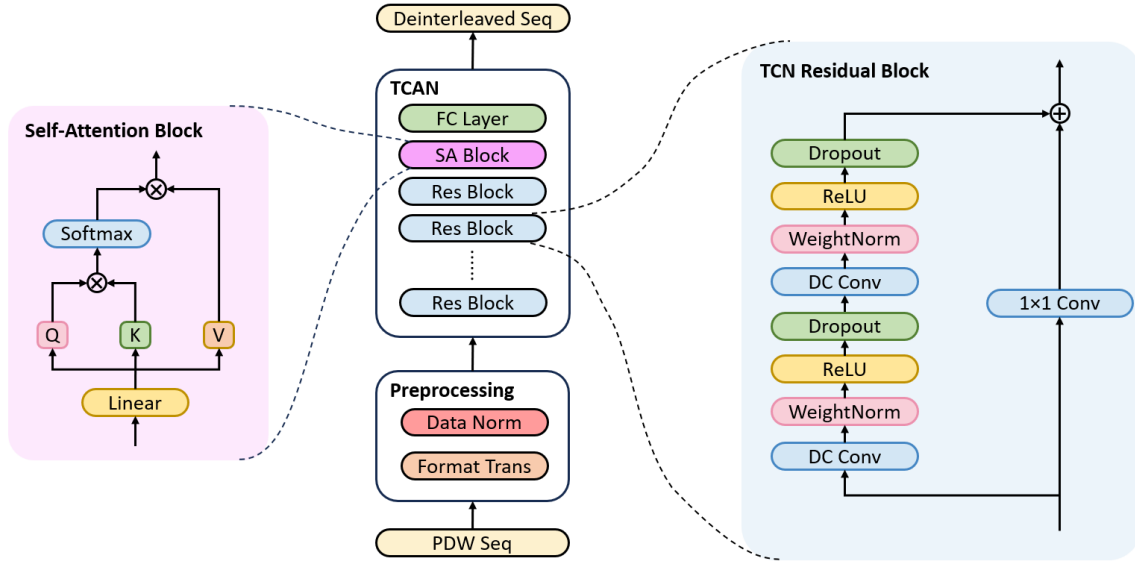


Fig. 5. The structure of proposed framework.

a binary sequence where the PRI pattern is encoded through the positions of 0s and 1s. The input x_i is defined as:

$$x_i = \begin{cases} 1, & S \cap [(i-1)T_s, iT_s] \neq \emptyset \\ 0, & S \cap [(i-1)T_s, iT_s] = \emptyset \end{cases} \quad (10)$$

where T_s denotes the sampling interval, $[(i-1)T_s, iT_s]$ signifies the i -th time unit, and $S = \{\text{TOA}_k\}$ is the set of TOA. The other PDW parameters do not undergo similar processing; instead, they are normalized, as delineated subsequently, and appended to the respective TOA positions. For conciseness, the mathematical formulation of x_i in the multi-parameter case is omitted. Furthermore, owing to the inherent flexibility of deep learning models in handling input scalability, the network's input is not restricted by the number of available parameters, as is the case with traditional methods. This adaptability enables us to utilize a unified framework to assess the deinterleaving performance of both TOA-based and multi-parameter-based approaches. The conversion to both formats is seamlessly integrated within the format transformation module (Format Trans in Fig. 5, indicated in orange).

The binarization of TOA, achieved by encoding TOA intervals through the strategic placement of 0s and 1s, serves to enhance the model's capability to discern PRI patterns. This enhancement, in turn, elevates deinterleaving performance in scenarios where only TOA data is available. A comprehensive exploration of this advantage will be provided in Section IV-B. This binary encoding scheme also simplifies the model by reducing its dimensionality and parameter count, thereby facilitating faster convergence. However, to ensure that adjacent pulses do not fall into the same time unit, a smaller sampling interval is often required. This choice results in a significant increase in sequence length and computation time per epoch. Conversely, the DTOA input format maintains the original sequence length, leading to faster computation times per epoch during training. Despite this advantage, the higher information density of DTOA input necessitates a network with higher

dimensionality and more parameters to comprehend the input data. Consequently, the number of epochs required for convergence dramatically increases when using DTOA input compared to binary input.

Furthermore, to address the challenges posed by substantial disparities in the magnitudes of parameters, such as RF typically measured in MHz and PW expressed in μs , normalization emerges as a pivotal step in augmenting both the convergence rate and the overall sorting efficacy of the model. The normalization procedure for each parameter within the interleaved sequence is executed using the formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (11)$$

where x represents the vector form of the input x_i . This normalization strategy guarantees that all parameters are confined within the unified range of $[0, 1]$, thereby enabling the model to proficiently extract pertinent features. This normalization is executed within the data normalization module (Data Norm in Fig. 5, highlighted in red).

C. Details in Proposed Framework

1) *TCN Residual Block*: The foundational element of our proposed framework is the TCN residual block, which plays a pivotal role in the extraction and fusion of features from the original input sequence [29]. Each TCN residual block is composed of two identical sub-structures, seamlessly integrating dilated causal convolution, weight normalization, a ReLU activation function, and a dropout layer. This architectural design is particularly adept at modeling long-range dependencies within the input sequence with high efficiency.

Central to these sub-structures is the concept of dilated causal convolution, which combines the attributes of causal convolution and dilated convolution. Causal convolution ensures that the output at any time step t is solely dependent on

inputs that occur at or precede t , thereby preserving causality. The formula for a standard causal convolution is given by:

$$y_t = \sum_{i=0}^{k-1} w_i \cdot x_{t-i} \quad (12)$$

where y_t represents the output, w_i denotes the kernel weights, x_t signifies the input sequence values, and k is the kernel size.

The dilation rate, denoted by d , controls the spacing between kernel elements, thereby effectively enlarging the receptive field to forge wider temporal linkages among each pulse without necessitating deeper network layers. The output of a dilated causal convolution is mathematically expressed as:

$$y_t = \sum_{i=0}^{k-1} w_i \cdot x_{t-d \cdot i} \quad (13)$$

Across all layers, the convolution kernel size remains constant, while the dilation factor increases exponentially with the depth of the network: $d_l = 2^{l-1}$, where l represents the network level. For instance, as illustrated in Fig. 5, d_l is initialized to 1 at the first layer (equivalent to standard convolutions) and progressively increases with each subsequent layer, ultimately reaching a value of 128 at the final hidden layer. This pyramidal structure significantly enhances the receptive field of the TCAN, enabling a more thorough examination of individual pulses and their interactions with neighboring pulses. By establishing connections to distant contextual information, this structure enhances sorting performance and effectively alleviates the challenges presented by non-ideal factors, such as pulse loss, spurious pulses, and signal sparsity.

Subsequent to the dilated convolution layer, weight normalization is employed to stabilize the learning process. The ReLU activation function introduces non-linearity, allowing the model to learn complex patterns, particularly in pulse streams influenced by non-ideal factors. Dropout is applied to regularize the network and reduce overfitting by randomly setting a fraction of the layer outputs to zero during training.

Furthermore, to mitigate the issues of vanishing and exploding gradients in deep networks and to accelerate convergence, each TCN residual block incorporates residual connections, which are integral to their design. When the dimensions of the input and output sequences do not match, the residual connection is typically replaced with a 1×1 convolution to ensure compatibility. This approach facilitates a unified framework capable of handling both TOA-based and multi-parameter-based inputs. This design not only enhances the stability and performance of the network but also contributes to its robustness and generalization capabilities. The output of a TCN residual block can be expressed as:

$$H(x) = F(x) + \text{conv}(x) \quad (14)$$

where $F(x)$ represents the function learned by the block, and $\text{conv}(x)$ denotes the 1×1 convolution used for dimensional matching when necessary.

2) *Self-Attention Block*: In the field of deep learning, particularly in addressing sequential data and natural language processing tasks, self-attention mechanisms have emerged as a cornerstone technique [30]. This research endeavor extends

the application of these mechanisms to the realm of signal sorting, where they play a pivotal role in enhancing the model's proficiency in discerning essential temporal features. Following a series of residual blocks, which are essential for retaining crucial information as it propagates through the network, a self-attention block is strategically deployed to dynamically calibrate the significance of each element within the overall context.

The core of this self-attention mechanism lies in its ability to assess similarities across different positions within a sequence, thereby assigning varied weights to distinct time steps. In the signal sorting task, this translates to allocating appropriate weights to different pulses based on the similarity of their extracted features. This is achieved through a process wherein the query Q , key K , and value V matrices are all derived from the output of the preceding residual block $X = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^{n \times d_x}$ via linear transformations, each $x_i \in \mathbb{R}^{d_x}$ being a feature vector. Mathematically, this process can be expressed as:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V \quad (15)$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d_x \times d_K}$ represent learnable weight matrices that facilitate the derivation of these matrices from X , $Q, K, V \in \mathbb{R}^{n \times d_K}$ are the resulting matrices of queries, keys, and values, and d_K denotes the dimension of the queries, keys, and values.

By utilizing these transformed matrices, the self-attention block effectively captures global dependencies in pulse streams based on the local features extracted by the residual blocks. This is achieved through the computation of attention weights, which hinge on the similarity between the query and key matrices. Initially, the attention score matrix $S \in \mathbb{R}^{n \times n}$ is computed by taking the dot product of Q and K^T , scaled by $\sqrt{d_K}$ to mitigate the risk of excessively large values:

$$S = \frac{QK^T}{\sqrt{d_K}} \quad (16)$$

Subsequently, S undergoes normalization via the softmax function, which translates the attention scores into a probability distribution:

$$\mathbf{A} = \text{softmax}(S) \quad (17)$$

$$A_{ij} = \frac{\exp(S_{ij})}{\sum_{k=1}^n \exp(S_{ik})} \quad (18)$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the attention weight matrix. The softmax ensures that each row of \mathbf{A} sums to 1, reflecting the relative importance of each key in contributing to the corresponding query.

The resulting attention weights are then employed to compute the final output of the self-attention block as a weighted aggregation of the value vectors. This can be succinctly expressed in matrix notation:

$$\mathbf{Z} = \mathbf{A}\mathbf{V} \quad (19)$$

where $\mathbf{Z} \in \mathbb{R}^{n \times d_K}$ is the output matrix. Each row of \mathbf{Z} constitutes a contextual representation, encapsulating pertinent information from all positions, weighted by their respective importance.

By integrating the self-attention mechanism, the model gains the ability to recognize common characteristics among pulses belonging to the same category within the features extracted by the TCN residual block. Prior to the final classification stage, it assigns greater weights to these identified pulses, thereby enhancing their probability of being accurately categorized. This capability notably alleviates the challenges presented by pulse loss and signal sparsity, where isolated pulses face difficulties in establishing meaningful connections owing to the absence of similar neighboring pulses. Furthermore, the similarity-based reweighting scheme employed by the self-attention mechanism effectively prevents pulses from establishing dependencies with low-similarity spurious pulses, thus enhancing the model's robustness against spurious signals.

Following this, the output from the self-attention block is funneled through a fully connected layer, which maps the extracted features to the corresponding label sequence. This sequence-to-sequence mapping is indispensable for the precise sorting and interpretation of signals within the given context.

D. Loss Function

In the training phase of our model, we employ focal loss as the loss function to address class imbalance in classification tasks, particularly in scenarios characterized by a significant disparity between the number of easy-to-classify examples (i.e., 0s in binary input) and harder examples (i.e., 1s in binary input). The standard cross-entropy loss treats all examples equally when computing the overall loss, which can be expressed as:

$$L_{CE} = - \sum_{i=1}^C y_i \log(p_i) \quad (20)$$

where C is the number of classes, y_i is the true label for class i , and p_i is the predicted probability that the sample belongs to class i . However, the abundance of irrelevant 0s in binary input can disrupt the balance of the dataset, leading to biased model training. To mitigate this issue, focal loss is designed to down-weight the loss contribution of easy examples, thereby enabling the model to focus more on challenging ones. It extends the standard binary cross-entropy loss and is formulated as:

$$L_{Focal} = - \sum_{i=1}^C \alpha_i (1 - p_i)^\gamma y_i \log(p_i) \quad (21)$$

where α_i is the weighting factor for class i , which helps balance the contribution of each class. Specifically, under-represented or minority classes can be assigned a larger α_i to ensure they are not overlooked due to their lower frequency. Additionally, focal loss incorporates a modulating factor $((1 - p_i)^\gamma)$ to reduce the loss contribution from well-classified examples, thereby directing the model's learning towards more difficult examples. The parameter γ controls the strength of this modulating factor, allowing for fine-tuning of the model's focus on harder examples. By carefully selecting appropriate hyperparameters for the focal loss, our model can effectively capture temporal information from the

training sequence while mitigating the adverse effects of data imbalance.

IV. EXPERIMENTS

To thoroughly assess the efficacy of the proposed deinterleaving method and demonstrate its robustness under non-ideal conditions, a comprehensive series of experiments were carried out in both TOA-only and multi-parameter scenarios. Section IV-A elaborates on the parameters utilized in generating the simulated datasets, the evaluation metrics employed to quantify the performance of the sorting process, the baseline methods selected for comparative analysis, and the overall experimental setup. The experimental results, along with their corresponding analysis, are presented in Sections IV-B through IV-E. These sections provide a detailed examination of the proposed method's performance across various conditions, highlighting its superior capabilities in accurately deinterleaving signals even when confronted with challenging non-ideal circumstances. Through this rigorous experimental validation, the study aims to establish the proposed deinterleaving method as a reliable and effective solution for signal sorting tasks.

A. Experimental Design

1) *Dataset*: Two distinct categories of datasets, encompassing both conventional radar signals and MFR signals, were created to assess the performance of various algorithms under diverse scenarios. The PDW data for both conventional radars and MFRs were simulated using the parameters detailed in Tables I and II, respectively.

Table I presents the parameters of the simulated conventional radar emitters, detailing PRI modulation types, PRI values, PW, RF, and DOA, as referenced in [32]. Each radar corresponds to a specific PRI modulation mode, including fixed PRI (540 μ s), jitter PRI (370 μ s, $J=0.25$, where J represents the PRI jitter rate), stagger PRI (110, 280, 800 μ s, which appear in sequence and repeat cyclically), and dwell & switch PRI (400 μ s \times 20, 600 μ s \times 15, 900 μ s \times 10, where each PRI corresponds to pulse groups of lengths 20, 15, and 10, respectively). Meanwhile, each radar operates within a specified range for PW, RF, and DOA, with these parameters varying randomly within their respective limits. Furthermore, Table II provides an in-depth overview of the parameters for MFR emitters. Unlike conventional radars, MFRs operate in multiple modes, each employing distinct PRI modulation techniques to support various tasks such as target search and tracking. Each MFR emitter in this dataset are operating in three modes, including fixed PRI, jitter PRI, stagger PRI, and dwell & switch PRI. As the operating mode shifts, the ranges of PW and RF adjust accordingly, while DOA, being a directional factor, remains constant.

The data of four radar emitters generated from each table's parameters were individually merged into a unified, interleaved pulse stream arranged in chronological order, spanning a duration of 20 seconds. This process culminated in the formation of the final datasets. Each dataset was subsequently partitioned into training and testing subsets, maintaining a ratio of 4 : 1, respectively. It is important to highlight that

the parameters were deliberately set with a certain degree of overlap to more accurately emulate the complexities of a realistic electronic countermeasure environment. Fig. 6(a) depicts a 100 ms segment from the testing set, where different colors denote different radar categories, encompassing the four PRI modulation types specified in Table I. In contrast, Fig. 6(b) showcases the deinterleaving results of a 10 ms testing sample processed by our proposed TCAN framework.

For TOA-based algorithms, the training process is conducted using exclusively TOA data and their corresponding labels extracted from the training set. In contrast, multi-parameter-based algorithms utilize the complete set of PDW parameters in conjunction with the labels for training purposes. Furthermore, the entire training procedure is executed on interleaved pulse sequences.

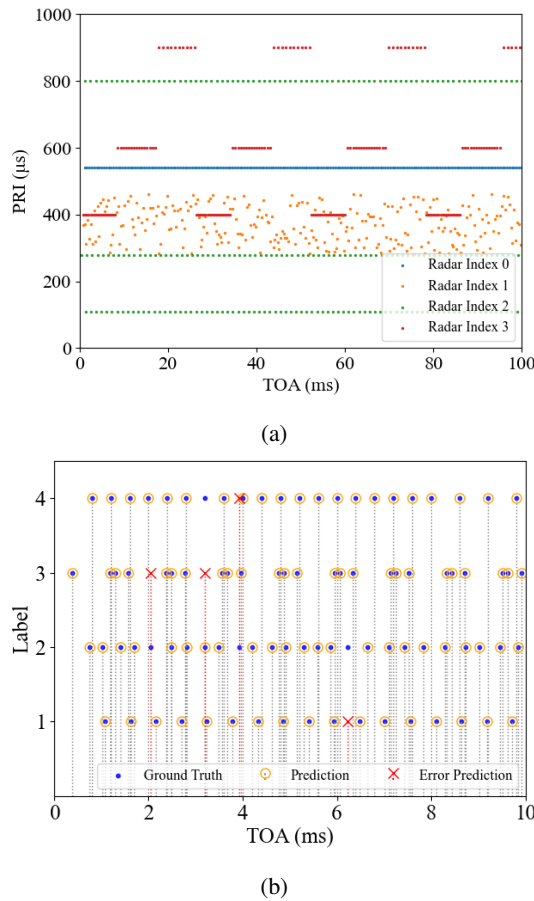


Fig. 6. Visualization of our dataset. (a) A segment from our dataset. (b) The sorting results of the proposed framework.

2) *Baseline Methods For Comparison:* To ensure a comprehensive evaluation of our proposed approach, we have selected several baseline methods for comparison. These methods encompass both well-established traditional techniques and emerging deep learning-based approaches, enabling us to conduct a thorough and impartial assessment of the relative performance of our model. This comparison aims to highlight any improvements achieved by our proposed method.

- 1) Long Short-Term Memory (LSTM): As a widely-adopted framework for processing time series data,

LSTM has naturally become a cornerstone in mainstream research for radar signal sorting [24]. Leveraging memory cells and gating mechanisms—specifically, input, forget, and output gates—LSTM scrupulously regulates the flow of information over time. This unique structure enables LSTM to retain pertinent information throughout extensive pulse sequences, rendering it particularly suitable for separating mixed radar signals.

- 2) Gated Recurrent Unit (GRU): Another prominent model designed for time series data, GRU (Gated Recurrent Unit) offers computational simplicity compared to LSTM while achieving comparable performance [25]. GRU combines the input and forget gates into a single update gate and eliminates the output gate, effectively managing long-term dependencies between pulses while reducing computational overhead in comparison to LSTM.
- 3) TCN: The original version of the proposed TCAN, TCN is a model entirely constructed using convolutions for sequence modeling, including signal sorting [29]. Unlike LSTM and GRU, which rely on sequential processing, TCN utilizes causal convolutions to capture long-range dependencies in pulse flow without necessitating recurrence. Additionally, TCN employs dilated convolutions and residual connections, enabling it to attain large receptive fields and maintain stable gradients. These characteristics render TCN highly effective for signal sorting, as it can efficiently handle long and complex pulse sequences with superior parallelization capabilities.
- 4) CDIF: As the most fundamental histogram-based approach for deinterleaving radar signals [5]. CDIF calculates the time differences between pulse arrivals step-by-step and generates a histogram of these time differences, thereby obtaining potential PRIs by setting a threshold on the histogram. Subsequently, it searches for pulses corresponding to the potential PRIs one-by-one across the pulse sequence.
- 5) SDIF: An enhanced variant of CDIF [6], SDIF optimizes the threshold design to better align with real-world conditions and, in contrast to the cumulative multi-level difference histogram employed by CDIF, directly detects a specific-order histogram, thereby improving computational efficiency.
- 6) PRI Transform: Another improved histogram-based method, the PRI Transform addresses the issue of subharmonics commonly introduced by methods like CDIF and SDIF [7]. By adding a phase factor to the autocorrelation function, the PRI Transform algorithm suppresses these subharmonics, retaining the true PRI while eliminating higher-order harmonics. However, this method comes with a notable drawback: its computational complexity is significantly increased due to the need for integral calculations similar to autocorrelation.

It is crucial to emphasize that among the baseline methods, deep learning-based approaches necessitate input adaptation to conform to the network's requirements and, consequently,

TABLE I
Parameters of Simulated Conventional Radar Emitters

Radar Index	PRI Modulation	PRI Value/ μ s	PW/ μ s	RF/GHz	DOA/ $^\circ$
1	Fixed	540	20–40	8.1–8.5	35–37
2	Jitter	370, J=0.25	30–50	8.15–8.3	36–39
3	Stagger	110, 280, 800	10–40	8.4–8.6	34–38
4	Dwell&Switch	400 \times 20, 600 \times 15, 900 \times 10	25–45	8.25–8.6	36–37

TABLE II
Parameters of Simulated Multi-Function Radar Emitters

Radar Index	Working Mode	PRI Modulation	PRI Value/ μ s	PW/ μ s	RF/GHz	DOA/ $^\circ$
1	Mode 1	Jitter	420, J=0.2	20–40	8.1–8.5	35–37
	Mode 2	Stagger	190, 240, 380	30–50	8.2–8.4	
	Mode 3	Dwell&Switch	290 \times 20, 350 \times 14, 180 \times 10	20–40	8.6–8.8	
2	Mode 1	Fixed	260	10–20	8.15–8.3	36–39
	Mode 2	Stagger	370, 410, 230	20–40	8.3–8.6	
	Mode 3	Jitter	330, J=0.1	30–50	8.2–8.4	
3	Mode 1	Fixed	320	30–40	8.4–8.6	34–38
	Mode 2	Stagger	460, 140	10–20	8.2–8.4	
	Mode 3	Dwell&Switch	280 \times 10, 360 \times 8, 80 \times 24, 190 \times 12	10–40	8.5–8.8	
4	Mode 1	Dwell&Switch	440 \times 15, 490 \times 20	25–45	8.1–8.3	36–37
	Mode 2	Fixed	390	30–50	8.2–8.5	
	Mode 3	Jitter	150, J=0.1	20–40	8.5–8.8	

employ the preprocessing module discussed in Section III-B. Conversely, traditional methods can proceed directly with their algorithmic procedures for signal sorting, without the necessity to alter the input format of the pulse sequences.

3) *Evaluation Metric*: To facilitate a quantitative comparison between baseline algorithms and our proposed methodology, it is imperative to adopt a unified evaluation metric that assesses their respective performances. In this study, we select recall as the primary performance indicator for evaluating the diverse algorithms under consideration. Recall, which signifies the ratio of samples correctly predicted as positive to the total actual positive samples, is particularly relevant in this context as it reflects the proportion of pulses from a specific radar that are accurately categorized.

- 1) True Positive (TP): This refers to pulses that originate from a particular radar and are successfully deinterleaved.
- 2) False Negative (FN): This pertains to pulses that belong to a specific radar but are erroneously assigned to other radar categories.

The recall for each radar category can be formulated as follows:

$$\text{recall}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \quad (22)$$

Furthermore, to obtain a comprehensive measure of performance across all radar categories, we compute the average recall:

$$\text{recall}_{\text{average}} = \frac{1}{N} \sum_{i=1}^N \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \quad (23)$$

where N represents the total number of radar categories and i denotes the i -th radar category. It is important to highlight that,

unless otherwise specified, the recall presented in the charts of the subsequent experimental results refers to the average recall.

4) *Experimental Arrangements*: To comprehensively validate and evaluate the efficacy of the proposed method, a meticulously designed series of experiments are conducted across various scenarios. The ensuing sections delve into four distinct cases, each addressing critical aspects of the methodology's performance:

- 1) Performance Analysis of Input Format Influence: Section IV-B systematically investigates the impact of different input formats on the overall effectiveness of both the proposed method and baseline approaches. A comparative analysis is presented, elucidating the differential effects these formats exert on performance metrics.
- 2) Exploring the Impact of Signal Sparsity on Algorithmic Performance: In Section IV-C, we delve into the intricate relationship between signal sparsity and the performance of multiple algorithms. This exploration highlights the proposed algorithm's potential strengths and challenges in managing sparse data, offering valuable insights into its adaptability.
- 3) Assessing Performance Degradation in Non-Ideal Conditions: Section IV-D shifts focus to the robustness of the proposed method under three prevalent non-ideal conditions encountered by interception receivers. This analysis provides a nuanced understanding of how these adverse scenarios impact performance, thereby shedding light on the method's resilience.
- 4) Evaluating Sorting Performance on MFR Streams: Lastly, Section IV-E scrutinizes the proposed method's sorting capabilities when applied to streams originating

from MFRs. This assessment underscores the method's proficiency in handling such data, offering a comprehensive evaluation of its practical utility.

5) *Simulation Settings*: The procedures outlined in this paper were executed on a high-performance desktop workstation running Windows 10 and equipped with an RTX 3090 GPU (24GB VRAM) to ensure robust computational capabilities. All network models used in the deep learning methodologies were implemented using Python 3.11.4 (64-bit) and PyTorch 2.0.1, leveraging the versatility and efficiency of these frameworks.

In the subsequent experiments, the maximum batch size that can be supported varies according to the input format and model complexity. Specifically, for binary input, TCN and TCAN can accommodate a batch size of up to 512, whereas GRU and LSTM are constrained to a maximum of 128. When using DTOA input, the maximum batch size is limited to 32 owing to the increased memory requirements. During the actual training process, the batch size can be adjusted according to the available VRAM to suit different GPU configurations. Given that training is typically performed in advance on high-performance GPUs, with the trained parameters subsequently distributed to various electronic warfare computing platforms, the hardware configuration for practical deployment is primarily determined by the desired inference speed and cost constraints.

B. Performance Analysis of Input Format Influence

As detailed in Section III-B, we introduced two typical methodologies to transform the intercepted sequence into suitable inputs for various models. In this section, we conduct a comparative analysis of the deinterleaving performance of different input formats for pulse sequences. To ensure an equitable comparison, two sequences derived from the same 16-second pulse stream, generated using the parameters outlined in Table I, were input into distinct networks. This allowed us to verify the efficacy of both binary input and DTOA input.

For the given dataset, as depicted in Table III, the performance of binary input surpassed that of DTOA input for arbitrary model sorting based on TOA. This suggests that binary input, which represents TOA using intervals of 0 and 1, aids the network in recognizing potential PRIs. Furthermore, the sparsity of the data results in a limited training dataset, and the binary input can effectively extend the data length, proving advantageous for network training. Conversely, as illustrated in Table IV, for deinterleaving methods based on multiple parameters, the improvement with binary input is marginal, while DTOA input demonstrates substantial enhancement. The performance of TCN and TCAN with binary input remains superior. However, GRU and LSTM algorithms exhibit better performance with DTOA input compared to the binary format. This can be attributed to the fact that binary input directs the model's focus more towards patterns in the TOA dimension, effectively characterizing TOA features, but it also attenuates the exploration of information embedded in other dimensions. Consequently, in subsequent experiments, all TOA-based deinterleaving algorithms will adopt binary

input as the preprocessing step, while GRU and LSTM will transition to DTOA input for multi-parameter sorting tasks.

Fig. 7 depicts the recall of pulses for each class across various algorithms, considering both TOA-based and multi-parameter-based scenarios. A notable observation is the variation in recall rates among different classes, where Class 2 and Class 3 consistently exhibit higher recall values across all deep learning methods. This can be ascribed to their relatively larger proportion in the dataset, as compared to the other two classes. Although focal loss was employed to alleviate the data imbalance issue, performance disparities among classes still persist. In contrast, traditional methods demonstrate commendable performance in sorting Fixed PRI (Class 1) and Stagger PRI (Class 3), yet they encounter challenges in identifying Jitter PRI (Class 2), leading to a significant degradation in their overall sorting performance.

Additionally, it is noteworthy that even though radar signals may be indistinguishable in the dimensions of PW, RF, and DOA, incorporating these features still yields additional improvements in sorting. Therefore, given that PDW data is comprehensive, all other dimensions should be harnessed during model training and inference to maximize performance.

The training and inference time-related metrics are presented in Fig. 8 and Table V, encompassing the average training time per epoch, the number of epochs needed for convergence, the overall training duration, and inference time for various methods. As elaborated in Section III-B, DTOA input leads to a reduced execution time per epoch compared to binary input; nevertheless, it necessitates a substantially higher number of epochs to achieve convergence. Owing to its greater model complexity, TCAN exhibits the longest computation time per epoch for a given input format. In terms of total training time, although GRU with DTOA input is the most time-efficient, its recall performance lags behind that of TCAN with binary input. Regarding inference speed, the runtime of each method is directly proportional to its per-epoch computation time during training. DTOA input offers a marked advantage in inference efficiency over binary input, with both outpacing traditional methods by a significant margin.

TABLE III
Recall of Different Input Formats
(TOA-based)

Input Format	TCN	TCAN	BGRU	BLSTM
Binary	84.38%	92.99%	74.34%	75.31%
DTOA	75.32%	70.47%	57.37%	71.03%

TABLE IV
Recall of Different Input Formats
(Multi-parameter-based)

Input Format	TCN	TCAN	BGRU	BLSTM
Binary	84.53%	94.06%	72.79%	73.93%
DTOA	83.02%	84.75%	88.84%	88.55%

TABLE V
Computation Time and Convergence Performance of Different Input Formats and Algorithms

Method	Avg. Execution Time Per Epoch (s)	Avg. Convergence Epochs	Avg. Time Consumed for Training (s)
TCN DTOA	0.0546	9750	532.35
TCAN DTOA	0.0646	2700	174.42
GRU DTOA	0.0326	1600	52.16
LSTM DTOA	0.0514	1850	95.09
TCN Binary	0.3050	100	30.50
TCAN Binary	0.8534	350	298.34
GRU Binary	0.6228	575	358.11
LSTM Binary	0.6572	650	427.18

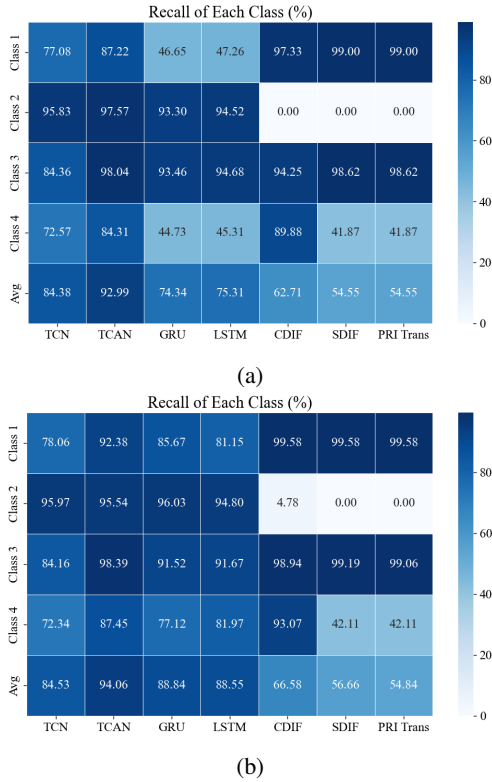


Fig. 7. Deinterleaving performance of each class. (a) TOA-based algorithms. (b) Multi-parameter-based algorithms.

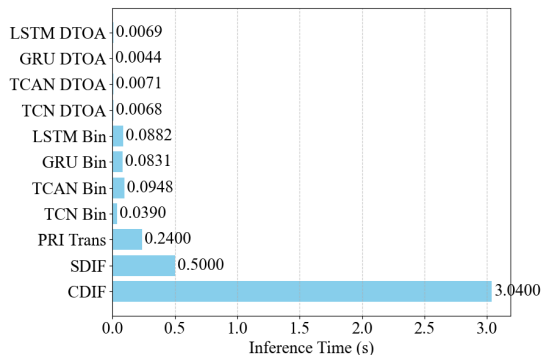


Fig. 8. Inference time of different algorithms.

C. Impact of Signal Sparsity on Algorithmic Performance

In this section, we investigate the influence of signal sparsity, induced by the scan cycle in real radar systems, on deinterleaving performance. To this end, we prepared intercepted sequences characterized by three distinct levels of sparsity to assess the impact on various algorithms. Here, x in Fig. 9 denotes the ratio of signal length to interval length within a scan period (i.e., 1:3, 1:5, and 1:8).

As depicted in Fig. 9, it is evident that the performance trend of deep learning algorithms, across datasets with varying sparsity levels, is closely associated with the input format. Specifically, with binary input, performance tends to improve as sparsity increases. For instance, TCAN (Binary) attains a recall rate of approximately 92% at 1:8 sparsity, compared to 89% at 1:3 sparsity. This can be attributed to the sampling-like preprocessing, which transforms the fixed-duration pulse sequence into a fixed-length 0-1 sequence sampled at regular intervals. As the pulse train becomes sparser, the density of 1s decreases, facilitating easier pattern identification by the model.

Conversely, the performance with DTOA input improves as the dataset becomes denser. For example, BGRU (DTOA) exhibits an increase in recall from 80% at 1:8 sparsity to 86% at 1:3 sparsity. This enhancement is due to the sequence length for DTOA input increasing with signal density, providing more training samples and leading to a more effective model. Traditional methods, such as CDIF and SDIF, which rely on histogram statistics for sorting, remain relatively unaffected by signal sparsity, with recall values consistently hovering around 70% – 75% across different sparsity levels. Notably, their sorting performance is still inferior to that of deep learning methods, regardless of sparsity changes, highlighting the efficacy of modern machine learning approaches in radar signal deinterleaving.

Furthermore, TCAN (Binary) consistently outperforms other methods across all sparsity levels, maintaining a recall rate above 90%. On the other hand, traditional methods like PRI Transform exhibit the lowest recall, remaining below 60%, which underscores their limitations in effectively handling sparse radar pulse sequences compared to advanced deep learning techniques.

D. Performance Degradation in Non-Ideal Conditions

In this section, we assess the sorting capabilities of the algorithms under three typical non-ideal scenarios: measurement

TABLE VI
Parameter Setting for Combination of Three Types of Non-ideality

Case	Measurement Error STD [TOA (μ s), PW (μ s), RF (GHz), DOA ($^\circ$)]	Pulse Loss Ratio (%)	Spurious Pulse Ratio (%)
0	[0, 0.0, 0, 0.0]	0	0
1	[2, 0.2, 10, 0.2]	5	5
2	[4, 0.4, 20, 0.4]	10	10
3	[6, 0.6, 30, 0.6]	15	15
4	[8, 0.8, 40, 0.8]	20	20
5	[10, 1.0, 50, 1.0]	25	25

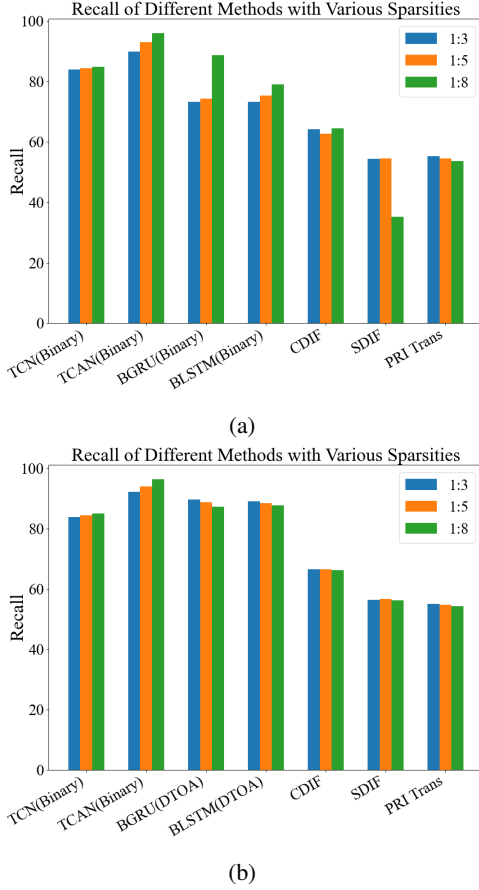


Fig. 9. Deinterleaving performance against varying levels of sparsity. (a) TOA-based algorithms. (b) Multi-parameter-based algorithms.

error, pulse loss, and spurious pulses. Measurement error is simulated by introducing Gaussian noise with zero mean and a standard deviation (STD) that incrementally rises from 0 to 10 μ s in steps of 2 μ s to the PDW data. Both pulse loss ratio and spurious pulse ratio are varied from 0 to 25% in increments of 5%. The pulse loss dataset is generated by randomly removing a specified percentage of pulses from the original pulse stream. For the dataset containing spurious pulses, we first generate a specified number of spurious pulses based on the predefined spurious pulse rate, which are then randomly inserted into the original pulse stream. Specifically, in the multi-parameter scenario, the PW, RF, and DOA parameters of

the spurious pulses are either randomly assigned or duplicated from nearby pulses, to mimic spurious pulses arising from receiver detection errors and pulse splitting.

From a comprehensive viewpoint, deep learning-based techniques exhibit notable superiority over traditional methods in signal sorting when constrained to using only TOA data, especially within increasingly complex electromagnetic environments, as illustrated in Fig. 10. Conversely, when full PDW information is available, the conventional two-step deinterleaving paradigm, consisting of pre-sorting followed by primary sorting, proves less effective than end-to-end deep learning frameworks that utilize the entire PDW data holistically, as depicted in Fig. 11.

As the intensity of non-ideal conditions, encompassing these three distinct types, escalates, all deep learning approaches undergo a significant decline in performance. Despite these adverse circumstances, the proposed TCAN substantially outperforms other methods when TOA data alone is utilized for deinterleaving, as shown in Fig. 10. Specifically, TCAN achieves an approximate 10% improvement in recall over the TCN and roughly a 15% improvement over GRU and LSTM networks. This advantage is primarily attributed to the attention mechanism, which reweighs the significance of different time steps, enabling the network to more effectively capture temporal similarities across adjacent time steps and facilitating the identification of signals from the same radar emitter.

In the context of deinterleaving using multiple parameters, TCAN continues to demonstrate robust performance. However, Figs. 11(a) and 11(b) reveal that, in terms of performance degradation due to measurement error and pulse loss, the TCAN model with Binary input experiences more pronounced degradation compared to GRU and LSTM models with DTOA input. At high noise levels, TCAN's performance is eventually surpassed by GRU and LSTM algorithms. Nonetheless, TCAN with Binary input is less susceptible to the impact of spurious pulses compared to GRU and LSTM with DTOA input, consistently maintaining a 5% to 7% performance edge, as shown in Fig. 11(c).

An analysis of the performance curves for all algorithms using only TOA with Binary input reveals a similar rate of decline for TCN, TCAN, GRU, and LSTM, suggesting that this phenomenon is intrinsically linked to the characteristics of the input representation. This observation implies that different input formats possess distinct advantages in addressing various

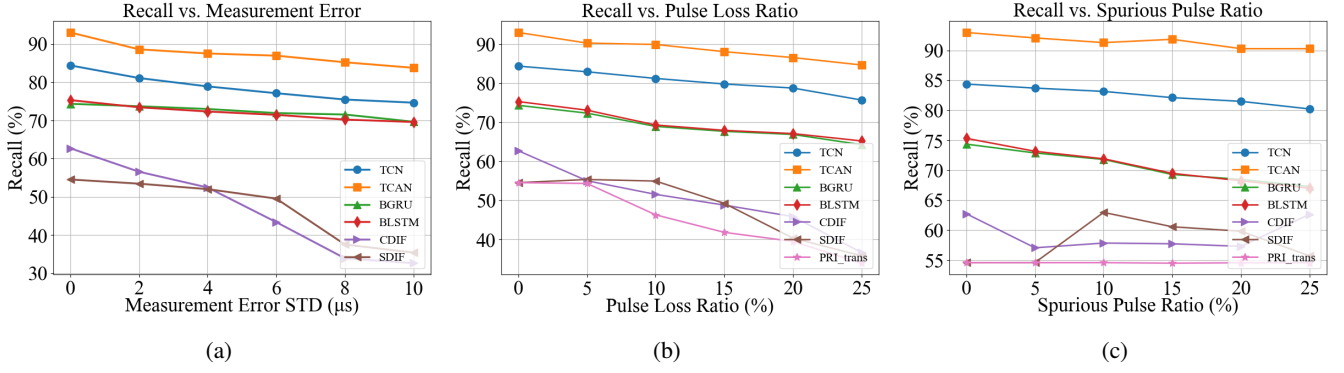


Fig. 10. Deinterleaving performance of TOA-based algorithms against three non-ideal conditions. (a) Measurement error. (b) Pulse loss. (c) Spurious pulses.

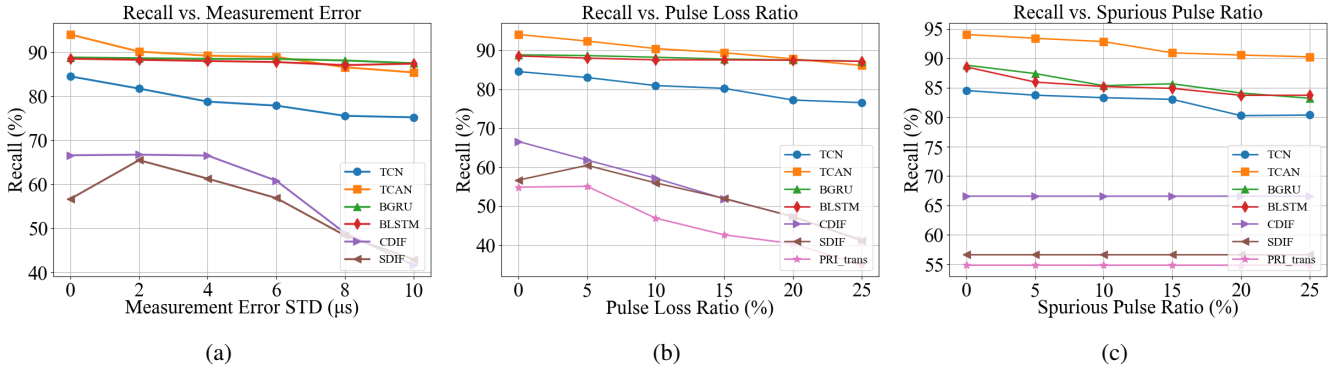


Fig. 11. Deinterleaving performance of multi-parameter-based algorithms against three non-ideal conditions. (a) Measurement error. (b) Pulse loss. (c) Spurious pulses.

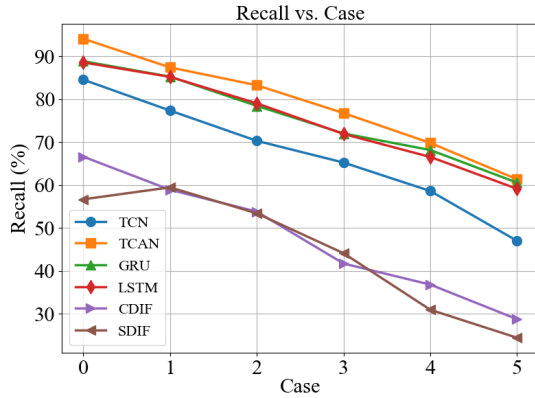


Fig. 12. Deinterleaving performance of multi-parameter-based algorithms under a combination of three non-ideal conditions.

non-ideal conditions. Binary input, which reconstructs sequence data in a sampling-like manner, is particularly sensitive to measurement errors, which can cause signals to deviate from their intended time windows. For instance, a slight timing error in the recording of a radar pulse may result in its assignment to an incorrect time unit, leading to misalignment of the input sequence. Additionally, pulse loss can cause signals to be absent from their original time windows, further complicating the sorting process and reducing recall. This severely impacts

algorithms that rely on the distribution of binary values for signal identification. In contrast, DTOA input, which does not depend on relative position for pulse pattern recognition, is less sensitive to the impacts of measurement errors and pulse losses, thereby preserving the network's ability to determine the relationships between different signal components.

Furthermore, we assessed the impact of combining all three non-ideal conditions within a single scenario. The specific settings for each case, which introduce varying levels of these three noise types, are detailed in Table VI. The sorting performance of different algorithms under these challenging conditions is depicted in Fig. 12. As anticipated, the concurrent presence of multiple data challenges results in a more pronounced decline in model performance compared to individual non-ideal conditions. Notwithstanding this, the TCAN algorithm consistently maintains a performance advantage over baseline methods in these complex scenarios. However, this advantage diminishes gradually, from an initial 5% margin to merely 1% in the final case.

It is worth noting that, as shown in the supplementary experimental results in Fig. 7, traditional methods are largely ineffective in sorting PRI jitter signals (Class 2), leading to poor deinterleaving performance. Moreover, since we utilized the original versions of these traditional methods, which were not specifically designed to address various non-ideal conditions, a direct comparison of their relative strengths is of

limited significance. Instead, their results are presented here solely as a reference for performance evaluation in complex environments.

E. Sorting Performance on MFR Streams

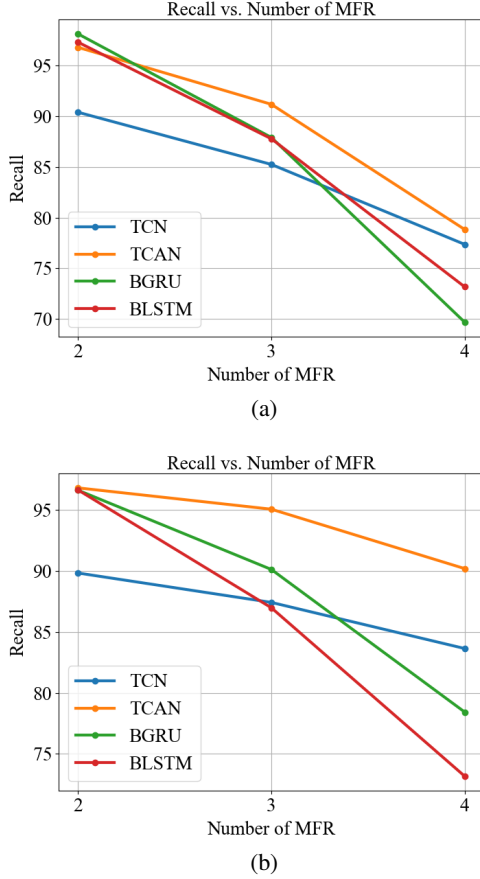


Fig. 13. Performance on deinterleaving MFR streams by (a) TOA-based algorithms; (b) multi-parameter-based algorithms.

In the concluding segment of our analysis, we shift our focus to assessing the deinterleaving effectiveness of an array of algorithms across datasets that encompass a varying number of MFRs. Unlike conventional radars, which operate within a single, predetermined mode, MFRs exhibit operational adaptability by dynamically transitioning between multiple modes according to mission-specific requirements. This adaptability introduces substantial challenges to the task of signal sorting. Traditional deinterleaving methodologies frequently fall short in distinguishing between the different operational modes of a single radar signal or differentiating distinct signals when prior information is lacking. Specifically, conventional algorithms may encounter difficulties in recognizing subtle variations in signal parameters as the same radar toggles between modes, resulting in frequent misclassifications, particularly in scenarios involving overlapping signals devoid of contextual knowledge.

In contrast, deep learning methodologies necessitate extensive training datasets to capture and learn the intricate patterns

embedded within these sophisticated signals. Consequently, all algorithms employed for deinterleaving MFR streams in our study are rooted in deep learning. The datasets used in our experiments consist of 2 to 4 MFRs, each of which operates in three distinct modes that switch stochastically, as detailed in Table II.

The results of our deep learning-based deinterleaving methods, one utilizing TOA features and the other leveraging multi-parameter features, are presented in Fig. 13(a) and 13(b). As the number of MFR sources escalates, both the TOA-based and multi-parameter-based algorithms exhibit a pronounced decline in sorting performance. This deterioration is primarily attributed to the heightened signal overlap and interference among multiple radar sources, which exacerbates the difficulty for the algorithms to accurately distinguish individual signals. Notwithstanding, the multi-parameter-based algorithm demonstrates a notable capacity to maintain a relatively high recall rate by integrating information from multiple feature dimensions. As data-driven algorithms, deep learning approaches capitalize on the diverse patterns present in multiple features, enabling the network to proficiently learn the characteristics of each operational mode of MFRs, thereby augmenting deinterleaving recall.

Furthermore, our proposed TCAN consistently outperforms other benchmark algorithms across datasets with an increasing number of radar sources, except in instances where the dataset comprises only two interleaved MFRs, where its performance is comparable to that of the benchmark approaches. It is worth noting that the TCAN algorithm exhibits minimal degradation in performance as the number of radar sources rises, with only a 5% drop in recall rate compared to the 15% – 20% decline observed in other benchmark algorithms. This underscores the robustness of the TCAN algorithm in managing the escalating complexity associated with multi-source environments. Such robustness suggests that the TCAN algorithm is particularly adept at preserving deinterleaving recall despite the challenges posed by interleaved radar signals and the rising density of emitters.

V. CONCLUSION

In the present study, we introduce a novel methodology for deinterleaving radar pulse streams through the application of Temporal Convolutional Attention Networks. By seamlessly integrating temporal convolutional architectures with a sophisticated attention mechanism, TCAN has exhibited exceptional proficiency in the sorting of radar signals, particularly within intricate electromagnetic environments. These environments are typified by challenging factors, including pulse loss, spurious pulses, measurement inaccuracies, and signal sparsity. Our experimental results demonstrate that TCAN outperforms conventional baseline approaches, maintaining elevated recall rates even as the complexity of radar sources escalates. Furthermore, our comprehensive analysis underscores the crucial role played by multi-parameter features and diverse input formats, both of which exert a substantial influence on the model's adaptability. The findings of this study suggest that TCAN offers a promising resolution to

the increasingly intricate demands of deinterleaving, thereby presenting a significant advancement in this domain.

It is important to note that this study focuses on closed-set signal sorting based on a known radar database. The open-set problem, which encompasses unknown classes and random PRI patterns, lies beyond our current scope. Addressing issues such as these, along with ambiguity, requires additional techniques or alternative models for effective resolution. These present promising avenues for our future research.

REFERENCES

- [1] S. Q. Wang, G. P. Hu, Q. L. Zhang, C. Y. Gao, and T. Cai, "The background and significance of radar signal sorting research in modern warfare," *Procedia Computer Science*, vol. 154, pp. 519–523, 2019.
- [2] S. Q. Wang, J. Bai, X. Y. Huang, C. Y. Gao, and P. F. Wan, "Analysis of radar emitter signal sorting and recognition model structure," *Procedia Computer Science*, vol. 154, pp. 500–503, 2019.
- [3] D. K. Barton, *Radar system analysis and modeling*. Artech House, 2004.
- [4] R. Wiley, *ELINT: The interception and analysis of radar signals*. Artech, 2006.
- [5] H. Mardia, "New techniques for the deinterleaving of repetitive sequences," in *IEE Proceedings F (Radar and Signal Processing)*, vol. 136, no. 4. IET, 1989, pp. 149–154.
- [6] D. Milojević and B. Popović, "Improved algorithm for the deinterleaving of radar pulses," in *IEE Proceedings F (Radar and Signal Processing)*, vol. 139, no. 1. IET, 1992, pp. 98–104.
- [7] D. Nelson, "Special purpose correlation functions for improved signal detection and parameter estimation," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4. IEEE, 1993, pp. 73–76.
- [8] K. Nishiguchi and M. Kobayashi, "Improved algorithm for estimating pulse repetition intervals," *IEEE transactions on Aerospace and Electronic Systems*, vol. 36, no. 2, pp. 407–421, 2000.
- [9] Y. Xi, X. Wu, Y. Wu, Y. Cai, and Y. Zhao, "A novel algorithm for multi-signals deinterleaving and two-dimensional imaging recognition based on short-time pri transform," in *2019 Chinese Automation Congress (CAC)*. IEEE, 2019, pp. 4727–4732.
- [10] Z. Ge, X. Sun, W. Ren, W. Chen, and G. Xu, "Improved algorithm of radar pulse repetition interval deinterleaving based on pulse correlation," *IEEE access*, vol. 7, pp. 30 126–30 134, 2019.
- [11] X. Feng, X. Hu, and Y. Liu, "Radar signal sorting algorithm of k-means clustering based on data field," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2017, pp. 2262–2266.
- [12] X. Gong, H. Meng, and X. Wang, "A gmm-based algorithm for classification of radar emitters," in *2008 9th International Conference on Signal Processing*. IEEE, 2008, pp. 2434–2437.
- [13] M. G. S. Ahmed and B. Tang, "New fcm's validity index for sorting radar signal," in *2014 IEEE 17th International Conference on Computational Science and Engineering*. IEEE, 2014, pp. 127–131.
- [14] S. Wang, C. Gao, Q. Zhang, V. Dakulagi, H. Zeng, G. Zheng, J. Bai, Y. Song, J. Cai, and B. Zong, "Research and experiment of radar signal support vector clustering sorting based on feature extraction and feature selection," *IEEE Access*, vol. 8, pp. 93 322–93 334, 2020.
- [15] W. Jian and W. M. Song, "A new radar signal sorting method based on data field," *Applied Mechanics and Materials*, vol. 610, pp. 401–406, 2014.
- [16] Z. Zhao, H. Zhang, and L. Gan, "A multi-station signal sorting method based on tdoa grid clustering," in *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*. IEEE, 2021, pp. 773–778.
- [17] Z. Yu, Y. Wang, and C. Chen, "Radar emitter signal sorting method based on density clustering algorithm of signal aliasing degree judgment," in *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2020, pp. 1027–1031.
- [18] J. Sui, Z. Liu, L. Liu, A. Jung, T. Liu, B. Peng, and X. Li, "Sparse subspace clustering for evolving data streams," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7455–7459.
- [19] J. Sui, Z. Liu, L. Liu, B. Peng, T. Liu, and X. Li, "Online non-cooperative radar emitter classification from evolving and imbalanced pulse streams," *IEEE Sensors Journal*, vol. 20, no. 14, pp. 7721–7730, 2020.
- [20] J. Sui, Z. Liu, L. Liu, A. Jung, and X. Li, "Dynamic sparse subspace clustering for evolving high-dimensional data streams," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 4173–4186, 2020.
- [21] M. A. Nuhoglu, Y. K. Alp, M. E. C. Ulusoy, and H. A. Cirpan, "Image segmentation for radar signal deinterleaving using deep learning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 1, pp. 541–554, 2022.
- [22] T. Chen, B. Yang, and L. Guo, "Radar pulse stream clustering based on maskrcnn instance segmentation network," *IEEE Signal Processing Letters*, vol. 30, pp. 1022–1026, 2023.
- [23] Z.-M. Liu and S. Y. Philip, "Classification, denoising, and deinterleaving of pulse streams with recurrent neural networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 4, pp. 1624–1639, 2018.
- [24] M. Zhu, S. Wang, and Y. Li, "Model-based representation and deinterleaving of mixed radar pulse sequences with neural machine translation network," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 3, pp. 1733–1752, 2021.
- [25] W. Chao, S. Liting, L. Zhangmeng, and H. Zhitao, "A radar signal deinterleaving method based on semantic segmentation with neural network," *IEEE Transactions on Signal Processing*, vol. 70, pp. 5806–5821, 2022.
- [26] A. Al-Malahi, A. Farhan, H. Feng, O. Almaqtari, and B. Tang, "An intelligent radar signal classification and deinterleaving method with unified residual recurrent neural network," *IET Radar, Sonar & Navigation*, vol. 17, no. 8, pp. 1259–1276, 2023.
- [27] X. Li, Z. Liu, and Z. Huang, "Deinterleaving of pulse streams with denoising autoencoders," *IEEE transactions on aerospace and electronic systems*, vol. 56, no. 6, pp. 4767–4778, 2020.
- [28] H. Xiang, F. Shen, and J. Zhao, "Deep toa mask-based recursive radar pulse deinterleaving," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 2, pp. 989–1006, 2022.
- [29] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.
- [30] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.
- [31] Z.-M. Liu, "Pulse deinterleaving for multifunction radars with hierarchical deep neural networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 6, pp. 3585–3599, 2021.
- [32] Z. Zhang, X. Shi, X. Guo, and F. Zhou, "Tr-ragcn-aff-ress: A method for radar emitter signal sorting," *Remote Sensing*, vol. 16, no. 7, p. 1121, 2024.