Multi-Feature Transformer-Based Learning for Continuous Human Motion Recognition with High Similarity Using mmWave FMCW Radar

Yuh-Shyan Chen *, Kuang-Hung Cheng, You-An Xu and Tong-Ying Juang

Department of Computer Science and Information Engineering, National Taipei University, No. 151, University Rd., San Shia District, New Taipei City 23741, Taiwan

Received: 8 October 2022
Accepted: 28 October 2022
Published: 1 November 2022

Abstract: Doppler-radar-based continuous human motion recognition recently has attracted extensive attention, which is a favorable choice for privacy and personal security. Existing results of continuous human motion recognition (CHMR) using mmWave FMCW Radar are not considered the continuous human motion with the high similarity problem. In this paper, we proposed a new CHMR algorithm with the consideration of the high similarity (HS) problem, called as CHMR-HS, by using the modified Transformer-based learning model. As far as we know, this is the first result in the literature to investigate the continuous HMR with the high similarity. To obtain the clear FMCW radar images, the background and target signals of the detected human are separated through the background denoising and the target extraction algorithms. To investigate the effects of the spectral-temporal multi-features with different dimensions, Doppler, range, and angle signatures are extracted as the 2D features and range-Doppler-time and range-angle-time signatures are extracted as the 3D features. The 2D/3D features are trained into the adjusted Transformer-encoder model to distinguish the difference of the high-similarity human motions. The conventional Transformer-decoder model is also re-designed to be Transformer-sequential-decoder model such that Transformer-sequential-decoder model can successfully recognize the continuous human motions with the high similarity. The experimental results show that the accuracy of our proposed CHMR-HS scheme are 95.2% and 94.5% if using 3D and 2D features, the simulation results also illustrates that our CHMR-HS scheme has advantages over existing CHMR schemes.

Keywords: FMCW; continuous human motion recognition; high similarity; multi-feature; Transformer

1. Introduction

Human Motion Recognition (HMR) has recently attracted extensive attention [1,2]. For instance, HMR can detect or prevent falls in residential care homes for elderly people. Human motion recognition (HMR) can be achieved using wearable-sensor-based technologies, camera-based technologies, and radar-based technologies, but wearable-sensor-based solutions require users to attach multiple sensors on their bodies, resulting in a great inconvenience, and camera-based technologies raise privacy concerns [1]. Due to the impact of the COVID-19 disease, most people are increasingly accepting the use of contactless radar-based methods to record human behaviors under privacy-preserving conditions, compared to wearable-sensor-based technologies and camera-based technologies. The key advantages of mmWave FMCW radar, compared with LiDar, are that it is not affected by light, and it is of low cost. Due to these characteristics, it is very useful to apply mmWave FMCW radar applications for automatic car driving and home care, especially for night care for the elderly.

The frequency modulated continuous wave (FMCW) radar-based solutions have achieved many results in human motion recognition (HMR) results [3,4]. In the early stage,
radar-based HMR results are mainly focused on a single HMR for an observation that just only includes one type of motion with fixed time duration from mmWave FMCW radar sensors. Conventional Doppler-radar-based HMR methods [1] mainly utilize the FMCW spectrogram generated using a short-time Fourier transform (STFT) to extract handcrafted features, and machine learning or deep learning classifier is utilized for the feature-based classification. For instance, these include the Doppler-radar-based classification of pedestrians, bicycle users, and human daily activities [5–7]. Machine learning or deep learning-based human motion recognition results recently are widely adopted to improve the recognition accuracy. For instance, the recognition of a human body’s gait using Convolutional Neural Networks (CNN) is investigated in [8,9].

In the normal human activities, people usually perform motions one after the other for varying durations [1]. More recently, continuous HMR based on micro-Doppler features from mmWave FMCW radar has become the focus of research [1]. Some continuous HMR methods mainly rely on individual recurrent neural network or sliding-window-based approaches, which cannot effectively exploit all the temporal information to predict motions. Zhao et al. [1] then proposed an end-to-end network for continuous motion recognition via radar radios. Zhang et al. [2] proposed a dynamic continuous hand gesture recognition based on a frequency-modulated continuous wave radar sensor. Shrestha et al. [10] proposed a continuous human activity monitoring and classification from FMCW radar based on recurrent LSTM and Bi-LSTM network architectures. Zhao et al. [11] also proposed a continuous human motion recognition using micro-Doppler signatures in the scenario with micro-motion interference, using Empirical Mode Decomposition (EMD) to remove non-target interference.

However, all of the mentioned existing continuous HMR results do not investigate the classification issue for the high-similarity motions. We consider three different high-similarity motion groups; walking group, running group, and vertical-motion group. The walking group includes motions of walking forward, walking backward, walking right, and walking left. The running group includes motions of running forward, running backward, running right, and running left. The vertical-motion group contains motions of lying down and falling. We observe that if the continuous HMR for the observed motions are from the same high-similarity motion group, the lower the recognition accuracy will be. This is because that the similar micro-Doppler signature is obtained for all motions selected from the same high-similarity motion group. Therefore, the main motivation and contribution of this work is to propose a new continuous HMR (CHMR) algorithm with the consideration of the high similarity (HS) problem, which is termed the CHMR-HS scheme in this paper.

For HMR and continuous HMR, the Doppler signature and the Range-Doppler-Time signature from FMCW radar are the most common extracted features in the existing literature [10–13]. Papanastasiou et al. [12] proposed a deep learning-based identification of human gait using the Range-Doppler-Time signature. Shrestha et al. [10] proposed a continuous human motion classification from the FMCW radar with Bi-LSTM networks using Doppler signature spectrograms. Zhao et al. [11] proposed a continuous human motion recognition using the Doppler signature. Apart from the use of Doppler-based signature spectrograms, some new recent works further consider the three-dimensional radar data cube; that is, the Range-Doppler-Angle, for the HMR. For instance, Zheng et al. [13] proposed a dynamic hand gesture recognition algorithm by extracting the features of the Range-Doppler Map (RDM), Range-Azimuth Map (RAM), and Range-Elevation Map (REM) of time signatures as an input of the trained model.

In this paper, we propose a continuous HMR algorithm with high similarity (CHMR-HS) by using multi-features. To investigate the effects of the spectral–temporal multi-features with different dimensions, Doppler, range, and angle signatures are extracted as the 2D features, and range-Doppler-time and range-angle-time signatures are extracted as the 3D features. The 2D/3D features are separately trained into the modified Transformer-encoder model to effectively distinguish the difference of the high-similarity human motions. Our CHMR-HR scheme is to re-design a signature-to-signature learning model,
where the input-signature are the 2D/3D feature signatures and the output-signature is the recognition result of the motions one after the other for varying durations. It is also worth noting that the 2D/3D feature signatures with multiple motions are fed into our designed signature-to-signature learning model to produce the corresponding multiple motion recognition results under the non-fixed recognition number.

In the sequence-to-sequence learning model, we propose a Transformer-based learning model, which is constructed using modified Transformer-encoder and Transformer-decoder models. In our Transformer-encoder part, the Doppler, angle, and range feature signatures are independently embedded and decoded using disjoint Transformer-encoders, and the corresponding output features are then fused to the fused feature. The conventional Transformer-decoder model is re-designed to be the Transformer-sequential-decoder model, such that the Transformer-sequential-decoder model can recognize continuous human motion with the high similarity from the trained Transformer-encoder model. In our Transformer-sequential-decoder part, an extra weight is trained sequentially via the order of the Doppler feature, angle feature, and range feature, with their cross-attentions from Transformer-encoder model. The trained extra weight will be added into the self-attention to enhance our signature-to-signature learning model. Furthermore, our Transformer-sequential-decoder is re-designed to allow our continuous HMR scheme to recognize the non-fixed number of motions with the dynamic motion duration. It is noted that connectionist temporal classification (CTC) [14] technique is widely used to perform the model training between blank segments and motion labels, due to the CTC classifiers being able to effectively predict the motions between motions in the early training stage. In our work, the CTC technique [14] is also tailored into our trained model, so that the early training stage of the CHMR-HS scheme can converge faster. As shown in Figure 1, our CHMR-HS scheme is illustrated. The contributions of this paper are listed as follows.

• As far as we know, this is the first result in the literature to investigate continuous HMR with the consideration of high-similarity motions. It is observed that if the observed continuous motions are from the same high-similarity motion group, the recognition accuracy is degraded due to the similar micro-Doppler signature being obtained for the motions selected from the same high-similarity motion group.

• To investigate the effects of the spectral–temporal multi-features with different dimensions, our proposed CHMR-HS scheme is trained using 2D/3D multi-features, where Doppler, range, and angle signatures are extracted as the 2D features, and the range-Doppler-time and range-angle-time signatures are extracted as the 3D features.

• To effectively recognize continuous human motion recognition with high similarity, our CHMR-HS scheme presents a new Transformer-based learning model, where the new Transformer-based learning model is constructed via modified Transformer-encoder and the new proposed Transformer-sequential-decoder models.

• The recognition accuracies of our proposed CHMR-HS scheme for continuous human motion recognition are 95.2% and 94.5% using 3D and 2D features under the motion samples of CHMR that are randomly selected from any motion groups, which are higher than those of the existing results by 2.1% and 1.4%, respectively.

• The recognition accuracies of our proposed CHMR-HS scheme are 94.5% and 94.1% using 3D and 2D features, which are higher than those of the existing results by 12% and 11.6%, if motion samples of CHMR are only selected from the same high-similarity motion groups. It is shown that our proposed CHMR-HS scheme for CHMR can more effectively improve the recognition accuracy for CHMR with high similarity.
Figure 1. Our proposed CHMR-HS scheme.

The rest of the paper is organized as follows. Section 2 describes the related work and motivation. Section 3 introduces the system model, problem formulation, and basic ideas. Section 4 describes the proposed CHMR-HS scheme. Section 5 discusses the performance results. Finally, Section 6 concludes the paper.

2. Related Work

This section first describes the related works in Section 2.1 and discusses the research motivation for the study in Section 2.2.

2.1. Related Work

The time-frequency diagram of FMCW radar is adopted for HMR, and the phase difference of the radar echo data is extracted using Fourier transform. Some related works for micro-Doppler modulation [15], background subtraction [16], and interference mitigation [17] are described. Chen et al. [15] proposed a micro-Doppler modulation model to study the vibration and rotation from the micro-motions of the detected objects, and to analyze the micro-Doppler effects. The moving target indication (MIT) digital filtering method is designed by Ash et al. [16] as the background subtraction to effectively remove returns from stationary and slow-moving objects. Constant false alarm rate (CFAR) detector-based approaches are proposed by Wang [17] for the interference mitigation of FMCW radars.

There are some interesting single HMR works based on machine learning or deep learning algorithms. Kim et al. [18] proposed a human detection and activity classification for single motion based on micro-Doppler signatures using deep convolutional neural networks. To alleviate the burden of collecting and annotating a large number of radar samples, Li et al. [8] proposed an HMR with an instance-based transfer learning (ITL) method under the limited radar micro-Doppler (MD) signatures. Sakagami et al. [9] proposed an accuracy improvement of HMR using CNN, using three 2D domains (Time-Doppler, Time-Range, and Range-Doppler) of mmWave FMCW as the motion data. Khalid et al. [19] proposed a multi-view CNN-LSTM architecture for the radar-based HMR, while the multi-view radar features includes raw features, energy dispersion based features, temporal difference based features, and auxiliary features.

Continuous HMR results are more practical, due to the human usually performing motions one after the other for varying durations. So far, there are some useful HMR results with the assistance of machine learning or deep learning algorithms. Shrestha
et al. [10] proposed a continuous human action classification from FMCW radar with Bi-LSTM networks, while this work utilizes radar data in the form of a continuous temporal sequence of micro-Doppler or range-time information. Zhao et al. [1] then proposed an end-to-end network for continuous motion recognition via radar radios. To guarantee the effective use of all the temporal information, the attention-based encoder–decoder structure encodes the fused data and selects the temporal information for recognition. Zhang et al. [2] proposed a dynamic continuous hand gesture recognition based on a frequency-modulated continuous wave radar sensor, and a recurrent 3D convolutional neural network is used to perform the classification of dynamic hand gestures. Distributed radar-based solutions and range-Doppler trajectory methods are recently further considered for CHMR. Guendel et al. [20] proposed a continuous Activities of Daily Living (ADL) recognition in an arbitrary movement direction using distributed pulsed Ultra-Wideband (UWB) radars in a coordinated network. To recognize continuous human motions with various conditions for the real-living environment, Ding et al. [21] proposed a novel dynamic range-Doppler trajectory (DRDT) method using the FMCW radar system.

Transformer-based encoder–decoder architecture is very suitable for the training the time-series data for the continuous HMR problem. Vaswani et al. [22] initially proposed a Transformer-based encoder–decoder architecture with the self-attention mechanism, while the self-attention mechanism of the Transformer-based model is very efficient for the sequence-to-sequence learning task. Hu et al. [23] modified the Transformer-based architecture by proposing a multimodal multitask learning with a unified Transformer. Lohrenz et al. [24] proposed a multi-encoder learning and stream fusion for Transformer-based end-to-end automatic speech recognition. It is valuable to develop the new continuous HMR scheme based on Transformer-based encoder–decoder architecture.

2.2. Motivation

Radar-based HMR technology can provide people with human motion monitoring services with high stability, high privacy, and more privacy. All of the mentioned existing CHMR results do not investigate the classification issue for the high-similarity motions. For instance, some CHMR results [1,2,11] utilized MLP classifier, CTC classifier, and RNN-based classifier to recognize the continuous human motion. In the paper, three different high-similarity motion groups, the walking group, running group, and vertical-motion group, are considered in this paper. The recognition of continuous human motion with high similarity has high challenges under the limitation of only radar-based images being allowed used to satisfy the purpose of the high stability, high privacy, and more privacy.

As mentioned before, the walking group includes motions of walking forward, walking backward, walking right, and walking left; the running group includes motions of running forward, running backward, running right, and running left; and the vertical-motion group contains motions of lying down and falling. It is observed that if the continuous HMR for the observed motions is from the same high-similarity motion group, the recognition accuracy is degraded. For example, as shown in Figure 2a, there are four motions. In order, they are running forward, running forward, running right, and running left from the same running group. It can be observed that each Doppler signature is very similar with each other. In contrast, as shown in Figure 2b, there are still four motions. In order, they are walking forward, falling, walking backward, and lying down. The motions of walking forward and walking backward belong to the walking group, but falling and lying down belong to the vertical-motion group. The Doppler signatures of walking forward and walking backward are similar, and the Doppler signatures of falling and lying are similar. However, the Doppler signatures of walking forward or walking backward with the Doppler signatures of falling and lying are quite different. If the Doppler signatures of the observed period are quite different, it will be more easy for the CHMR to recognize the different motions. However, if the all the Doppler signatures of the observed period are similar, it will increase the difficulty of CHMR and degrade the recognition accuracy for continuous human motions. Therefore, the main motivation of this work is to propose a
new CHMR algorithm with the consideration of the high similarity (HS) issue, called the CHMR-HS scheme in this paper.

![Figure 2. Schematic diagram of continuous human motion with high similarity; (a) continuous motion of running motion group, (b) continuous motion of walking, lying down, and falling motions.](image)

3. Preliminaries

This section describes the system model, the problem formulation, and the basic idea in Sections 3.1–3.3, respectively.

3.1. System Model

The system model of the CHMR-HS scheme consists of the radar data collection module, the raw data preprocessing module, and the human motion recognition module. The radar data collection module is to collect the raw frequency-modulated continuous wave (FMCW) radar, the raw data preprocessing module is to extract 2D/3D features from the raw FMCW data, and the human motion recognition module is to perform the continuous HMR with HS capability. Twelve motion patterns are considered in this paper, including waving, jumping forward, falling, lying down, walking forward, walking backward, walking left, walking right, running forward, running backward, running left, and running right. Among them, the motions of walking forward, walking backward, walking left, and walking right with similar micro-Doppler effects are collected as the walking group. The motions of running forward, running backward, running right, and running left with similar micro-Doppler effects are collected as the running group. The motions of lying down and falling with similar micro-Doppler effects are collected as the vertical-motion group.

In the radar data collection module, as shown in Figure 3, the raw data are extracted from the AWR1642-BOOST mmWave FMCW sensing device from Texas Instruments [25], which is the IWR1642 single-chip from a 76 GHz to 81 GHz mmWave sensor integrating DSP and microcontroller (MCU) evaluation module. The basic principle of the mmWave FMCW sensing device is to use a frequency-modulated continuous wave (FMCW) for frequencies emitted at different times via MIMO; the distance between objects is detected through the difference between the transmitting antenna and the receiving antenna, and the Doppler frequency shift is calculated through the phase difference in the sine wave.
The criteria for designing the FMCW Radar is that FMCW Radars transmit a frequency modulated continuous wave signal, and a reflected FMCW signal from the remote target is mixed with the transmitted signal to generate a signal at an intermediate frequency (IF), which is used in the range, Doppler, and angle calculations [1,2].

The raw data preprocessing module, as shown in Figure 3, is to eliminate the noise and interference problems, to provide more easily recognized FMCW radar images to the human motion recognition module. The main work is to perform a background removal operation, which results in the cross-reflection effects of non-dynamic objects due to the obfuscation of detected objects and the background. Another major task is to further perform the feature enhancement operation to enhance the features of high-movement observed human motions through the Doppler frequency shift of the highly reflective objects. After the background removal and feature enhancement operations, the 2D/3D features will be extracted.

In the human motion recognition module, as shown in Figure 3, the raw 2D/3D features are separately trained and fused together to be fused features in the Transformer-encoder model. The importance of the 2D/3D features of each time segment of continuous human motion will be trained in the Transformer-sequential-decoder model, as shown in Figure 3. A joint decoding is used for the recognition optimization.

![System structure of CHMR-HS scheme.](image)

**Figure 3.** System structure of CHMR-HS scheme.

### 3.2. Problem Formulation

Before defining the problem formulation of our work, some notations are defined. Let $S$ be a set of training samples of radar images, $S = x_1, ..., x_i, ..., x_l$, where $1 \leq i \leq l$, $x_i$ is $i$-the training sample of radar images from continuous human motions for a period of time, and let $l$ be the total training samples. Note that $x_i$ can be a sample of 2D radar features or 3D radar features for a period of time; let $S_{2D}$ be simplified as $S$ if all $x_i \in S$ are 2D radar features for a period of time, where each $x_i$ contains $i$-th training sample of the Doppler signature, angle signature (including elevation or azimuth), and range signature for a period of time. In addition, we may let $S_{3D}$ be also simplified as $S$ if all $x_i \in S$ are 3D radar features for a period of time, where each $x_i$ contains the $i$-th training sample of the range-Doppler signature and range-angle signature for a period of time. In this work,
we define $K$ different motions and $n$ high-similarity motion groups, such that $m$ different motions are distributed into $n$ high-similarity motion groups. For $x_i \in S$, for $S \in S_{2D}$ or $S_{3D}$, each continuous human motion contains $J$ motions, their corresponding labels of $J$ motions are denoted as label sequence $(l_1^i, ..., l_J^i, ..., l_K^i)$ for $x_i$, where $1 \leq j \leq J$. Let $Y = y_1, ..., y_I, ..., y_{1}$ be the set of label sequences for $S$, if $S \in S_{2D}$ or $S_{3D}$, where the label sequence of $x_i$ is $y_i = (l_1^i, ..., l_J^i, ..., l_K^i)$ and $J$ is a dynamic value for different $x_i$.

Our work has three decoders, there are the CTC decoder, Transformer-sequential-decoder, and Transformer-importance-decoder. We first describe the loss functions of these three decoders as follows.

First, the loss function of the CTC decoder, $L_{CTC}$, is described below. To overcome the blank segment and the meaningless feature sequence for the long-time series motion recognition, the connectionist temporal classification (CTC) [14] technique is useful for our scheme, so the CTC technique is integrated into our scheme as one of our feature decoders. Given a set of training samples of radar images, $S = (x_1, ..., x_I, ..., x_I)$, where $1 \leq i \leq I$, and $Y = (y_1, ..., y_I, ..., y_{1})$ is a set of label sequences for $S$, if $S \in S_{2D}$ or $S_{3D}$, where the label sequence of $x_i$ is $y_i = (l_1^i, ..., l_J^i, ..., l_K^i)$ and $l_j^i$ represents the label of the $j$-th motion representation for $x_i$, where $1 \leq j \leq J$ and $J$ is a dynamic value for different $x_i$. Let $i$-th continuous human motions $x_i$ of $S$ have the label sequence $(l_1^i, ..., l_J^i, ..., l_K^i)$, if it is assumed that the predicted result classified by the CTC decoder of $x_i$ is $\hat{y}_i = (\hat{l}_1^i, ..., \hat{l}_J^i, ..., \hat{l}_K^i)$, where $\hat{y}_i = P(y_i|x_i)$ under $P(y_i|x_i)$ is probability of hypothesis $y_i$. We have a set of predicted label sequences $\hat{Y} = (\hat{y}_1, ..., \hat{y}_I, ..., \hat{y}_{1})$ for $S = (x_1, ..., x_I, ..., x_I)$. This work aims to find the best solution for the classification of the prediction results of the CTC decoder. Let $\hat{Y}^* = \arg \max P(Y | S)$, where $\hat{Y}^* = (\hat{y}_1^*, ..., \hat{y}_I^*, ..., \hat{y}_{1}^*)$ is the best solution, and where $\hat{y}_i^* = (\hat{l}_1^i, ..., \hat{l}_J^i, ..., \hat{l}_K^i)$ is obtained by the CTC decoder, where each $\hat{l}_k^i$ is the best solution. Based on the cross-entropy loss function [26], the loss function of CTC decoder, $L_{CTC}$ is defined.

$$L_{CTC}(Y, \hat{Y}) = -\frac{1}{I \times J} \sum_{i=1}^{I} \sum_{j=1}^{J} y_{ij} \times \log(\hat{y}_{ij}) = -\frac{1}{I \times J} \sum_{i=1}^{I} \sum_{j=1}^{J} I \sum_{k=1}^{K} \hat{l}_j^i \times \log \times (\hat{l}_j^i),$$

where, $y_{ij}, l_j^i$ represents the ground-truth label of the $j$-th motion of $x_i$, $\hat{y}_{ij} = (\hat{l}_1^i, ..., \hat{l}_J^i, ..., \hat{l}_K^i)$ is the predicted label sequence for the CTC classification decoder for $x_i$ to the $j$-th motion corresponding to $k$-th motion category for all $1 \leq k \leq K$, and $\hat{l}_j^i$ is the predicted label of the $j$-th motion corresponding to the $k$-th motion category, where $1 \leq k \leq K$. Consequently, the best loss function $L_{CTC}(Y, \hat{Y}^*)$ is acquired under the best solution $\hat{Y}^*$ is obtained.

Next, the Transformer-sequential-decoder loss function $L_{Seq}$ is described below. To make the continuous human motion more effectively learn its temporal correlation, we build a second decoder, called a Transformer-sequential-decoder, in our trained model. The prediction result of the Transformer-sequential-decoder classification is denoted as $\bar{Y}_i = (\bar{l}_1^i, ..., \bar{l}_J^i, ..., \bar{l}_K^i)$, where $\bar{l}_j^i$ represents the predicted label of the $j$-th motion representation for $x_i$, where $1 \leq j \leq J$, and $J$ is a dynamic value for different $x_i$, if $\bar{y}_i = P(y_i | x_i)$ under $P(y_i | x_i)$ is the predicted probability. Therefore, we have a set of predicted label sequences $\bar{Y} = (\bar{y}_1, ..., \bar{y}_I, ..., \bar{y}_{1})$ for $S = (x_1, ..., x_I, ..., x_I)$. The goal is to find the best solution for the classification of the prediction results of the Transformer-sequential decoder $\bar{Y}^* = \arg \max P(Y | S)$, where $\bar{Y}^* = (\bar{y}_1^*, ..., \bar{y}_I^*, ..., \bar{y}_{1}^*)$ is the best classification solution, and where $\bar{y}_i^* = (\bar{l}_1^i, ..., \bar{l}_J^i, ..., \bar{l}_K^i)$ is obtained using the Transformer-sequential decoder, where each $\bar{l}_k^i$ is the best solution. Based on the cross-entropy loss function [26], the loss function of the Transformer-sequential-decoder, $L_{Seq}$, is defined.

$$L_{Seq}(Y, \bar{Y}) = -\frac{1}{I \times J} \sum_{i=1}^{I} \sum_{j=1}^{J} y_{ij} \times \log(\bar{y}_{ij}) = -\frac{1}{I \times J} \sum_{i=1}^{I} \sum_{j=1}^{J} I \sum_{k=1}^{K} \bar{l}_j^i \times \log \times (\bar{l}_j^i),$$
where, $y_{ij}$, $l_i^j$ represent the ground-truth label of the $j$-th motion of $x_i$, $\hat{y}_{ij} = (\hat{l}_{i,j,1}^1, ..., \hat{l}_{i,j,k}^j, ..., \hat{l}_{i,j,K}^j)$ is the predicted label sequence for the Transformer-sequential decoder for $x_i$ to the $j$-th motion corresponding to $k$-th motion category for all $1 \leq k \leq K$, and $\hat{l}_{ij,k}$ is the predicted label of the $j$-th motion corresponding to the $k$-th motion category, where $1 \leq k \leq K$. Consequently, the best loss function $\mathcal{L}_{Seq}^*(Y, \hat{Y}^*)$ acquired under the best solution $\hat{Y}^*$ is obtained.

The loss $\mathcal{L}_{Imp}^*$ of the Transformer-importance-decoder is described below. To handle the classification of continuous human motions with high similarity, we build a third decoder, called a Transformer-importance-decoder, in our trained model. Assume that the prediction result of the Transformer-importance-decoder classification of $x_i$ is $\bar{y}_i = (\bar{l}_{i,1}^1, ..., \bar{l}_{i,f}^j, ..., \bar{l}_{i,K}^j)$, where $\bar{l}_{ij}^j$ representing the predicted label of the $j$-th motion representation for $x_i$ for $1 \leq j \leq f$, $\bar{y}_i = P(y_i \mid x_i)$ under $P(y_i \mid x_i)$ is the predicted probability. Therefore, we have a set of predicted label sequences $\bar{Y} = (\bar{y}_{i,1}, ..., \bar{y}_{i,f}, ..., \bar{y}_{i,S})$ for $S = x_1, ..., x_{\mu}, ..., x_I$. The goal is to find the best solution for the classification of the prediction results of the Transformer-importance-decoder if $\hat{Y}^* = \text{arg} \max P(Y \mid S)$, where $\hat{Y}^* = (\hat{y}_{1,1}^1, ..., \hat{y}_{1,f}^f, ..., \hat{y}_{I,K}^j)$ is the best classification solution. $\hat{Y}^* = (\hat{l}_{1,1}^1, ..., \hat{l}_{1,f}^f, ..., \hat{l}_{I,K}^j)$ obtained using the Transformer-importance-decoder, and $\hat{l}_{ij}^j$ is the best solution. Based on the cross-entropy loss function [26], the loss function of the Transformer-importance-decoder, $\mathcal{L}_{Imp}^*$ is defined.

$$
\mathcal{L}_{Imp}^*(Y, \hat{Y}) = -\frac{1}{I \times J} \sum_{i=1}^{I} \sum_{j=1}^{J} y_{ij} \times \log(\bar{y}_{ij}) = -\frac{1}{I \times J \times K} \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \hat{l}_{ij,k} \times \log \left(\hat{l}_{ij,k}^j\right),
$$

where, $y_{ij}$, $l_i^j$ represents the ground-truth label of the $j$-th motion of $x_i$, $\bar{y}_{ij} = (\bar{l}_{i,j,1}^1, ..., \bar{l}_{i,j,k}^j, ..., \bar{l}_{i,j,K}^j)$ is the predicted label sequence for the Transformer-importance-decoder for $x_i$ to the $j$-th motion corresponding to $k$-th motion category for all $1 \leq k \leq K$, and $\bar{l}_{ij,k}$ is the predicted label of the $j$-th motion corresponding to the $k$-th motion category, where $1 \leq k \leq K$. Consequently, the best loss function $\mathcal{L}_{Imp}^*(Y, \hat{Y}^*)$ acquired under the best solution $\hat{Y}^*$ is obtained.

Finally, the overall best loss function $\mathcal{L}_{CHMR}^*$ is

$$
\mathcal{L}_{CHMR}^* = \alpha \times \mathcal{L}_{CTC}^*(Y, \hat{Y}^*) + \beta \times \mathcal{L}_{Seq}^*(Y, \hat{Y}^*) + \gamma \times \mathcal{L}_{Imp}^*(Y, \hat{Y}^*),
$$

where $\alpha$, $\beta$, and $\gamma$ are the tunable parameters and $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$, $0 \leq \gamma \leq 1$, and $\mathcal{L}_{CTC}^*(Y, \hat{Y}^*)$ is the best loss function of CTC decoder, $\mathcal{L}_{Seq}^*(Y, \hat{Y}^*)$ is the best loss function of the Transformer-sequential-decoder, and $\mathcal{L}_{Imp}^*(Y, \hat{Y}^*)$ is the best loss function of the Transformer-importance-decoder.

### 3.3. Basic Idea

One novel CHMR result [11] extracts the motion features through SENe; BLSTM is used as an encoder to learn the time series feature, and the Connectionist Temporal Classification (CTC) [14] is then utilized as a decoder to solve blank and featureless segmentation problems. Figure 4a shows this method. One other novel result for recognizing single gesture motion is from [13], where a Transformer-encoder architecture is used by learning the temporal features from time series of the Range-Doppler Map (RDM), Range Azimuth Map (RAM), and Range Elevation Map (REM), and these temporal features are fused as a fused feature. Figure 4b displays the method. In the paper, the temporal features of the time series of the Range-Doppler Map (RDM), Range Azimuth Map (RAM), and Range Elevation Map (REM) are simplified as 3D features.

In our proposed Transformer-encoder model, the Doppler-based feature, angle-based feature, and range-based feature will be independently learnt as different tasks. In our proposed Transformer-decoder model, the Doppler-based feature, angle-based feature, and range-based features will be sequentially decoded by order through a cross-attention mech-
anism, called the Transformer-sequential-decoder. A joint decoding is then used to combine 
the loss value from CTC [14] and two loss values from our proposed Transformer-sequential-
decoder model for the recognition optimization. Figure 4c illustrates this concept.

The importance ratio of the Doppler-based feature, angle-based feature, and range-
based features for each different time segment of the observed time period should be 
different, as shown in Figure 5. The importance ratio of the Doppler-based feature, angle-
based feature, and range-based features of each time segment will be learned in our 
design algorithm. This importance ratio is used to significantly improve our CHMR result, 
especially for the high-similarity motions.

Figure 4. Comparison with (a) the MSENet BLSTM scheme, (b) Transformer-Encoder scheme, and 
(c) our proposed CHMR-HS scheme.
4. Our Proposed CHMR-HS Algorithm

In this section, our Transformer-Based Learning with High Similarity for CHMR is divided into four phases, as follows. The flowchart of the proposed algorithm is also given in Figure 6.

1. **Data pre-processing phase**: The main work is to obtain a wider range of field-of-view data, eliminate the radar signal noise, extract the distance frequency and Doppler frequency information, and reduce the noise frequency generated by the Doppler effects by extracting meaningful 2D and 3D features.

2. **Multi-feature embedding phase**: This phase is to embed 2D and 3D features through the CNN embedding module into our designed network.

3. **Transformer-based encoding phase**: This phase is to encode 2D and 3D embedded features, including positional embedding, Transformer-encoder, and feature fusion operations, into the Transformer-based network.

4. **Transformer-sequential-decoding phase**: This phase describes how to perform CTC-decoder, Transformer-sequential-decoder, and Transformer-importance-decoder, and how to obtain the best sequence-to-sequence result from joint-decoding among three decoders.

![Figure 6](image-url)
\( S_{\text{FMCW}}^t \) is represented as the \( t \)-th continuous FMCW radar signals, where \( m \) is the sampling chirp number, \( n \) is the received antenna number, and \( T \) is the chirp number times the chirp duration time. If \( x_i \in S_{3D} \), \( x_i \) consists of multi-features, including a \( T \) number of range-Doppler signatures, denoted as \( RD_{ST} \), \( T \) number of range-Doppler signatures, denoted as \( RAS_T \), and \( T \) number of range-elevation signatures, denoted as \( RES_T \), which are extracted from continuous \( T \) FMCW signal samples \( S_{\text{FMCW}}^t \), for \( 1 \leq t \leq T \), we let \( x_i = RD_{ST}, RAS_T, RES_T \in S_{3D} \). If \( x_i \in S_{2D} \), \( x_i \) consists of multi-features, including a \( T \) number of Doppler signatures, denoted as \( DS_T \), \( T \) number of range signatures, denoted as \( RDS_T \), \( T \) number of azimuth signatures, denoted as \( AST_T \), and \( T \) number of elevation signatures, denoted as \( ES_T \), which are also extracted from the continuous \( T \) FMCW signal samples \( S_{\text{FMCW}}^t \), for \( 1 \leq t \leq T \), as follows.

To obtain a wider range of field-of-view data, a Binary Phase Modulation (BPM) [27], or called as BPM-MIMO scheme, is firstly executed by utilizing the FMCW MIMO, by applying the BPM function [27] to yield \( S_{\text{FMCW}}^t \) as \( S_{\text{FMCW}}^{p,m,n,c} \), for \( 1 \leq t \leq T \), under \( S_{\text{FMCW}}^t \), for \( 1 \leq t \leq T \), the continuous \( T \) FMCW signal samples, where \( S_{\text{FMCW}}^{p,m,n,c} \in \mathbb{R}^{m \times n \times c} \), where \( 1 \leq t \leq T \), each is represented as the \( m \times n \times c \) 3-dimensional data matrix, where \( m \) is the sampling chirp number, \( n \) is the received antenna number, and \( c \) is the chirp number utilized after performing the BPM algorithm.

To eliminate the radar signal noise, a Moving Target Indication (MTI) filtering algorithm [16] is then utilized, the MTI noise filtering function is performed on \( S_{\text{FMCW}}^{p,m,n,c} \) to obtain \( M_{m,n,c}^{p} \), where \( 1 \leq t \leq T \), such that \( M_{m,n,c}^{p} = \text{MTI}(S_{\text{FMCW}}^{p,m,n,c}) \), where \( M_{m,n,c}^{p} \) is represented as an \( m \times n \times c \) three-dimensional data matrix after performing the MTI noise filtering function to eliminate the radar signal noise, where \( 1 \leq t \leq T \). To avoid the different background issue, we adopt the MTI (Moving Target Indication) filter algorithm in our paper. The signal of the non-moving object (no phase difference) will be removed to ensure that the detected object is only detected in a moving condition, such that the movement, or called as phase difference, can be directly retained. We may also call this the static removal operation.

To extract the distance frequency and the Doppler frequency information of the detected objects, a Fast Fourier Transform (FFT) algorithm [4] is applied to the \( M_{m,n,c}^{p} \) data matrix to obtain \( F_{m,n,c}^{p} \) data matrix, such that \( F_{m,n,c}^{p} = \text{FFT}(M_{m,n,c}^{p}) \), where \( 1 \leq t \leq T \), by performing the FFT procedure on each column of \( M_{m,n,c}^{p} \), where \( 1 \leq t \leq T \). Then, a Fast Fourier Transform (FFT) [4] algorithm is performed on \( F_{m,n,c}^{p} \) to obtain the \( F_{m,n,c}^{p} \) data matrix, such that \( F_{m,n,c}^{p} = \text{FFT}(F_{m,n,c}^{p}) \), where \( 1 \leq t \leq T \), by performing the FFT procedure on each row of \( F_{m,n,c}^{p} \), where \( 1 \leq t \leq T \).

To reduce the noise frequency generated by the Doppler effects, a Constant False Alarm Rate (CFAR) filtering algorithm [17] is also utilized for the FFT data matrix \( F_{m,n,c}^{p} \) to acquire a new data matrix \( C_{m,n,c}^{p} \), such that \( C_{m,n,c}^{p} = \text{CFAR}(F_{m,n,c}^{p}) \), where \( 1 \leq t \leq T \).

The purpose of the background denoising algorithm is to extract the phase frequency between the signals. In our work, we only take the data frames with the largest signals after the Fourier transformation, and then concatenate these data frames together. The intuition is a continuous echo map of the torso. If one or two noises are larger than the torso reflections, or there is a small noise in the process of denoising, it does not matter for the misguided training, since one or two noises of the tens to hundreds of data frames will be moderately retained and will not be particularly problematic.

In the following, we describe how to obtain \( x_i = \{ RD_{ST}, RAS_T, RES_T \} \in S_{3D} \) from the \( C_{m,n,c}^{p} \), where \( 1 \leq t \leq T \). We show how to generate \( RD_{ST} = \{ RDM_{m,n,c}^{p}, ..., RDM_{m,n,c}^{p} \} \), \( 1 \leq t \leq T \), from \( C_{m,n,c}^{p} \) as follows. We let \( RDM_{m,n,c}^{p} \) be an \( m \times n \) two-dimensional data matrix using \( RDM_{m,n,c}^{p} = \sum_{c=1}^{c} C_{m,k,c}^{p} \), so the range-Doppler signature \( RD_{ST} = \{ RDM_{m,n,c}^{p}, ..., RDM_{m,n,c}^{p}, RDM_{m,n,c}^{p} \} \), where \( 1 \leq t \leq T \), is finally obtained. Based on the beamforming [28] processing principle, we may ob-
tain the range-azimuth signature, $RAS_T = \{RAM_{m,c}^1, ..., RAM_{m,c}^r, RAM_{m,c}^T\}$, where $1 \leq t \leq T$, and range-elevation signature, $RES_T = \{REM_{m,c}^1, ..., REM_{m,c}^r, REM_{m,c}^T\}$, where $1 \leq t \leq T$. A beamforming function $B$, Reference [28] is applied to extract the angle frequency information from the received antennas; $BA_{m,c}^1$ and $BE_{m,c}^1$ are obtained, where $BA_{m,c}^1 = B(\hat{f}_{m,1,c}^1, \hat{f}_{m,2,c}^1, ..., \hat{f}_{m,n,c}^1)$ and $BE_{m,c}^1 = B(\hat{f}_{m,1,c}^1, \hat{f}_{m,2,c}^1, ..., \hat{f}_{m,n,c}^1)$, where the first to $j$-th azimuth antenna data are $\hat{f}_{m,1,c}^j, \hat{f}_{m,2,c}^j, ..., \hat{f}_{m,n,c}^j$, and $(j+1)$-th to $n$-th elevation antenna data are $\hat{f}_{m,j+1,c}^j, \hat{f}_{m,j+2,c}^j, ..., \hat{f}_{m,n,c}^j$ of $\hat{f}_{m,n,c}^j$, where $1 \leq t \leq T$. Finally, $RAM_{m,c}^1 = CFAR(BA_{m,c}^1)$ and $REM_{m,c}^1 = CFAR(BE_{m,c}^1)$ are obtained, where $1 \leq t \leq T$.

Figure 7 shows an example for our range-Doppler signature $RDS_T$, range-azimuth signature $RAS_T$, and range-elevation signature $RES_T$.

In the following, we describe how to obtain $x_{i} = \{DS_{T}, RS_{T}, AS_{T}, ES_{T}\} \in S_{2D}$ from $\{RDS_T, RAS_T, RES_T\} \in S_{3D}$. First, we show how to obtain the Doppler signature $DS_T = \{D_{m,c}^1, ..., D_{m,c}^r, D_{m,c}^T\}$, where $1 \leq t \leq T$. Given $\hat{f}_{m,c}^t = FFT(M_{m,c}^t)$, where $1 \leq t \leq T$, we let $D_{m,c}^t = \sum_{m=1}^{m} STFT(\sum_{j=1}^{c} \hat{f}_{m,j}^t)$, by performing the Short-Time Fourier Transform (STFT) operation on $\sum_{j=1}^{c} STFT(\sum_{t=1}^{T} \hat{f}_{m,j}^t)$, where $1 \leq t \leq T$. Next, the range signature $RS_T = \{R_{m,c}^1, ..., R_{m,c}^r, R_{m,c}^T\}$ is constructed using $R_{m,c}^t = \sum_{c=1}^{c} RDM_{m,c}^t$, where $R_{m,c}^t$ is the one-dimensional range data of $m$ sampling chirp numbers within $t$ sampling time, under $RDS_T = \{R_{m,c}^1, ..., R_{m,c}^r, R_{m,c}^T\}$, where $1 \leq t \leq T$. The azimuth signature $AS_T = \{A_{m,c}^1, ..., A_{m,c}^r, A_{m,c}^T\}$ is then constructed, such that $A_{m,c}^t = \sum_{c=1}^{c} RDM_{m,c}^t$, where $A_{m,c}^t$ is the one-dimensional azimuth data of $m$ sampling chirp numbers within $t$ sampling time, under a range-azimuth signature $RAS_T = \{RAM_{m,c}^1, ..., RAM_{m,c}^r, RAM_{m,c}^T\}$, where $1 \leq t \leq T$. Finally, the elevation signature $ES_T = \{E_{m,c}^1, ..., E_{m,c}^r, E_{m,c}^T\}$ is formed by $E_{m,c}^t = \sum_{c=1}^{c} REM_{m,c}^t$, where $E_{m,c}^t$ is the one-dimensional elevation data of $m$ sampling chirp numbers within $t$ sampling time, under the range-elevation signature $RES_T = \{REM_{m,c}^1, ..., REM_{m,c}^r, REM_{m,c}^T\}$, where $1 \leq t \leq T$.

An example is shown in Figure 8, with the generation of 2D feature signatures of the Doppler signature $DS_T$, range signature $RS_T$, azimuth signature $AS_T$, and elevation signature $ES_T$. An example of CHM is given in Figure 9a for the human motions of wave, jumping forward, and the vertical-motion group (including falling and lying down). Figure 9b provides the human motions of the walking group (including walking forward, walking backward, walking left, and walking right). Figure 9c displays the human motions of the running group (including running forward, running backward, running left, and running right). Figure 10a give an examples of continuous human motions with 2D feature signatures under four motions in 12 s, and Figure 10b shows another example of two motions in 12 s.
Figure 8. Example of the generation of 2D feature signatures.

(a)

Figure 9. Cont.
Figure 9. (a) Human motions of wave, jumping forward, vertical-motion group (including falling and lying down), (b) human motions of walking group (including walking forward, walking backward, walking left, walking right), (c) human motions of running group (including running forward, running backward, running left, running right).
Figure 10. Examples of continuous human motions with 2D feature signatures under (a) four motions (12 s), (b) two motions (12 s).

4.2. Multi-Feature Embedding Phase

The task of the multi-feature embedding phase is to embed 2D features \( x_i = \{DS_T, RS_T, AS_T, ES_T\} \in S_{2D} \), and \( S = \{x_1, ..., x_i, ..., x_I\} \), where \( DS_T = \{D_{m_1}, ..., D_{m_i}, ..., D_{m_I}\} \),
\( RS_T = \{ R_{m1}^t, ..., R_{mT}^t \}, AS_T = \{ A_{m1}^t, ..., A_{mT}^t \}, \) and \( ES_T = \{ E_{m1}^t, ..., E_{mT}^t \} \), where \( 1 \leq t \leq T \), through a CNN embedding module into our designed network.

In the following, we describe how to perform the 2D feature data embedding. Let a CNN embedding module be represented as \( E_{CNN}^{2D} \), and each 2D feature from \( \{ DS_T, RS_T, AS_T, ES_T \} \) is independently embedded by a different \( E_{CNN}^{2D} \). In this work, \( E_{CNN}^{2D} \) consists of a first convolution layer of \( 3 \times 1 \times 16 \), with stride of \( 3 \times 1 \), batch norm., ReLU, a max pooling layer with a kernel size of \( 3 \times 1 \), batch norm., ReLU, a second convolution layer by \( 3 \times 1 \times 16 \) with stride of \( 3 \times 1 \), batch norm., ReLU, a third convolution layer by \( 3 \times 1 \times 16 \) with stride of \( 3 \times 1 \), batch norm., ReLU, and an average pooling is finally used to extract all features. Note that the 2D feature data embedding \( E_{CNN}^{2D} \) aim to embed the spatial information from 2D features \( x_t = \{ DS_T, RS_T, AS_T, ES_T \} \), an example of \( E_{CNN}^{2D} \) can be seen in Figure 11.

The 2D feature data embedding operations performed through the CNN embedding module \( E_{CNN}^{2D} \) are given as follows.

\[
\begin{align*}
DS^T_E &= E_{CNN}^{2D}(DS_T), \\
RS^T_E &= E_{CNN}^{2D}(RS_T), \\
AS^T_E &= E_{CNN}^{2D}(AS_T), \\
ES^T_E &= E_{CNN}^{2D}(ES_T),
\end{align*}
\]

where \( DS^T_E, RS^T_E, AS^T_E, ES^T_E \) are the embedded features for the 2D original features \( DS_T, RS_T, AS_T, ES_T \), where the data size of each embedded feature is \( m \times T \).

Figure 11a gives an example of \( DS^T_E, RS^T_E, AS^T_E, ES^T_E \), which are embedded features for the 2D features \( DS_T, RS_T, AS_T, \) and \( ES_T \).

Another task of the multi-feature embedding phase is to embed the 3D features 
\( x_t = \{ RDS_T, RAS_T, RES_T \} \in S_{3D} \), and \( S = \{ x_1, ..., x_T \} \), where 
\( RDS_T = \{ RDM_{m1}^{t1}, ..., RDM_{m1}^{tT}, RDM_{mT}^{t1}, ..., RDM_{mT}^{tT} \} \), 
\( RAS_T = \{ RAM_{m1}^{t1}, ..., RAM_{m1}^{tT}, ..., RAM_{mT}^{t1}, ..., RAM_{mT}^{tT} \} \), and 
\( RES_T = \{ REM_{m1}^{t1}, ..., REM_{m1}^{tT}, ..., REM_{mT}^{t1}, ..., REM_{mT}^{tT} \} \), where \( 1 \leq t \leq T \), through the CNN network into our designed network. In the following, we describe how to perform the 3D feature data embedding. Let a CNN embedding module be represented as \( E_{CNN}^{3D} \), and each 3D feature from \( \{ RDS_T, RAS_T, RES_T \} \) is independently embedded by a different \( E_{CNN}^{3D} \). In
this work, $E_{3D}^{CNN}$ consists of a first convolution layer by $3 \times 3 \times 16$ with stride of $1 \times 1$, batch norm., ReLU, max pooling layer with kernel size of $2 \times 2$ with stride of $1 \times 1$, batch norm., ReLU, a second convolution layer by $3 \times 3 \times 16$ with a stride of $1 \times 1$, batch norm., ReLU, max pooling layer with kernel size of $2 \times 2$ with stride of $2 \times 2$; and then an average pooling is used to extract all the features. Finally, a Multilayer Perceptron (MLP) is additionally used to embed the features into the temporal space. Note that the purpose of the 3D feature data embedding $E_{3D}^{CNN}$ aims to first embed the 3D features into the spatial space, and MLP is then used to embed the 3D features into the spatial space. An example of $E_{3D}^{CNN}$ can be seen in Figure 12.

The 3D feature data embedding operations performed through the CNN embedding module $E_{3D}^{CNN}$ is given as follows:

$$
\begin{align*}
RDS_E &= E_{3D}^{CNN}(RDS_T), \\
RAS_E &= E_{3D}^{CNN}(RAS_T), \\
RES_E &= E_{3D}^{CNN}(RES_T),
\end{align*}
$$

(6)

where $RDS_E, RAS_E,$ and $RES_E$ are the embedded features for the 3D original features $RDS_T, RAS_T,$ and $RES_T$, where the data size of each embedded feature is $m \times c \times T$.

Figure 12a gives an example of $RDS_E, RAS_E,$ and $RES_E$, embedded features for the 3D features $RDS_T, RAS_T,$ and $RES_T$.

**Figure 12.** 3D Transformer-based encoder with (a) CNN embedding module, (b) multi-head attention module, and (c) fusion feature module.

4.3. Transformer-Based Encoding Phase

In the multi-feature embedding phase, the 2D embedded features $DS_E, RS_E, AS_E, ES_E,$ or the 3D embedded features $RDS_E, RAS_E,$ and $RES_E$ are embedded into our Transformer-based network, then we describe how to perform our Transformer-based encoding proce-
where \( F \) is the 2D embedded feature or different 3D embedded feature be embedded by the softmax function. The multi-head features transformation matrices, while the self-attention operation matrices with \( QW \) the self-attention weight matrices, the key and value of \( F \) follows the existing result from \[13\], which is to calculate the linear transformation of \( F \) the Transformer encoder for \( T \) the former encoder, \( T \) the 3D position embedding feature be ded features. Let the 2D position embedding feature be \( F \) function follows the existing position embedding function from \[13\], such that \( F \) the 3D position embedding feature, \( F \) operation to the 3D embedded features \( F \) embedded features. Let \( T \) be the 3D embedding feature after performing the \( F \) position embedding after performing the \( F \) transformation \( P \) are position embedding after performing the \( F \) are obtained using multi-head self-

\[
\begin{align*}
DS_p &= T_{pos}(DS_E) \\
RS_p &= T_{pos}(RS_E) \\
AS_p &= T_{pos}(AS_E) \\
ES_p &= T_{pos}(ES_E),
\end{align*}
\]

where \( DS_p \), \( RS_p \), \( AS_p \), and \( ES_p \) are position embedding after performing the \( T_{pos} \) function to the 2D embedded features \( DS_E \), \( RS_E \), \( AS_E \), and \( ES_E \), respectively. We denote the 2D position embedding feature as \( F_{2D_{pos}} \in \{DS_p, RS_p, AS_p, ES_p\} \).

Let \( T_{pos} \) be the position embedding function for the 3D embedded features \( RDS_E \), \( RAS_E \), and \( RES_E \), as follows.

\[
\begin{align*}
RDS_p &= T_{pos}(RDS_E) \\
RAS_p &= T_{pos}(RAS_E) \\
RES_p &= T_{pos}(RES_E),
\end{align*}
\]

where \( RDS_E \), \( RAS_E \), and \( RES_E \) are position embedded after performing the \( T_{pos} \) function to the 3D embedded features \( RDS_E \), \( RAS_E \), and \( RES_E \), respectively. We denote the 3D position embedding feature, \( F_{3D_{pos}} \in \{RDS_p, RAS_p, RES_p\} \). The \( T_{pos} \) function follows the existing position embedding function from \[13\], such that \( T_{pos} = \)

\[
\begin{align*}
P_{E_i \text{pos}, 2i} &= \sin(\text{pos}/10000^{2i/d_{\text{model}}}) \\
P_{E_i \text{pos}, 2i + 1} &= \cos(\text{pos}/10000^{2i/d_{\text{model}}}).
\end{align*}
\]

Next, we describe the Transformer-encoder operation by applying our 2D/3D embedded features. Let the 2D position embedding feature be \( F_{2D_{pos}} \in \{DS_p, RS_p, AS_p, ES_p\} \), and the 3D position embedding feature be \( F_{3D_{pos}} \in \{RDS_p, RAS_p, RES_p\} \). We then utilize a Transformer encoder, \( T_{Enc} \), for the 2D position embedding feature, \( F_{2D_{pos}} \in \{DS_p, RS_p, AS_p, ES_p\} \).

\[
F_{2D_{Enc}} = T_{Enc}(F_{2D_{pos}}),
\]

where \( F_{2D_{Enc}} \in \{RSE_{Enc}, RS_{Enc}, AS_{Enc}, RES_{Enc}\} \) is the 2D embedding feature after performing the Transformer encoder for \( F_{2D_{pos}} \in \{RS_p, RS_p, AS_p, RS_p\} \). Similarly, we then also utilize a Transformer encoder, \( T_{Enc} \), for the 3D position embedding feature, \( F_{3D_{pos}} \in \{RDS_p, RAS_p, RES_p\} \), as follows.

\[
F_{3D_{Enc}} = T_{Enc}(F_{3D_{pos}}),
\]

where \( F_{3D_{Enc}} \in \{RSE_{Enc}, RAS_{Enc}, RES_{Enc}\} \) is the 3D embedding feature after performing the Transformer encoder for \( F_{3D_{pos}} \in \{RDS_p, RAS_p, RES_p\} \). The Transformer encoder \( T_{Enc} \) follows the existing result from \[13\], which is to calculate the linear transformation of the self-attention weight matrices, the key and value of \( F_{2D_{pos}} \) (or \( F_{3D_{pos}} \)). Let \( Q \), \( K \), and \( V \) be the calculation matrices of \( F_{2D_{pos}} \) (or \( F_{3D_{pos}} \)), and let \( W^Q \), \( W^K \), and \( W^V \) be the linear transformation matrices, while the self-attention operation matrices with \( QW^Q \), \( KW^K \), and \( VW^V \) corresponding to \( F_{2D_{pos}} \) (or \( F_{3D_{pos}} \)), are obtained through the linear mapping, normalized by the softmax function. The multi-head features \( head_i \) are obtained using multi-head self-
We will describe how to generate these predicted label sequences from the CTC-decoder, Transformer-sequential-decoder, and Transformer-importance-decoder; each decoder provides a best predicted label sequence, which is nearer to an optimal solution. First, the sequence for the long-time series motion recognition; the CTC technique is integrated is then performed. As mentioned in Section 3.2, the connectionist temporal classification (CTC) [14] technique is utilized to overcome the blank segment and meaningless feature sequence for the long-time series motion recognition; the CTC technique is integrated into our scheme as the first one feature decoder. Let a set of predicted label sequences

\[
T_{Enc} = \begin{cases}
MHA(Q, K, V) = (head_1 \oplus head_2 \oplus \cdots \oplus head_d)W^O \\
head_i = S_A_i(QW_i^Q, KW_i^K, VW_i^V), i = 1, 2, \ldots, h \\
S_A_i(QW_i^Q, KW_i^K, VW_i^V) = softmax((QW_i^Q)(KW_i^K)T/V_{di}), i = 1, 2, \ldots, h
\end{cases}
\]

where \(S_A_i(QW_i^Q, KW_i^K, VW_i^V)\) is the feature signature from the linear transformation matrices \(W_i^Q, W_i^K, \) and \(W_i^V\), where \(1 \leq i \leq h\) and \(h\) is the weight between the features, after the dot product operation of the number of layers Q and K in the multi-head operation, and is divided by the variance value \(\sqrt{d_k}\) to obtain the weight matrix between 0 and 1; the weight is extracted through the normalization of the softmax function. Finally, the \(i\)-th self-attention \(head_i\) is obtained through multiplication by \(V\), and the multiple self-attention \(head_1, \ldots, head_i, \ldots, head_d\) are concatenated to produce the final self-attention output features.

In the following, we describe how to perform the feature fusion operation. After the Transformer-encoder operation, 2D position embedding feature, \(F_{pos}^{2D} \in \{DS_p, RSp, AS_p, ES_p\}\) and the 3D position embedding feature \(F_{pos}^{3D} \in \{RDS_p, RAS_p, RES_p\}\) are obtained from different Transformer encoders. All of these disjoint embedded features should be merged together before our proposed Transformer-based decoder. Let \(MF^{2D}\) be denoted as the merged 2D embedded features for all 2D position embedding features; we have \(MF^{2D} = DS_{Enc} \oplus RS_{Enc} \oplus AS_{Enc} \oplus ES_{Enc}\). Let \(MF^{3D}\) be denoted as the merged 3D embedded features for all 3D position embedding features, we have \(MF^{3D} = RDS_{Enc} \oplus RAS_{Enc} \oplus RES_{Enc}\). Finally, we further utilize the multi-layer perceptron MLP, and the average pooling layer AvgPool, for the feature training and feature extraction, as follows. \(FF^{2D} = MLP(Avgpool(MF^{2D}))\), \(FF^{3D} = MLP(Avgpool(MF^{2D}))\). Note that \(FF^{2D}\) or \(FF^{3D}\) are simplified as \(FF\) for the following explanations, where \(FF^{2D} \in \mathbb{R}^{m \times c}\) and \(FF^{3D} \in \mathbb{R}^{m \times c}\).

Examples of the position embedding of the 2D signature, \(DS_p\), and the 3D signature, \(RDS_p\), are given in Figures 11b and 12b. The Transformer-encoder operations are performed to have \(DS_{Enc}\) and \(RDS_{Enc}\), as shown in Figures 11b and 12b. The fusion features \(FF^{2D}\) and \(FF^{3D}\) are obtained in Figures 11c and 12c.

### 4.4. Transformer-Sequential-Decoding Phase

For the fused feature \(FF\), where \(FF \in FF^{2D}\) or \(FF^{3D}\) after performing the Transformer-based encoder phase, the proposed Transformer-sequential-decoding phase is then performed. The proposed Transfer-sequential-decoding phase consists of the CTC-decoder, Transformer-sequential-decoder, and Transformer-importance-decoder; each decoder produces a set of predicted label sequences, where \(\hat{Y} = (\hat{y}_1, \ldots, \hat{y}_l, \ldots)\), \(\bar{Y} = (\bar{y}_1, \ldots, \bar{y}_l, \ldots)\), and \(\hat{Y} = (\hat{y}_1, \ldots, \hat{y}_l, \ldots, \hat{y}_1)\) are sets of predicted label sequences of CTC-decoder, Transformer-sequential-decoder, and Transformer-importance-decoder, under the ground-truth label sequence \(Y = (y_1, \ldots, y_l, \ldots, y_1)\) for all \(y_l = (l_1^y, \ldots, l_j^y, \ldots, l_i^y)\) for \(x_i\) and \(S = \{x_1, \ldots, x_j, \ldots, x_i\}\). We will describe how to generate these predicted label sequences from the CTC-decoder, Transformer-sequential-decoder, and Transformer-importance-decoder. Finally, a joint-decoding operation is proposed to jointly combine these predicted label sequences to provide a best predicted label sequence, which is nearer to an optimal solution. First, the CTC decoder is described.

**CTC decoder:** The fused feature \(FF\), where \(FF \in FF^{2D}\) or \(FF^{3D}\), occurs after performing the Transformer-based encoder phase; the proposed Transfer-sequential-decoding phase is then performed. As mentioned in Section 3.2, the connectionist temporal classification (CTC) [14] technique is utilized to overcome the blank segment and meaningless feature sequence for the long-time series motion recognition; the CTC technique is integrated into our scheme as the first one feature decoder. Let a set of predicted label sequences
\[ \hat{Y} = (\hat{y}_1, ..., \hat{y}_t, ..., \hat{y}_t), \text{ for all } \hat{y}_i = (\tilde{l}_i^1, ..., \tilde{l}_i^y, ..., \tilde{l}_i^f) = \text{CTC}(FF) \text{ for } x_t, \text{ where the fused feature of } x_t \text{ is } FF \in \{2D, 3D\}. \] 

As shown in Figure 13c, the task of the CTC decoder is performed using the fusion feature FF to obtain \( \hat{Y} = (\hat{y}_1, ..., \hat{y}_t, ..., \hat{y}_t), \) and the loss \( \mathcal{L}_{CTC} \) can be calculated via Equation (1) under the predicted label sequence \( \hat{Y} \) and the ground-truth label sequence \( Y. \)

**Transformer-sequential-decoder:** To allow the continuous human motion to more effectively learn its temporal correlation, a second decoder, the Transformer-sequential-decoder, is built in our trained model. The output of the Transformer-sequential-decoder is sequentially decoding the Doppler, Doppler-angle, and Doppler-angle-range features.

The sequential operation of Transformer-sequential-decoder is represented by

\[
L_3^\text{Dec} = T^\text{Dec}_3(3^\text{Enc}(2^\text{Enc}(1^\text{Enc}(M,E),E),E),E),
\]

where \( L_1^\text{Enc} \in \{DS_\text{Enc}\}, L_2^\text{Enc} \in \{AS_\text{Enc}, ES_\text{Enc}\}, \) and \( L_3^\text{Enc} \in \{RS_\text{Enc}\} \) if 2D features are used; \( L_1^\text{Enc} \in \{RDS_\text{Enc}\} \) and \( L_2^\text{Enc} \in \{RAS_\text{Enc}, RE_\text{Enc}\} \), if 3D features are utilized. The additional weight, \( E \), is the Doppler feature enhancement matrix, which is obtained by \( E = L_3^\text{Enc} \times W^E \), where \( W^E \) is the transformation weight matrix \([22]\) and \( M \) is the label feature matrix, which is obtained by \( M = \text{MHA}(Y + Y^M) \), by performing the mask self-attention operation on the input label \( Y \) and mask matrix \( Y^M \) \([22]\) to prevent the mechanism of observing a subsequent time scale problem.

The sequential operation is stated below: The first function \( T^\text{Dec}_3(L_1^\text{Enc}, M) \) is initially performed, where \( T^\text{Dec} \) is same as \( T^\text{Enc} \) in Equation (11), if \( L_1^\text{Enc} \in \{DS_\text{Enc}\} \) or \( RDS_\text{Enc}. \) The output of \( L_1^\text{Dec} = T^\text{Dec}(L_1^\text{Enc}, M) \) is used again for the second function \( T^\text{Dec}_2(L_2^\text{Enc}, T^\text{Dec}(L_1^\text{Enc}, M), E) \) if \( L_2^\text{Enc} \in \{AS_\text{Enc}, ES_\text{Enc}\} \) or \( \{RAS_\text{Enc}, RE_\text{Enc}\} \) with the additional weight, \( E \). Finally, the output of \( L_2^\text{Dec} = T^\text{Dec}_3(L_3^\text{Enc}, T^\text{Dec}_2(L_2^\text{Enc}, M), E) \) is used again for the third function \( L_3^\text{Dec} = T^\text{Dec}(L_3^\text{Enc}, T^\text{Dec}_2(L_2^\text{Enc}, T^\text{Dec}_1(L_1^\text{Enc}, M), E), E), \) where \( L_1^\text{Dec}, L_2^\text{Dec}, \) and \( L_3^\text{Dec} \in \mathbb{R}^{m \times c}. \) We finally apply the MultiLayer Perceptron (MLP) to decode \( \hat{Y} = MLP(L_3^\text{Dec}), \) where \( \hat{Y} = (\hat{y}_1, ..., \hat{y}_t, ..., \hat{y}_t) \). The loss function \( \mathcal{L}_{Seq} \) is calculated by Equation (2). The detailed procedures of the Transformer-sequential-decoder module are proposed in Algorithm 1.

Our multi-head attention function, \( T^\text{Dec}_i, \) modified from \([14]\), is as follows.

\[
T^\text{Dec}_i = \begin{cases} 
\text{EMHA}(Q,K,V,E) = (\text{head}_1 \oplus \text{head}_2 \oplus \cdots \oplus \text{head}_h)W^O \\
\text{head}_i = \text{SA}_i(QW^Q_i, KW^K_i, VW^V_i), i = 1, 2, ..., h \\
\text{SA}_i(QW^Q_i, KW^K_i, VW^V_i) = \text{softmax}((QW^Q_i(EW^F_i)^T KW^K_i)^T)W^V_i, i = 1, 2, ..., h
\end{cases}
\]

Note that the key difference is that the weight calculation for self-attention is modified as \( QW^Q_i (EW^F_i)^T (KW^K_i)^T \), where \( W^E \) is the weight parameter set for the linear mapping, and \( E \) is the output feature of the decoder.

Our CHMR-HS architecture with the Transformer-sequential-decoder module, Transformer-importance-decoder module, CTC decoder, and Joint decoding module are illustrated in Figure 13a–d.

Figure 14 displays the detailed operation of the Transformer-sequential-decoder for sequentially decoding the Doppler, Doppler-angle, and Doppler-angle-range features.
Figure 13. Our CHMR-HS architecture with (a) a Transformer-sequential-decoder module, (b) a Transformer-importance-decoder module, (c) CTC decoder, and (d) joint decoding module.

Figure 14. Transformer-sequential-decoder for sequentially decoding the Doppler, Doppler-angle, and Doppler-angle-range features.
Algorithm 1 Transformer-sequential-decoder processing.

**Input:** \( L^1_{\text{Enc}} \in \{ DS_{\text{Enc}} \}, L^2_{\text{Enc}} \in \{ AS_{\text{Enc}}, ES_{\text{Enc}} \} \) and \( L^3_{\text{Enc}} \in \{ RS_{\text{Enc}} \) if 2D features are used, \( L^1_{\text{Enc}} \in \{ RDS_{\text{Enc}} \) and \( L^2_{\text{Enc}} \in \{ RAS_{\text{Enc}}, RES_{\text{Enc}} \), if 3D features are utilized, \( Y \) is ground truth label sequence.

**Output:** Transformer sequential decoder output \( \hat{Y} \), if \( L^1_{\text{Enc}} \in DS_{\text{Enc}} \) or \( RDS_{\text{Enc}} \), output is \( L^1_{\text{Dec}} \), if \( L^2_{\text{Enc}} \in AS_{\text{Enc}}, ES_{\text{Enc}} \) or \( RAS_{\text{Enc}}, RES_{\text{Enc}} \) with the additional weight, \( E \). Finally, the output is \( L^3_{\text{Dec}} \) and third output \( L^3_{\text{Dec}} \) is 2D signature only.

1. Given \( M \) is the label feature matrix
2. \( M = \text{MHA}(Y + Y^M) \)
3. Given \( h \) is the number of heads for multi-head attention \( \text{MHA} \)
4. \( h \leftarrow \) number of heads
5. Given \( k \) is the number of heads for enhance multi-head attention \( \text{EMHA} \)
6. \( k \leftarrow \) number of heads
7. Given \( m \) is the number of heads for enhance multi-head attention \( \text{EMHA} \)
8. \( m \leftarrow \) number of heads
9. Perform range attention
10. \( \text{foreach} \ i = 1 \text{ to } m \) do
11. Perform angle attention
12. \( \text{foreach} \ i = 1 \text{ to } k \) do
13. Perform Doppler attention
14. \( \text{foreach} \ i = 1 \text{ to } h \) do
15. Calculate self-attention weights
16. \( Q^D_i \leftarrow L^1_{\text{Enc}}, K^M_i, V^M_i \leftarrow M \)
17. Perform self-attention
18. \( \text{head}_i^D \leftarrow \text{SA}(Q^D_i, K^M_i, V^M_i) \)
19. Get Doppler decoder features
20. \( L^1_{\text{Dec}} = (\text{head}_1^D \oplus \cdots \oplus \text{head}_i^D \cdots \oplus \text{head}_h^D) \)
21. Calculate self-attention weights
22. \( Q^A_i \leftarrow L^2_{\text{Enc}}, K^D_i, V^D_i, E^D_i \leftarrow L^1_{\text{Dec}} \)
23. \( Q^F_i \leftarrow L^2_{\text{Enc}}, K^D_i, V^D_i, E^D_i \leftarrow L^1_{\text{Dec}} \)
24. Get angle decoder features
25. \( L^2_{\text{Dec}} = (\text{head}_1^A \oplus \cdots \oplus \text{head}_i^A \oplus \cdots \oplus \text{head}_h^A) \)
26. Perform self-attention
27. \( \text{head}_i^A \leftarrow \text{SA}(Q^D_i, K^D_i, V^D_i, E^D_i) \)
28. \( \text{head}_i^F \leftarrow \text{SA}(Q^F_i, K^D_i, V^D_i, E^D_i) \)
29. Calculate self-attention weights
30. \( Q^R_i \leftarrow L^1_{\text{Enc}}, K^A_i, V^A_i \leftarrow L^2_{\text{Dec}} \)
31. \( \text{head}_i^R \leftarrow \text{SA}(Q^R_i, K^A_i, V^A_i, E^D_i) \)
32. Get range decoder features
33. \( L^3_{\text{Dec}} = (\text{head}_1^R \oplus \cdots \oplus \text{head}_i^R \cdots \oplus \text{head}_m^R) \)
34. \( \hat{Y} = \text{MLP}(L^3_{\text{Dec}}) \)
35. **return** \( \hat{Y}, L^1_{\text{Dec}}, L^2_{\text{Dec}}, L^3_{\text{Dec}} \)

Transformer-importance-decoder: In the Transformer-sequential-decoder, the first output \( L^1_{\text{Dec}} = T_{\text{Dec}}(L^1_{\text{Enc}}, M) \) is to only decode the Doppler feature; the second output \( L^2_{\text{Dec}} = T^D_{\text{Dec}}(L^2_{\text{Enc}}, T_{\text{Dec}}(L^1_{\text{Enc}}, M), E) \) is to sequentially decode the Doppler and angle features. Finally, the third output \( L^3_{\text{Dec}} = T^E_{\text{Dec}}(L^3_{\text{Enc}}, T^E_{\text{Dec}}(L^2_{\text{Enc}}, T_{\text{Dec}}(L^1_{\text{Enc}}, M), E), E) \) is to sequentially decode the Doppler, angle and range features, where \( L^1_{\text{Dec}}, L^2_{\text{Dec}} \) and \( L^3_{\text{Dec}} \in R^{m \times c} \). In the following, we further design a third decoder, called the Transformer-sequential-decoder, to learn the selection from \( L^1_{\text{Dec}}, L^2_{\text{Dec}} \) or \( L^3_{\text{Dec}} \) among the fixed number of segments, where the fixed number of segment is \( \left\lceil \frac{c}{m} \right\rceil \). Note that the fused
feature $FF \in \mathbb{R}^{m \times c}$ is divided into $\left[ \frac{c}{m} \right]$ segments; each segment is an $m \times m$ matrix, and $\in \mathbb{R}^{m \times m}$. Each segment of these $\left[ \frac{c}{m} \right]$ segments should learn its importance to smartly select $L_{Dec}^1$, $L_{Dec}^2$, or $L_{Dec}^3$ as one of the inputs to perform the MHA function, where the multi-head self-attention (MHA) function is simplified as the MHA function, based on the expected value of distribution of the Doppler, angle, and range features.

First, we let $FF_{Dec} = MHA(FF, M)$ to perform the multi-head self-attention (MHA) function to the fused feature $FF$ and the label feature matrix $M$, where $FF \in FF^2D$ or $FF^3D$, $M$ is the label feature matrix, and $FF_{Dec} \in \mathbb{R}^{m \times c}$. To estimate the expected value of the distribution of Doppler, angle, and range features, we also calculate $D_{m,c} = MHA(L_{Dec}^1, FF_{Dec})$, $DA_{m,c} = MHA(L_{Dec}^2, FF_{Dec})$, and $DAR_{m,c} = MHA(L_{Dec}^3, FF_{Dec})$, where $DA_{m,c}$, $DAR_{m,c}$, $DA_{m,c}$ are performed MHA functions to $L_{Dec}^1$, $L_{Dec}^2$, and $L_{Dec}^3$ with $FF_{Dec}$, respectively. We also have the results of $D_{m,c}, DA_{m,c}, DAR_{m,c} \in \mathbb{R}^{m \times c}$, because $FF_{Dec}, L_{Dec}^1, L_{Dec}^2$ and $L_{Dec}^3 \in \mathbb{R}^{m \times c}$.

If $D_{m,c}, DA_{m,c}, DAR_{m,c} \in \mathbb{R}^{m \times c}$, we further divide each of $D_{m,c}, DA_{m,c}, DAR_{m,c}$ into $\left[ \frac{c}{m} \right]$ segments, as $D_{m,c} = \{d_{m,m},...,d_{m,m},...,d_{m,m}\}$, $DA_{m,c} = \{d_{m,m},...,d_{m,m},...,d_{m,m}\}$ and $DAR_{m,c} = \{d_{m,m},...,d_{m,m},...,d_{m,m}\}$, $1 \leq i \leq \left[ \frac{c}{m} \right]$, where the sub-relational features $d_{m,m}, d_{m,m}, d_{m,m} \in \mathbb{R}^{m \times m}$, where $1 \leq i \leq \left[ \frac{c}{m} \right]$.

To calculate the expected value of the distribution of the sub-relational features $d_{m,m}, d_{m,m}, d_{m,m}$ and $d_{m,m}$, where $1 \leq i \leq \left[ \frac{c}{m} \right]$, the eigenvectors $v_d, v_d$, and $v_d$, where $1 \leq i \leq \left[ \frac{c}{m} \right]$ are used to perform the determinant operation to obtain the directed area scalar of the eigenvector. In addition, the eigenvalue $\lambda_d, \lambda_d$ and $\lambda_d$, where $1 \leq i \leq \left[ \frac{c}{m} \right]$, are used to perform the operation of the expected value of the distribution. Therefore, we have

$$\left\{ \begin{array}{l}
\frac{d_{i,m}v_d^j}{\lambda_d} = \frac{\lambda_d}{\lambda_d}v_d^j, \\
\frac{d_{i,m}v_d^j}{\lambda_d} = \frac{\lambda_d}{\lambda_d}v_d^j, \\
\frac{d_{i,m}v_d^j}{\lambda_d} = \frac{\lambda_d}{\lambda_d}v_d^j,
\end{array} \right. \quad 1 \leq i \leq \left[ \frac{c}{m} \right].$$

Finally, let $\text{index}_i$ be denoted as the index of the maximum expected value of the $i$-th segment, where $1 \leq i \leq \left[ \frac{c}{m} \right]$; the sub-relational features of each segment may be $d_{m,m}, d_{m,m}, d_{m,m}$, and $d_{m,m}$ based on their maximum expected value of the distribution, where $1 \leq i \leq \left[ \frac{c}{m} \right]$. Consequently, we have $\text{index}_i = \max \left( \frac{\det(v_d)}{\lambda_d}, \frac{\det(v_d)}{\lambda_d}, \frac{\det(v_d)}{\lambda_d} \right)$, where $1 \leq i \leq \left[ \frac{c}{m} \right]$, by calculating the expected value of the distribution of $\frac{\det(v_d)}{\lambda_d}, \frac{\det(v_d)}{\lambda_d}$, and $\frac{\det(v_d)}{\lambda_d}$. If $\text{index}_i$ is equal to $\frac{\det(v_d)}{\lambda_d}$, then $i$-segment is adopted $d_{m,m}$ by only considering the Doppler feature, where $1 \leq i \leq \left[ \frac{c}{m} \right]$. If $\text{index}_i$ is equal to $\frac{\det(v_d)}{\lambda_d}$, then $i$-segment adopts the $d_{m,m}$ by considering both the Doppler and angle features, where $1 \leq i \leq \left[ \frac{c}{m} \right]$. If $\text{index}_i$ is equal to $\frac{\det(v_d)}{\lambda_d}$, then $i$-segment is adopted the $d_{m,m}$ by considering the Doppler, angle, and range features, where $1 \leq i \leq \left[ \frac{c}{m} \right]$. This work will be performed thorough the importance feature combination (IFC) operation, as defined below.

Then, the output of the Transformer-importance-decoder, $\hat{Y}$, can be represented, as given below: $\hat{Y} = \epsilon^l \{\text{IFC}(d_{m,m}, d_{m,m}, d_{m,m}, \text{index}_1) + \cdots + \text{IFC}(d_{m,m}, d_{m,m}, d_{m,m}, \text{index}_1) + \cdots + \text{IFC}(d_{m,m}, d_{m,m}, d_{m,m}, \text{index}_1) \}$, where $\epsilon^l$ is a parameter concatenation function to concatenate the training parameters by order through the features generated by iterative training, and updates the parameters obtained by the $t$-th decoding parameters from the previous $(t - 1)$-th decoding parameters, where $\epsilon^l = \epsilon^{l - 1} - \nabla L(Y, \hat{Y})$. In addition, the importance feature combination (IFC) function, defined as the sub-relation features $d_{m,m}, d_{m,m}$ or $d_{m,m}$ is selected based on the value
of index_i, where 1 ≤ i ≤ ⌈MT⌉. Figure 15 shows the example of the feature selection of the Transformer-importance-decoder. The loss function L\text{Imp}_i is calculated using Equation (3).

\[ \hat{Y} = \text{arg min}_{\hat{Y}} L_\text{CTC}(Y, \hat{Y}), \hat{Y} = \hat{Y} \]
\[ \hat{Y} = \hat{Y} \hat{Y} \]
\[ \hat{Y} = \hat{Y} \]

where \( L_\text{CTC}_i \), \( L_\text{Seq}_i \), and \( L_\text{Imp}_i \) are the final loss values of the CTC-decoder, Transformer-sequential-decoder, and Transformer-importance-decoder. After the obtaining the best sequence-to-sequence results of \( \hat{Y}_*, \hat{Y}_*, \hat{Y}_* \) of the CTC-decoder, Transformer-sequential-decoder, and Transformer-importance-decoder, the next purpose of the joint-decoding operation is to search for the best training parameters a, b, and \( \gamma \), as follows, to satisfy Equation (4).

\[ L_* = \text{arg min}_{a,b,\gamma} L_\text{CTC}_* + \beta L_\text{Seq}_* + \gamma L_\text{Imp}_* \]
\[ Y_* = a\hat{Y}_* + \beta\hat{Y}_* + \gamma\hat{Y}_* \]

subject to
\[ 0 \leq a \leq 1 \]
\[ 0 \leq b \leq 1 \]
\[ 0 \leq \gamma \leq 1 \]
\[ a + b + \gamma = 1 \]
5. Experimental Result

In this section, the experimental setup is explained in Section 5.1, the CHR dataset is described in Section 5.2, and the performance evaluation is discussed in Section 5.3.

5.1. Experimental Setup

The environment setup is described, including the model parameter settings for our experimental results. In our experiment, the AI GPU utilizes NVIDIA RTX 3090, and the AI framework adopts TensorFlow 2.8 [29]. The programming environment is the Python 3.8 version under Windows 10. The mmWave sensor evaluation board is used AWR 1642 BOOST ODS [25] by Texas Instruments, which is a single-chip 76-GHz to 81-GHz automotive radar sensor integrating DSP and MCU, and the obstacle detection sensor includes onboard-etched antennas for the four receivers and two transmitters [25]. We also use DCA1000EVM [30] as the real-time data-capture adapter for the radar sensing evaluation module, as shown in Figure 16.

Our work is to investigate the continuous human motions, which include many single human motions. In our experiment, all of the single human motions include the wave, jumping forward, falling, lying down, walking forward, walking backward, walking left, walking right, running forward, running backward, running left, and running right. We additionally consider three different high-similarity motion groups; the walking group, running group, and vertical-motion group, among these single human motions. The single human motion of the walking group includes walking forward, walking backward, walking left, and walking right. The single human motion of the running group includes running forward, running backward, running left, and running right. The single human motion of the vertical-motion group includes falling and lying down. Note that each pair of single human motions from the same high-similarity motion groups is more difficult to distinguish that that of each pair of single human motions from the different high-similarity motion groups. We will further investigate the effects for our designed CHMR-HR scheme.

To see the effects of the radar sensing images with the capability of time-variable length, four different lengths are considered; there are 3, 6, 9, and 12 s. To further investigate the impact of radar sensing images with the time-variable length, we denoted a variable-time ratio = (1:1:1:1) for the same amount of training data sets of 3, 6, 9, and 12 s. To further understand the impacts of different variable-time ratios, we also consider the different variable-time ratios = (2:1:2:1), (1:2:1:1), and (1:1:1:2) to increase the double amount of 3, 6, 9, and 12 s, respectively. In addition, each of the radar sensing images with time-variable lengths may randomly exist in one, two, three, or four different single human motions, in our experiment. The radar sensing data are collected using a mmWave sensor with onboard-etched MIMO antennas in a room with a length of four meters and a width of six meters, as shown in Figure 17. The experimental parameters are given in Table 1. To search for the best training parameters $\alpha$, $\beta$, and $\gamma$ for $L^* = \arg \min_{\alpha, \beta, \gamma \in [0, 1]} \alpha L_{CTC} + \beta L_{Seq} + \gamma L_{Imp}$, we denote the training parameters ratio as $(\alpha, \beta, \gamma)$, where $\alpha, \beta, \gamma \in [0, 1]$ and $\alpha + \beta + \gamma = 1$, in our experiment.
Figure 16. The hardware of mmWave sensor evaluation board and data-capture adapter.

Figure 17. Environment of the experimental setup with radar antennas.

Table 1. Experiment parameter.

| Radar                          | AWR 1642BOOST ODS |
|-------------------------------|-------------------|
| Start Frequency               | 77 Ghz            |
| End Frequency                 | 81 Ghz            |
| Number of Sample              | 120               |
| Number of Chirp               | 120               |
| Number of Tx antennas         | 2                 |
| Number of Rx antennas         | 4                 |
| Learning rate                 | 0.001             |
| Drop out                      | 0.5               |
| Number of encoder             | 6                 |
| Number of decoder             | 6                 |

5.2. The CHR Dataset

The dataset of continuous human motions for using AWR 1642 BOOST ODS and DCA1000EVM [25,30] is illustrated in Figure 17. We collected a training dataset of
1700 records: 500 records for the 3 s data, 426 records for the 6 s data, 416 records for the 9-second data, and 358 records for the 12 s data. Additionally, we collected a test dataset of 500 records, with 160 records for the 3 s data, 148 records for the 6 s data, 110 records for the 9 s data, and 82 records for the 12 s data. All these datasets were collected by 10 participants in our laboratory. There were five men and five women. They were 22–28 years old, 45–85 kg in weight, and 155–183 cm in height. To see the effect of data from the different backgrounds, we collected data from two different rooms with the different environments. We collected 80% raw data from a big-space room without a complicated background, and we also collected 20% raw data from the another small-space room with a more complicated background.

5.3. Performance Analysis

This subsection discussed the performance analysis. In our experimental, the real sample is the data that the trainer manually labels for each CHM radar data, the positive samples are the CHM radar data whose label is consistent with the real sample, and the negative sample are the CHM radar data whose label is inconsistent with the real sample.

We use the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) for our experiment, as follows. In our work, True Positive (TP) denoted that the predicted result and the labels of continuous motion in the positive sample are the same as the correct label prediction, in the positive samples of the CHM dataset. True Negative (TN) denoted that the predicted results and the continuous motions labels in the negative samples are both wrong label predictions in the negative samples of the CHM dataset. False Positive (FP) denoted that the predicted result is different from the wrong label in the negative sample; the model’s prediction is the result of the correct label in the negative samples of the CHM dataset. False Negative (FN) denoted that the predicted result is different from the correct label in the positive sample, and the model’s prediction is the result of the wrong label in the positive samples of the CHM dataset.

The performance metrics of our experimental to be observed are:

1. **Accuracy (AC)** is the value of 1 − L*, and $L^* = \arg\min_{a,\beta,\gamma \in [0,1]} aL^{*}_{CTC} + \beta L^{*}_{Seq} + \gamma L^{*}_{Imp}$, where $L^*_{CTC}$, $L^*_{Seq}$, and $L^*_{Imp}$ are the final loss values of CTC-decoder, Transformer-sequential-decoder, and Transformer-importance-decoder, under the real samples, including positive and negative samples, are considered for the test dataset.

2. **Recall (RE)** is True Positive (TP) divided by the sum of True Positive (TP) with False Negative (FN), i.e., $\frac{TP}{TP+FN}$.

3. **Precision (PC)** is True Positive (TP) divided by the sum of True Positive (TP) with False Positive (FP), i.e., $\frac{TP}{TP+FP}$.

4. **F1-score (F1)** is twice the Precision (PC) times Recall (RE) divided by the sum of Precision (PC) with Recall (RE), i.e., $\frac{2 \times PC \times RE}{PC + RE}$.

An efficient CHMR-HS scheme using mmWave radar sensing data is achieved with a high AC, high RE, high PC, and high F1. Efforts will be made in this paper to improve the AC, RE, PC, and F1.

5.3.1. Accuracy (AC)

Before discussing with the accuracy, we must determine the best training parameters $\alpha$, $\beta$, and $\gamma$ for $L^* = \arg\min_{a,\beta,\gamma \in [0,1]} aL^{*}_{CTC} + \beta L^{*}_{Seq} + \gamma L^{*}_{Imp}$ such that the training parameter ratio $(a, \beta, \gamma)$, where $a, \beta, \gamma \in [0,1]$ and $a + \beta + \gamma = 1$, can be determined. To show how to determine the best training parameter ratio $(a, \beta, \gamma)$, nine different training parameter ratios $(a, \beta, \gamma)$ are set as (0.8, 0.1, 0.1), (0.1, 0.8, 0.1), (0.1, 0.1, 0.8), (0.6, 0.2, 0.2), (0.2, 0.6, 0.2), (0.2, 0.6, 0.6), (0.4, 0.3, 0.3), (0.3, 0.4, 0.4), and (0.3, 0.3, 0.4). We apply these different values of $(a, \beta, \gamma)$ to calculate the accuracy, and we observed that an accuracy of 94.5% is obtained if $(a, \beta, \gamma) = (0.3, 0.4, 0.3)$, which is higher than those of other values.
of \((\alpha, \beta, \gamma)\) if 2D data are used, as shown in Figure 18a. Similarly, we also observed that an accuracy = 95.2% is obtained if \((\alpha, \beta, \gamma) = (0.3, 0.3, 0.4)\), which is higher than those of other values of \((\alpha, \beta, \gamma)\) if the 3D data are used, as shown in Figure 18b. Therefore, the best training parameters \((\alpha, \beta, \gamma) = (0.3, 0.4, 0.3)\) and \((\alpha, \beta, \gamma) = (0.3, 0.3, 0.4)\) are used for the 2D and 3D data, for the following discussions.

To make a fair comparison, the proposed CHMR-HS scheme for the 3/2D features (using red lines with square and circle marks), the proposed CHMR-HS scheme without CTC for 3/2D features (using red lines with cross-box and cross-circle marks), the CTC scheme for 3/2D features (using red lines with hollow-box and hollow-circle marks), and a CHMR scheme is proposed by Zhao et al. \[11\], simplified as the Zhao-I scheme (using a blue line). An end-to-end network for CHMR is proposed by Zhao et al. \[1\], simplified as the Zhao-II scheme (green line), and another CHMR scheme is presented by Shrestha et al. \[10\], simplified as the Shrestha scheme (orange line) are compared in our experiment.

![Graph](image_url)

Figure 18. The performance of accuracy if \((\alpha, \beta, \gamma) = (0.8, 0.1, 0.1), (0.1, 0.8, 0.1), (0.1, 0.1, 0.8), (0.6, 0.2, 0.2), (0.2, 0.6, 0.2), (0.2, 0.2, 0.6), (0.4, 0.3, 0.3), (0.3, 0.4, 0.4)\) and \((0.3, 0.5, 0.4)\), for (a) 2D data, and (b) 3D data.
The experimental results of accuracy (AC) vs. epoch are shown in Figure 19. As shown in Figure 19, we observed that the accuracy (AC) of the CHMR-HS scheme with 3D features (red line with square mark) = 95.2% > that of the CHMR-HS scheme with 2D features (red line with circle mark) = 94.5% > that of the CHMR-HS scheme without CTC for the 3D features (red line with cross-box mark) = 94.1% > that of the CHMR-HS scheme without CTC for 2D features (red line with cross-circle mark) = 93.4% > that of the Zhao-I scheme (blue line) = 93.1% > that of the Zhao-II scheme (green line) = 92.7% > that of the Shrestha scheme (orange line) = 92.1% > that of the CTC scheme for 3D features (red line with hollow-box mark) = 91.9% > that of the CTC scheme for 2D features (red line with hollow-circle mark) = 91.3% after the epoch is equal to 300. In addition, we also observed that the AC convergence rate of the CHMR-HS scheme with 3D features is lower than that of the CHMR-HS scheme with 2D features, but the final accuracy of the CHMR-HS scheme with 3D features is 95.2%, which is higher than that of the CHMR-HS scheme with 2D features, which is 94.5%.

To investigate the impacts of different number of features, we perform our CHMR-HR under a different number of 3D features and 2D features. As shown in Figure 20, we observed that the accuracy of the CHMR-HS scheme with RDS + RAS + RES = 95.2% > that of CHMR-HS scheme with DS + AS + ES + RS = 94.5% > that of CHMR-HS scheme with RDS = 93.6% > that of CHMR-HS scheme with DS + AS + RS = 94% > that of CHMR-HS scheme with DS = 92.9%. Basically, the more features that are utilized for training, the higher the accuracy will be.
Figure 20. Performance of accuracy (AC) vs. epoch under different number of features.

To investigate the ability of the proposed method to recognize different motions under different durations, we used 3 s, 6 s, 9 s, 12 s continuous motion samples directly collected in a real environment for testing. The example timeline of the proposed method and the compared methods is shown in Figure 21 for recognizing walking forward, walking right, running back, and falling, where the recognition results generated by each method are recorded along the time axis. It can be seen that misjudgments of CHMR-HS with the 3/2D features mostly occur at the time frames of motion transitions due to the use of the proposed Transformer-sequential-decoder and the Transformer-importance-decoder. Moreover, the Zhao-I, Zhao-II, and Shrestha schemes have high false recognition rates. This verifies that the proposed method illustrates that a better recognition performance is achieved.

Figure 21. Example timeline of different CHMR methods for recognizing walking forward, walking right, running back, and falling.

In the following Tables, bold and underlined numbers are denoted as as the first and second places of our simulation results. All motions denoted as the motions of CHMR randomly comes from all groups; the walking group, running group, and vertical-motion
group denoted as the motions of CHMR come from the walking group, running group, and vertical-motion group, respectively. As shown in Table 2, the accuracy of the CHMR-HS scheme with 3D features are always better than that of the CHMR-HS scheme with 2D features, except for the vertical-motion group. This is because that the classification of the vertical-motion group can be influenced by the oscillation of the characteristic frequency of 2D features, so that the learning model can more easily learn the difference of the acceleration gap between different motions in a highly similar motion.

Table 2. The comparison of accuracy (AC) for all motions and walking, running, and vertical-motion groups.

| Data Type | Models                                | All Motions | Walking Group | Running Group | Vertical-Motion Group |
|-----------|---------------------------------------|-------------|---------------|---------------|-----------------------|
| 2D Features | CHMR-HS (DS+AS+ES+RS)                | 94.5%       | 94.6%         | 92.9%         | 95%                   |
|           | CHMR-HS without CTC (DS+AS+ES+RS)    | 93.4%       | 84.5%         | 86.1%         | 89.3%                 |
|           | Zhao-I (DS)                           | 93.1%       | 82.5%         | 79.2%         | 88.2%                 |
|           | Zhao-II (DS)                          | 92.7%       | 83.34%        | 81.9%         | 83.57%                |
|           | Shrestha (DS)                         | 92.1%       | 79.3%         | 83.33%        | 83.56%                |
|           | CTC (DS+AS+ES+RS)                     | 91.3%       | 81.6%         | 83.5%         | 86.2%                 |
| 3D Features | CHMR-HS (RDS+RAS+RES)                | 95.2%       | 95.3%         | 93.8%         | 94.2%                 |
|           | CHMR-HS without CTC (RDS+RAS+RES)    | 94.1%       | 85.4%         | 88.5%         | 91.4%                 |
|           | CTC (RDS+RAS+RES)                     | 90.6%       | 81.9%         | 84.6%         | 87.3%                 |

5.3.2. Recall (RE)

The experimental results of recall (RE) vs. epoch are shown in Figure 22. As shown in Figure 22, we observed that the recall (RE) of the CHMR-HS scheme with 3D features (RDS + RAS + RES) = 93.7% > that of the CHMR-HS scheme with 2D features (DS + AS + ES + RS) = 92.2% > that of CHMR-HS scheme without CTC for 3D features (RDS + RAS + RES) = 91.9% > that of the CHMR-HS scheme without CTC for 2D features (DS + AS + ES + RS) = 91.2% > that of Zhao-I scheme (DS) = 90.9% > that of Zhao-II scheme (DS) = 90.1% > that of CTC scheme for 3D features (RDS + RAS + RES) = 87.9% > that of Shrestha scheme (DS) = 88.2% > that of CTC scheme for 2D features (DS + AS + ES + RS) = 87.9% if the epoch is 300. In addition, we also observed that the RE convergence rate of the CHMR-HS scheme with 3D features is lower than that of the CHMR-HS scheme with 2D features.

As shown in Table 3, the Recall (RE) of the CHMR-HS scheme with 3D features is also always better than that of the CHMR-HS scheme with 2D features, except for the vertical-motion group, due to the classification of vertical-motion group being more easily influenced by the oscillation of the characteristic frequencies of 2D features.

Table 3. The comparison of recall (RE) for all motions and walking, running, and vertical-motion groups.

| Data Type | Models                                | All Motions | Walking Group | Running Group | Vertical-Motion Group |
|-----------|---------------------------------------|-------------|---------------|---------------|-----------------------|
| 2D Features | CHMR-HS (DS+AS+ES+RS)                | 92.2%       | 93.1%         | 93.8%         | 94.2%                 |
|           | CHMR-HS without CTC (DS+AS+ES+RS)    | 91.2%       | 82%           | 83.5%         | 86.4%                 |
|           | Zhao-I (DS)                           | 90.6%       | 81%           | 77%           | 87.21%                |
|           | Zhao-II (DS)                          | 90.1%       | 82%           | 81.5%         | 83%                   |
|           | Shrestha (DS)                         | 88.2%       | 78.2%         | 83%           | 82.9%                 |
|           | CTC (DS+AS+ES+RS)                     | 87.9%       | 80.3%         | 84.2%         | 83.2%                 |
| 3D Features | CHMR-HS (RDS+RAS+RES)                | 93.7%       | 94.4%         | 93%           | 94.1%                 |
|           | CHMR-HS without CTC (RDS+RAS+RES)    | 91.9%       | 83.9%         | 88.5%         | 90.8%                 |
|           | CTC (RDS+RAS+RES)                     | 78.9%       | 79.2%         | 82.1%         | 85.2%                 |
5.3.3. Precision (PR)

The experimental results of precision (PR) vs. epoch are shown in Figure 23. As shown in Figure 23, we observed that the precision (PR) of the CHMR-HS scheme with 3D features (RDS + RAS + RES) = 84.8% > that of the CHMR-HS scheme without CTC for 3D features (RDS + RAS + RES) = 82.8% > that of CHMR-HS scheme with 2D features (DS + AS + ES + RS) = 81.9% > that of the Zhao-I scheme (DS) = 81.8% > that of the Zhao-II scheme (DS) = 79.1% > that of the CHMR-HS scheme without CTC for 2D features (DS + AS + ES + RS) = 78.5% > that of the CTC scheme for 3D features (RDS + RAS + RES) = 78.9% > that of the CTC scheme for 2D features (DS + AS + ES + RS) = 78.5% > that of the Shrestha scheme (DS) = 77.1% if the epoch is 300. In addition, we also observed that the RR convergence rate of the CHMR-HS scheme with 3D features is lower than that of CHMR-HS scheme with 2D features.

We observed that the performance of recall (RE) of our CHMR-HS scheme with 2D features is higher than that of the CHMR-HS scheme without CTC-decoder, but the precision (PC) of our CHMR-HS scheme with 2D features is lower than that of the CHMR-HS scheme without CTC, where RE is \( \frac{TP}{TP + FN} \), and PC is \( \frac{TP}{TP + FP} \). This is because the prediction of the False Positive (FP) of the CHMR-HS scheme without CTC-decoder will be better than the prediction of the False Negative (FN), due to the CTC-decoder being too optimistic to predict for the positive samples.

Figure 22. The performance of recall (RE) vs. epoch.
As shown in Table 4, the precision (PR) of the CHMR-HS scheme with 3D features are also always better than that of CHMR-HS scheme with 2D features, except for the vertical-motion group.

Table 4. The comparison of precision (PR) for all motions and walking, running, and vertical-motion groups.

| Data Type | Models | Precision | All Motions | Walking Group | Running Group | Vertical-Motion Group |
|-----------|--------|-----------|-------------|---------------|---------------|-----------------------|
| 2D Features | CHMR-HS (DS + AS + ES + RS) | 81.1% | 84.5% | 82.9% | 84% |
| 2D Features | CHMR-HS without CTC (DS + AS + ES + RS) | 81.8% | 73% | 70% | 77.2% |
| 2D Features | Zhao-I (DS) | 79.1% | 74% | 72% | 74% |
| 2D Features | Zhao-II (DS) | 77.1% | 80.3% | 74% | 72% |
| 2D Features | Shrestha (DS) | 78.5% | 73.9% | 74.3% | 74.2% |
| 3D Features | CHMR-HS (RDS + RAS + RES) | 84.8% | 86% | 84% | 83.2% |
| 3D Features | CHMR-HS without CTC (RDS + RAS + RES) | 82.8% | 73.1% | 79% | 82.3% |
| 3D Features | CTC (RDS + RAS + RES) | 81.4% | 70.1% | 73.2% | 77.9% |

5.3.4. F1-Score (F1)

The experimental results of the F1-score (F1) vs. epoch are shown in Figure 23. As shown in Figure 24, we observed that the F1-score (F1) of the CHMR-HS scheme with 3D features (RDS + RAS + RES) = 89% > that of the CHMR-HS scheme without CTC for 3D features (RDS + RAS + RES) = 87.1% > that of the CHMR-HS scheme with 2D features (DS + AS + ES + RS) = 86.7% > that of the Zhao-I scheme (DS) = 85% > that of the CHMR-HS scheme without CTC for 2D features (DS + AS + ES + RS) = 85.8% > that of the Zhao-II scheme (DS) = 84.2% > that of the CTC scheme for 3D features (RDS + RAS + RES) = 83.5% > that of the CTC scheme for 2D features (DS + AS + ES + RS) = 82.9% > that of the
Shrestha scheme (DS) = 82.2% if the epoch is 300. In addition, we also observed that the RR convergence rate of the CHMR-HS scheme with 3D features is lower than that of the CHMR-HS scheme with 2D features.

We observed that the performance of recall (RE) of our CHMR-HS scheme with 2D features is higher than that of the CHMR-HS scheme without CTC, but the F1-score (F1) of our CHMR-HS scheme with 2D features is also lower than that of CHMR-HS scheme without CTC, due to the precision (PC) of our CHMR-HS scheme with 2D features being lower than that of the CHMR-HS scheme without CTC, where the F1-score (F1) function is $\frac{2 \times PC \times RE}{PC + RE}$.

![Figure 24. The performance of F1-score (F1) vs. epoch.](image)

As shown in Table 5, the F1-score (F1) of the CHMR-HS scheme with 3D features are also always better than that of the CHMR-HS scheme with 2D features, except for the vertical-motion group.
Table 5. The comparison of F1-score (F1) for all motions and walking, running, and vertical-motion groups.

| Data Type | Models | All Motions | Walking Group | Running Group | Vertical-Motion Group |
|-----------|--------|-------------|---------------|---------------|-----------------------|
| 2D Features | CHMR-HS (DS+AS+ES+RS) | 86.7% | 88.5% | 87.4% | 88.8% |
| CHMR-HS without CTC (DS+AS+ES+RS) | 85.8% | 80.1% | 82.1% | 83.6% |
| Zhao-I (DS) | 85.9% | 75.4% | 70.8% | 75.2% |
| Zhao-II (DS) | 84.2% | 77.8% | 76.4% | 78.2% |
| Shrestha (DS) | 82.2% | 79.2% | 78.2% | 77% |
| CTC (DS+AS+ES+RS) | 82.9% | 75.2% | 81.3% | 86.7% |
| 3D Features | CHMR-HS (RDS+RAS+RES) | 89% | 90% | 88.2% | 89.3% |
| CHMR-HS without CTC (RDS+RAS+RES) | 87.1% | 78.3% | 83.4% | 86.3% |
| Zhao-I (DS) | 85.9% | 76.7% | 73.3% | 81.9% |
| Zhao-II (DS) | 84.2% | 77.8% | 76.4% | 78.2% |
| Shrestha (DS) | 82.2% | 79.2% | 78.2% | 77% |
| CTC (RDS+RAS+RES) | 82.9% | 75.2% | 81.3% | 86.7% |

5.3.5. Discussion on High Similarity

We investigate the ability of the proposed method to recognize different motions under different durations. We used 3 s, 6 s, 9 s, and 12 s continuous motion samples directly collected in a real environment for testing. To discuss the impact for the continuous human motions with high similarity, we try to independently increase in proportion for 3 s, 6 s, 9 s, 12 s; i.e., variable-time ratio = (1:1:1), (2:1:1), (1:2:1), (1:1:2), or (1:1:1:2) are considered.

Table 6 illustrates that the performance comparison of all motions and walking, running, and vertical-motion groups, if using all of the data of the training dataset. In Table 6, all of the performance results of accuracy, recall, precision, and F1-score of CHMR-HS with 3D features are better than those of CHMR-HS with 2D features, except for the vertical-motion group. The reason for this that the classification of the vertical-motion group can be more easily influenced by the oscillation of characteristic frequency if using 2D features.

Table 6. The performance comparison of all motions and walking, running, and vertical-motion groups, if using all data of the training dataset.

| Data Type | Models | All Motions | Walking Group | Running Group | Vertical-Motion Group |
|-----------|--------|-------------|---------------|---------------|-----------------------|
| 2D Features | CHMR-HS | 94.5% | 92.2% | 81.9% | 86.7% |
| CHMR-HS without CTC | 93.4% | 91.2% | 81.1% | 85.8% |
| Zhao-I | 93.1% | 91.0% | 80.9% | 85.9% |
| Zhao-II | 93.7% | 91.3% | 81.0% | 85.8% |
| Shrestha | 91.0% | 84.2% | 77.9% | 80.1% |
| CTC | 91.3% | 87.2% | 79.9% | 82.2% |
| 3D Features | CHMR-HS | 95.2% | 93.7% | 84.8% | 89.0% |
| CHMR-HS without CTC | 94.1% | 91.9% | 82.8% | 87.1% |
| Zhao-I | 93.1% | 90.7% | 81.7% | 85.4% |
| Zhao-II | 93.7% | 91.3% | 81.0% | 85.8% |
| Shrestha | 91.0% | 84.2% | 77.9% | 80.1% |
| CTC | 90.7% | 84.2% | 77.9% | 80.1% |

Table 7 illustrates that the performance comparisons of all motions under variable-time ratios = (2:1:1), (1:2:1), and (1:1:2). To consider all the motions, we observed that the performance results of accuracy, recall, precision, and F1-score of CHMR-HS with 2D features are better than that of CHMR-HS with 3D features if variable-time ratios = (2:1:1) or (1:2:1) are used. However, the performance results of accuracy, recall, precision, and F1-score of CHMR-HS with 3D features are better than that of CHMR-HS with 2D features if the variable-time ratios = (1:1:2), and (1:1:1:2) are used. This is because the 3D features require a longer data length to learn the dependencies of continuous motions.

Table 7. The performance comparison of all motions and walking, running, and vertical-motion groups, if using all data of the training dataset.

| Data Type | Models | All Motions | Walking Group | Running Group | Vertical-Motion Group |
|-----------|--------|-------------|---------------|---------------|-----------------------|
| 2D Features | CHMR-HS | 94.1% | 92.2% | 82.3% | 86.2% |
| CHMR-HS without CTC | 93.4% | 91.2% | 81.1% | 85.8% |
| Zhao-I | 93.7% | 91.3% | 81.0% | 85.8% |
| Zhao-II | 93.7% | 91.3% | 81.0% | 85.8% |
| Shrestha | 91.0% | 84.2% | 77.9% | 80.1% |
| CTC | 90.7% | 84.2% | 77.9% | 80.1% |
| 3D Features | CHMR-HS | 95.2% | 93.7% | 84.8% | 89.0% |
| CHMR-HS without CTC | 94.1% | 91.9% | 82.8% | 87.1% |
| Zhao-I | 93.1% | 90.7% | 81.7% | 85.4% |
| Zhao-II | 93.7% | 91.3% | 81.0% | 85.8% |
| Shrestha | 91.0% | 84.2% | 77.9% | 80.1% |
| CTC | 90.7% | 84.2% | 77.9% | 80.1% |

Sensors 2022, 22, 8409
36 of 40
Table 8 illustrates the performance comparisons of the walking group under variable-time ratios = (2:1:1:1), (1:2:1:1), (1:1:2:1), and (1:1:1:2). To consider the CHM from the walking group, we observed that the performance results of accuracy, recall, precision, and F1-score of CHMR-HS with 3D features are better than those of CHMR-HS with 2D features for all variable-time ratios = (2:1:1:1), (1:2:1:1), (1:1:2:1), or (1:1:1:2) used.

As compared with the performance result of the walking group in Table 6, we observed that the better performance results of the walking group are obtained if the variable-time ratios = (2:1:1:1) or (1:2:1:1) are used, but the lower performance results of walking group will occur if the variable-time ratios = (1:1:2:1) or (1:1:1:2) are used. This is because the training data under the variable-time ratios (2:1:1:1) and (1:2:1:1) are provided to learn the periodic characteristics of the micro-vibrations of the walking motions, such that the learning model can be strengthened to identify micro-differences between the walking motions through the feature performance of a short period of time of training data.

Table 8. The performance comparisons of walking group under variable-time ratios = (2:1:1:1), (1:2:1:1), (1:1:2:1), and (1:1:1:2).

| Data Type | Models    | Walking Group | Walking Group | Walking Group | Walking Group |
|-----------|-----------|----------------|----------------|----------------|----------------|
|           |           | 2D Features    | 3D Features    | 2D Features    | 3D Features    |
|           |           | Accuracy       | Recall         | Precision      | F1-Score       | Accuracy       | Recall         | Precision      | F1-Score       |
| CHMR-HS   | Zhao-I    | 94.8%          | 93.2%          | 85.5%          | 88.7%          | 94.8%          | 93.2%          | 85.5%          | 88.7%          |
|           | CHMR-HS   | 95.8%          | 94.5%          | 87.3%          | 88.4%          | 94.9%          | 93.6%          | 87.3%          | 88.4%          |
|           | CTC       | 95.6%          | 94.8%          | 87.3%          | 88.3%          | 94.4%          | 93.8%          | 87.3%          | 88.3%          |
|           | CHMR-HS   | 89%            | 88.5%          | 82%            | 85.5%          | 88%            | 88.5%          | 82%            | 85.5%          |
|           | Zhao-I    | 88.5%          | 88.5%          | 82%            | 85.5%          | 88%            | 88.5%          | 82%            | 85.5%          |
|           | CHMR-HS   | 89.5%          | 89%            | 84.5%          | 86.5%          | 89%            | 89%            | 84.5%          | 86.5%          |
|           | Zhao-I    | 88.5%          | 88.5%          | 82%            | 85.5%          | 88%            | 88.5%          | 82%            | 85.5%          |
|           | CHMR-HS   | 89.5%          | 89%            | 84.5%          | 86.5%          | 89%            | 89%            | 84.5%          | 86.5%          |
|           | Zhao-I    | 88.5%          | 88.5%          | 82%            | 85.5%          | 88%            | 88.5%          | 82%            | 85.5%          |
|           | CHMR-HS   | 89.5%          | 89%            | 84.5%          | 86.5%          | 89%            | 89%            | 84.5%          | 86.5%          |
|           | Zhao-I    | 88.5%          | 88.5%          | 82%            | 85.5%          | 88%            | 88.5%          | 82%            | 85.5%          |

Table 9 illustrates the performance comparisons of the running group under variable-time ratios = (2:1:1:1), (1:2:1:1), (1:1:2:1), and (1:1:1:2). To consider the CHM from the running group, we observed that the performance results of accuracy, recall, precision, and F1-score of CHMR-HS with 3D features are better than those of CHMR-HS with 2D features for all variable-time ratios = (2:1:1:1), (1:2:1:1), (1:1:2:1), or (1:1:1:2) are used.

Table 9. The performance comparisons of running group under variable-time ratios = (2:1:1:1), (1:2:1:1), (1:1:2:1), and (1:1:1:2).

| Data Type | Models    | Running Group | Running Group | Running Group | Running Group |
|-----------|-----------|----------------|----------------|----------------|----------------|
|           |           | 2D Features    | 3D Features    | 2D Features    | 3D Features    |
|           |           | Accuracy       | Recall         | Precision      | F1-Score       | Accuracy       | Recall         | Precision      | F1-Score       |
| CHMR-HS   | Zhao-I    | 93.5%          | 91.4%          | 80.5%          | 86.7%          | 93.5%          | 91.4%          | 80.5%          | 86.7%          |
|           | CHMR-HS   | 94%            | 93.6%          | 81%            | 87.3%          | 94.1%          | 93.6%          | 81%            | 87.3%          |
|           | CTC       | 95%            | 94.2%          | 82%            | 88.1%          | 95.1%          | 94.2%          | 82%            | 88.1%          |
|           | CHMR-HS   | 89%            | 89.5%          | 83%            | 88.5%          | 90%            | 89.5%          | 83%            | 88.5%          |
|           | Zhao-I    | 88.5%          | 88.5%          | 82%            | 85.5%          | 88%            | 88.5%          | 82%            | 85.5%          |
|           | CHMR-HS   | 90%            | 90%            | 84%            | 89%            | 91%            | 90%            | 84%            | 89%            |
|           | Zhao-I    | 88.5%          | 88.5%          | 82%            | 85.5%          | 88%            | 88.5%          | 82%            | 85.5%          |
|           | CHMR-HS   | 90%            | 90%            | 84%            | 89%            | 91%            | 90%            | 84%            | 89%            |
|           | Zhao-I    | 88.5%          | 88.5%          | 82%            | 85.5%          | 88%            | 88.5%          | 82%            | 85.5%          |
|           | CHMR-HS   | 90%            | 90%            | 84%            | 89%            | 91%            | 90%            | 84%            | 89%            |
|           | Zhao-I    | 88.5%          | 88.5%          | 82%            | 85.5%          | 88%            | 88.5%          | 82%            | 85.5%          |

As compared with the performance result of the running group in Table 6, we observed that the better performance results of the running group are obtained if variable-time ratios = (2:1:1:1) or (1:2:1:1) are utilized, but the lower performance results of the running group will be if variable-time ratios = (1:1:2:1) or (1:1:1:2) are utilized. This is because the training data under the variable-time ratios (2:1:1:1) and (1:2:1:1) are provided to learn the periodic characteristics of the micro-vibrations of the running motions, such that the learning model can be strengthened to identify micro-differences between running motions through the feature performance of a short period of time of training data.

Table 10 illustrates the performance comparisons of the vertical-motion group under variable-time ratios = (2:1:1:1), (1:2:1:1), (1:1:2:1), and (1:1:1:2). To consider the CHM from the vertical-motion group, we observed that the performance results of accuracy, recall, precision, and F1-score of CHMR-HS with 2D features are better than those of CHMR-HS with 3D features for all variable-time ratios = (2:1:1:1), (1:2:1:1), (1:1:2:1), or (1:1:1:2) are used.
Table 10. The performance comparisons of vertical-motion group under variable-time ratios = (2:1:1:1), (1:2:1:1), (1:1:2:1), and (1:1:1:2).

| Data Type | Models | Vertical-Motion Group |
|-----------|--------|-----------------------|
|           |        | (2:1:1:1) | (1:2:1:1) | (1:1:2:1) | (1:1:1:2) |
| 2D Features | CHMR-HS | 94.3% | 93.1% | 82.1% | 86.8% | 94.1% | 93.5% | 82.4% | 87.3% | 95.3% | 94.5% | 84.7% | 89.3% | 95.4% | 94.7% | 86.1% | 89.4% |
|           | CHMR-HS without CTC | 88.6% | 86% | 78.7% | 82.4% | 88.1% | 85.4% | 78.1% | 82.1% | 90% | 87.3% | 80% | 84% | 90.4% | 87% | 80.4% | 85.1% |
|           | Zhao-I | 87.1% | 85.4% | 79.1% | 82% | 80.4% | 78.4% | 79.1% | 77% | 86.6% | 83.2% | 78.3% | 80% | 86% | 83.1% | 79.1% | 78% | 80% |
|           | Shao et al | 83.1% | 81.5% | 75.2% | 78.1% | 81.2% | 78.1% | 75.2% | 75% | 84.6% | 81.4% | 75.1% | 78.2% | 81.1% | 79% | 78.1% | 78% |
|           | CTC | 85% | 83.4% | 77.2% | 77.2% | 84.8% | 83.7% | 77.2% | 77.2% | 84.8% | 83.7% | 77.2% | 77.2% | 84.8% | 83.7% | 77.2% | 77.2% |
| 3D Features | CHMR-HS | 93.8% | 93.5% | 82.1% | 87.3% | 93.3% | 93.2% | 81.8% | 87.2% | 94.5% | 94.4% | 84.6% | 89.2% | 94.6% | 95% | 84.2% | 89.9% |
|           | CHMR-HS without CTC | 91% | 89.9% | 81.3% | 85.1% | 90.4% | 89.1% | 81.2% | 85.1% | 92% | 91.5% | 83.5% | 87.6% | 92.3% | 91.2% | 84% | 87.8% |
|           | CTC | 86.2% | 84.1% | 76.2% | 79.2% | 86.2% | 84.2% | 75.9% | 79.2% | 86.2% | 84.2% | 75.9% | 79.2% | 86.2% | 84.2% | 75.9% | 79.2% |

As compared with the performance results of the vertical-motion group in Table 6, we observed that the better performance results of the vertical-motion group are obtained if variable-time ratios = (2:1:1:1) or (1:2:1:1) are utilized, but the lower performance results of the vertical-motion group will occur if variable-time ratios = (1:1:2:1) or (1:1:1:2) are utilized. This is because that the training data under the variable-time ratios (2:1:1:1) and (1:2:1:1) are provided to learn the periodic characteristics of the micro-vibrations of the vertical-motion, such that the learning model can be strengthened to identify micro-differences between vertical-motions through the feature performance of a short period of time of training data.

6. Conclusions

In this paper, we proposed a fully new CHMR-HS scheme for continuous human motion recognition using mmWave FMCW radar, to specially consider the continuous human motion recognition with a high-similarity problem. A Transformer-sequential-based learning mechanism is re-designed from the traditional Transformer-encoder–decoder architecture. We also adopt the 2D/3D features extracted from mmWave FMCW radar images as the input of our Transformer-sequential-based learning model. The experimental results show that the accuracies of our proposed CHMR-HS scheme are 95.2% and 94.5%, without considering the high-similarity condition, higher than those of existing CHMR schemes, and the accuracies of our proposed CHMR-HS scheme are 94.4% and 94.1%, especially considering the high-similarity condition. The results illustrate that our proposed CHMR-HS scheme can significantly improve the accuracy of continuous human motions, even for the high-similarity condition.

In future works, we will make research of the human motion recognition for multi-model cross learning, by combining the FMCW radar images with camera RGB images to increase the recognition accuracy. We also conduct further research toward continuous human motion recognition and vital signal measurements through FMCW radar for the multiple moving targets under deploying distributed radars.

Author Contributions: Conceptualization, Y.-S.C. and K.-H.C.; methodology, Y.-S.C., K.-H.C. and Y.-A.X.; software, Y.-A.X.; validation, Y.-S.C. and K.-H.C.; formal analysis, Y.-S.C. and Y.-A.X.; resources, Y.-S.C. and T.-Y.J.; data collection and experiments, Y.-A.X. and K.-H.C.; writing—original draft preparation, Y.-S.C. and Y.-A.X.; writing—review and editing, Y.-S.C. and K.-H.C.; visualization, Y.-A.X.; supervision, Y.-S.C. and T.-Y.J.; project administration, Y.-S.C.; funding acquisition, Y.-S.C. and T.-Y.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MOST 110-2221-E-305-001, MOST 109-2622-E-305-004, and MOST 109-2221-E-305-004-MY3.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

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

1. Zhao, R.; Ma, X.; Liu, X.; Liu, J. An End-to-End Network for Continuous Human Motion Recognition via Radar Radios. IEEE Sens. J. 2021, 21, 6487–6496. [CrossRef]

2. Zhang, Z.; Tian, Z.; Zhou, M. Latern: Dynamic Continuous Hand Gesture Recognition Using FMCW Radar Sensor. IEEE Sens. J. 2018, 18, 3278–3289. [CrossRef]

3. Yu, C.; Fang, S.-H.; Lin, L.; Chien, Y.-R.; Xu, Z. The Impact of Environmental Factors on mm-Wave Radar Point-Clouds for Human Activity Recognition. In Proceedings of the IEEE International Workshop on Electromagnetics: Applications and Student Innovation Competition (iWEM), Penghu, Taiwan, 26–28 August 2020.

4. Ahmed, S.; Park, J.; Cho, S.H. FMCW Radar Sensor Based Human Activity Recognition using Deep Learning. In Proceedings of the International Conference on Electronics Information and Communication (ICEIC), Jeju, Korea, 6–9 February 2022.

5. Wang, L.; Lu, Y. Classification and Identification of Micro-Doppler Signatures of Pedestrian and Cyclists Using Simulated Data and ELM. In Proceedings of the IEEE International Conference on Signal, Information and Data Processing, (ICSIDP), Chongqing, China, 11–13 December 2019.

6. Sun, Y.; Fei, T.; Schliep, F.; Pohl, N. Gesture Classification with Handcrafted Micro-Doppler Features using a FMCW Radar. In Proceedings of the IEEE MTT-S International Conference on Microwaves for Intelligent Mobility, (ICMIM), Munich, Germany, 15–17 April 2018.

7. Saeed, U.; Shah, S.Y.; Shah, S.A.; Ahmad, J.; Alotaibi, A.A.; Althobaiti, T.; Ramzan, N.; Alomainy, A.; Abbasi, Q.H. Discrete Human Activity Recognition and Fall Detection by Combining FMCW RADAR Data of Heterogeneous Environments for Independent Assistive Living. Electronics 2021, 10, 2237. [CrossRef]

8. Li, X.; He, Y.; Fioranelli, F.; Jing, X.; Yarovoy, A.; Yang, Y. Human Motion Recognition With Limited Radar Micro-Doppler Signatures. IEEE Trans. Geosci. Remote Sens. 2020, 59, 6586–6599. [CrossRef]

9. Sakagami, F.; Yamada, H.; Muramatsu, S. Accuracy Improvement of Human Motion Recognition with MW-FMCW Radar Using CNN. In Proceedings of the International Symposium on Antennas and Propagation (ISAP), Osaka, Japan, 25–28 January 2021.

10. Shrestha, A.; Li, H.; Kernec, J.L.; Fioranelli, F. Continuous Human Activity Classification From FMCW Radar with Bi-LSTM Networks. IEEE Sens. J. 2020, 20, 13607–13619. [CrossRef]

11. Zhao, R.; Ma, X.; Liu, X.; Li, F. Continuous Human Motion Recognition Using Micro-Doppler Signatures in the Scenario with Micro Motion Interference. IEEE Sens. J. 2020, 21, 5022–5034. [CrossRef]

12. Papanastasiou, V.S.; Trommel, R.P.; Harmanny, R.I.A.; Yarovoy, A. Deep Learning-based Identification of Human Gait by Radar Micro-Doppler Measurements. In Proceedings of the 17th European Radar Conference (EuRAD), Utrecht, The Netherlands, 13–15 January 2021.

13. Zheng, L.; Bai, J.; Zhu, X.; Huang, L.; Shan, C.; Wu, Q.; Zhang, L. Dynamic Hand Gesture Recognition in In-Vehicle Environment Based on FMCW Radar and Transformer. Sensors 2021, 21, 6368. [CrossRef] [PubMed]

14. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine learning (ICML), Pittsburgh, PA, USA, 25–29 June 2006.

15. Chen, V.C. Analysis of radar micro-Doppler with time-frequency transform. In Proceedings of the Tenth IEEE Workshop on Statistical Signal and Array Processing, Pocono Manor, PA, USA, 16 August 2000.

16. Ash, M.; Ritchie, M.; Chetty, K. On the Application of Digital Moving Target Indication Techniques to Short-Range FMCW Radar Data. IEEE Sens. J. 2018, 18, 4167–4175. [CrossRef]

17. Wang, J. CFAR-Based Interference Mitigation for FMCW Automotive Radar Systems. IEEE Trans. Intell. Transp. Syst. 2021, 23, 1–10. [CrossRef]

18. Kim, Y.; Moon, T. Human Detection and Activity Classification Based on Micro-Doppler Signatures Using Deep Convolutional Neural Networks. IEEE Geosci. Remote Sens. Lett. 2015, 13, 8–12. [CrossRef]

19. Khalid, H.; Gorji, A.; Bourdoux, A.; Pollin, S.; Sahli, H. Multi-View CNN-LSTM Architecture for Radar-Based Human Activity Recognition. IEEE Access 2020, 10, 24509–24519. [CrossRef]

20. Guendel, R.G.; Unterhorst, M.; Gambi, E.; Fioranelli, F.; Yarovoy, A. Continuous Human Activity Recognition for Arbitrary Directions with Distributed Radars. In Proceedings of the IEEE Radar Conference (RadarConf21), Atlanta, GA, USA, 7–14 May 2021.

21. Ding, C.; Hong, H.; Zou, Y.; Chu, H.; Zhu, X.; Fioranelli, F.; Kernec, J.L.; Li, C. Continuous Human Motion Recognition With a Dynamic Range-Doppler Trajectory Method Based on FMCW Radar. IEEE Trans. Geosci. Remote Sens. 2019, 57, 6821–6831. [CrossRef]

22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. arXiv 2017, arXiv:1706.03762.

23. Hu, R.; Singh, A. UniT: Multimodal Multitask Learning with a Unified Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 10–17 October 2021.

24. Lohrenz, T.; Li, Z.; Fingscheidt, T. Multi-Encoder Learning and Stream Fusion for Transformer-Based End-to-End Automatic Speech Recognition. arXiv 2021, arXiv:2104.00120.

25. AWR1642 Obstacle Detection Sensor with Wide Field-of-View (FOV) Antenna Evaluation Module. Available online: https://www.ti.com/tool/AWR1642BOOST-ODS (accessed on 7 October 2022).
26. Polat, G.; Ergenc, I.; Kani, H.T.; Alahdab, Y.O.; Atug, O.; Temizel, A. Class Distance Weighted Cross-Entropy Loss for Ulcerative Colitis Severity Estimation. *arXiv* 2022, *arXiv*:2202.05167.

27. Liu, C.; Gonzalez, H.A.; Vogginger, B.; Mayr, C.G. Phase-based Doppler Disambiguation in TDM and BPM MIMO FMCW Radars. In *Proceedings of the IEEE Radio and Wireless Symposium (RWS)*, San Diego, CA, USA, 17–22 January 2021.

28. Kılıç, B.; Arıkan, O. Capon’s Beamformer and Minimum Mean Square Error Beamforming Techniques in Direction of Arrival Estimation. In *Proceedings of the 29th IEEE Signal Processing and Communications Applications Conference (SIU)*, Istanbul, Turkey, 9–11 June 2021.

29. Create Production-Grade Machine Learning Models with TensorFlow. Available online: [https://www.tensorflow.org/?hl=zh-tw](https://www.tensorflow.org/?hl=zh-tw) (accessed on 7 October 2022).

30. Real-Time Data-Capture Adapter for Radar Sensing Evaluation Module. Available online: [https://www.ti.com/tool/DCA1000EVM](https://www.ti.com/tool/DCA1000EVM) (accessed on 7 October 2022).