ICT Express 12 (2026) 512–516

Contents lists available at ScienceDirect

ICT Express

journal homepage: www.elsevier.com/locate/icte

Activity recognition and user identification using mmWave radar with a shared-backbone graph network and task-specific heads

Jun Yong Eom, Daewon Seo *

Department of Electrical Engineering and Computer Science, DGIST, Daegu, South Korea

ARTICLE INFO

Keywords:
Activity recognition
User identification
Multi-task model
Graph neural network
MmWave radar

ABSTRACT

Identity-aware activity recognition is a key enabler for customized services. However, joint modeling of activity recognition and user identification from wireless signals remains underexplored. This work presents a dual-task graph model for millimeter-wave (mmWave) frequency-modulated continuous-wave (FMCW) radar point-cloud sequences. We construct directed graphs that capture a user's spatial structure and motion over time. A shared graph neural backbone processes these graphs and produces node embeddings that encode local spatial features and short-term dynamics. Each task-specific head first aggregates node embeddings into a graph-level representation and then performs activity or identity classification. Experiments on two public datasets demonstrate that the proposed scheme achieves classification performance comparable to single-task baselines for both activity recognition and user identification while maintaining low-latency inference. Codes are available at https://github.com/junyongeom/mmActId/.

1. Introduction

Activity recognition is a fundamental capability for systems designed to monitor human behavior continuously and unobtrusively across healthcare [1], eldercare [2], public safety [3], and smart facilities [4], where both highly reliable and low-latency inference is essential. By interpreting everyday motion in real time, it supports fall and anomaly detection [5], occupancy monitoring and safety management [6], and ambient human–computer interaction [7]. These applications often run on edge devices with limited computation and energy budgets.

Another functionality that such systems often need to support is the user identification, which attributes actions to specific individuals. Linking actions to actors enables tailored feedback and more personalized support. A unified solution must preserve privacy, remain robust to lighting changes, and meet strict latency and memory budgets, since running separate models for recognition and identification increases delay and memory traffic, especially on low-end edge hardware used in real-time fall detection, occupancy monitoring, and in-vehicle driver/occupant monitoring for rapid assistance.

However, in literature, joint modeling of activity recognition and user identification remains underexplored, with only a few studies attempting to connect the two tasks. Shrestha et al. [8] used wearable sensors with deep learning models to classify both activity and user's identity. Cao et al. [9] used channel state information (CSI) from Wi-Fi devices and proposed a dual-task deep learning network to recognize gestures and identify users. Yu et al. [10] showed that millimeter-wave (mmWave) radar point clouds are effective for activity recognition. Building on point-cloud-based mmWave sensing, Xu et al. [11] proposed a network to jointly perform gesture recognition and user identification.

Despite these efforts, existing systems do not fully benefit from a dual-task design. In most cases, only input-level preprocessing of sensing signals is shared, while the inference stage is naïvely split into task-specific feature extractors and classifiers. This duplication increases model sizes, leading to higher memory and power requirements as well as longer end-to-end delays. As a result, the practical advantages that motivate dual-task modeling are not realized, especially on resource-constrained edge platforms.

We address these limitations with a shared-backbone graph network on mmWave FMCW radar. The radar is compact and cost effective, robust to illumination, and privacy preserving while producing real time 3D point clouds. From point cloud sequences, we construct directed spatio-temporal graphs that follow the time order and preserve fine-grained motion cues [12]. Nodes correspond to detected points in each frame, and directed edges connect the nodes to points in the next frame. Edge weights depend on Euclidean distance, with closer pairs receiving stronger connections. The representation keeps variable point counts without voxelization or heavy resampling and tolerates missing detections within a data sample.

* Corresponding author.
E-mail addresses: junyongeom@dgist.ac.kr (J.Y. Eom), dwseo@dgist.ac.kr (D. Seo).

https://doi.org/10.1016/j.icte.2026.02.003
Received 19 September 2025; Received in revised form 29 December 2025; Accepted 4 February 2026
Available online 5 February 2026

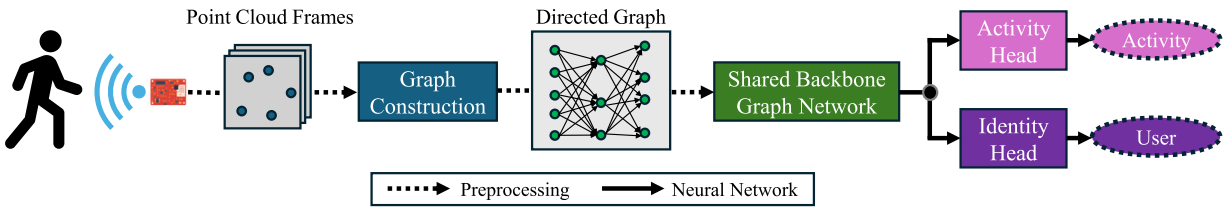

Fig. 1. System Overview.

In the proposed dual-task model, a single encoder processes the directed graph once and produces node embeddings that capture local geometry and short-term motion. Two lightweight task-specific heads then read the shared embeddings: The activity head uses a gated aggregation with a normalized half-sine position so that the model can emphasize informative frames. The identity head aggregates distributional structure with stable statistics and a compact kernel-based summary built with random Fourier features. Both heads are lightweight and operate on the same updated node features, which highly reduces the number of model parameters, and hence shortens memory usage and latency.

The model is trained in an end-to-end manner. The heads and the backbone are trained with task losses and a joint objective, respectively, concentrating computation in a single encoder and improving sample efficiency compared to separate pipelines. At inference stage, a single pass through the backbone and the two heads produces activity and identity decisions with low per-sample latency. Such a structure is well-suited for edge devices, where both memory and computational resources are limited. In experiments on public datasets, the model demonstrates accuracy on par with independently-trained task-specific models, while reducing inference time.

The main contributions are summarized as follows.

- We design a shared-backbone graph network trained end-to-end with a joint loss on the backbone and task-specific losses on the heads, which reduces the number of parameters and enables single-pass and low-latency inference.
- We introduce task-specific graph readouts where the user identification head emphasizes spatial topology and the activity recognition head emphasizes temporal evolution, both operating on shared node embeddings with lightweight classifiers.

2. Methodology

In this section, we describe the proposed identity-aware activity recognition system in detail. Fig. 1 summarizes the pipeline from mmWave point clouds to the joint predictions produced by a shared backbone graph network with task-specific heads. Given a sequence of point cloud frames, we first construct a trajectory-aware directed graph. A shared graph neural network then processes this graph and encodes every node into an embedding that captures local spatial features and short-term motion. Two lightweight task-specific heads apply tailored graph readouts to the shared node embeddings and map the readouts to identity and activity labels, respectively.

2.1. Input and preprocessing

2.1.1. Point cloud

mmWave FMCW radar estimates 3D object coordinates in real time [13]. It transmits chirps and mixes the received echoes with the transmitted chirps to obtain intermediate frequency (IF) signals. Then, the range is obtained from the intermediate frequency spectrum. Radial velocity is inferred from phase changes across consecutive chirps. Angles of arrival, azimuth and elevation, are derived from phase differences across antennas. These measurements are converted to 3D Cartesian coordinates for each object.

2.1.2. Graph construction

We build a directed spatio-temporal graph from input point cloud frames based on [12]. Let $\mathcal{P} = \{P_t\}_{t=1}^{T}$, where T is the number of frames and is assumed $T \geq 3$. Frame t contains N_t points with coordinates $\mathbf{p}_i^t = (x_i^t, y_i^t, z_i^t)$, $1 \leq i \leq N_t$. We create one node v_i^t for each point. For $1 \leq t < T$, we connect every node in frame t to every node in frame $t+1$ and denote the edge set by $E = \{(v_i^t, v_j^{t+1}) \mid 1 \leq i \leq N_t,\ 1 \leq j \leq N_{t+1}\}$.

Edge weights decay according to spatial separation. For an edge (v_i^t, v_j^{t+1}), we compute $d_{ij}^{(t,t+1)} = \left\|\mathbf{p}_i^t - \mathbf{p}_j^{t+1}\right\|_2$ and assign $w_{ij}^{(t,t+1)} = \exp\left(-\beta\, d_{ij}^{(t,t+1)}\right)$, where $\beta > 0$ is a scaling factor. Larger β results in sharper emphasis on closer pairs while smaller β spreads the edge weights more broadly. Note that this construction preserves temporal order and does not require voxelization. It keeps the variable number of points per frame.

The node feature is $\mathbf{x}_i^t = [x_i^t, y_i^t, z_i^t, \phi(t)]$, where $\phi(t) = \sin\left(\frac{\pi}{2} \cdot \frac{t-1}{T-1}\right)$ encodes the frame index as a half sine function and adds point appearance time and sequence progression to each node. The half sine is normalized across the data sample so that the index ranges from zero to the last frame. This gives a consistent time scale across data samples with different lengths.

2.2. Dual-task architecture

2.2.1. Shared graph backbone

The backbone updates node embeddings by aggregating information from directed temporal neighborhoods so that each node encodes local geometry and short-term motion [12]. The input multilayer perceptron (MLP) maps each node feature $\mathbf{x}_i^t \in \mathbb{R}^4$ to a D dimensional latent vector $\mathbf{h}_i^{(0)}$. We then apply L message passing layers that aggregate neighbors and update embeddings with a residual path and dropout during training.

Let $\mathcal{N}_{\text{in}}(i)$ be the set of nodes that have directed edges into node i from the previous frame. At layer l, the incoming message and the update are

$$\mathbf{m}_i^{(l)} = \sum_{j \in \mathcal{N}_{\text{in}}(i)} w_{ji}\mathbf{h}_j^{(l-1)}, \tag{1}$$

$$\mathbf{h}_i^{(l)} = \text{BN}^{(l)}\left(\text{ReLU}\left(\text{MLP}^{(l)}\left(\mathbf{h}_i^{(l-1)} + \mathbf{m}_i^{(l)}\right)\right)\right). \tag{2}$$

Here, ReLU [14] is the rectified linear unit, and $\text{BN}^{(l)}$ denotes batch normalization at layer l. The residual term $\mathbf{h}_i^{(l-1)}$ improves stability.

2.2.2. Task-specific heads

Activity specific readout

As actions change over time, the information of actions in data is unequally distributed over frames. We combine content with a half-sine positional encoding, which provides a smooth, bounded temporal weighting across frames and helps the model emphasize informative frames. Let a graph have N nodes with embeddings $\{h_i\}_{i=1}^{N}$ and frame encodings $\{\phi(t_i)\}_{i=1}^{N}$. We compute a score for each node

$$s_i = u^{\top} \tanh\left(W_h h_i + w_t\, \phi(t_i)\right). \tag{3}$$

Scores are then converted to coefficients with a softmax over the N nodes

$$\alpha_i = \frac{\exp(s_i)}{\sum_{k=1}^{N} \exp(s_k)}. \tag{4}$$

Finally, the representation of action is a weighted sum of node embeddings

$$r^{\text{act}} = \sum_{i=1}^{N} \alpha_i \, h_i. \tag{5}$$

The half sine $\phi(t)$ is normalized within each sample from zero at the first frame to one at the last frame so that the coefficients are comparable across different frame lengths. The parameters w_t, W_h, and u control the effect of position and content in a data-driven manner. To be specific, when $|w_t|$ is small, the coefficients are dominated by the term $W_h h_i$. If $w_t > 0$, increasing $|w_t|$ assigns larger coefficients to later frames. If $w_t < 0$, increasing $|w_t|$ assigns larger coefficients to earlier frames. Computing the softmax within the sample makes $\sum_i \alpha_i = 1$, which provides size invariance when N changes and keeps the scale of r^{act} stable across data samples. In large graphs, we optionally keep only the top $K = \lceil \rho N \rceil$ scores before the softmax with $0 < \rho \leq 1$ to reduce noise and cost while preserving the definition. For robustness on noisy samples, we may concatenate r^{act} with simple statistics such as mean and max pooling and pass them through a light projection before the classifier. The overall cost of the readout is $O(ND)$ for N nodes and embedding size D.

Identity specific readout

A user can be identified based on how embeddings are distributed across the sequence. Given N node embeddings, $\{h_i\}_{i=1}^{N}$ we compute the mean and the standard deviation

$$\mu = \frac{1}{N}\sum_{i=1}^{N} h_i, \quad \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(h_i - \mu)^2 + \varepsilon}. \tag{6}$$

Here $\varepsilon > 0$ is a small constant to prevent exploding gradients when the variance approaches zero. We also build a kernel mean summary using random Fourier features [15,16]

$$z(h) = \sqrt{\tfrac{2}{m}} \cos\big(W^{\top} h + b\big), \quad \bar{z} = \frac{1}{N}\sum_{i=1}^{N} z(h_i). \tag{7}$$

The identity representation is

$$r^{\text{id}} = \psi\big([\ \mu \parallel \sigma \parallel \bar{z}\ ]\big). \tag{8}$$

where ψ is a projection layer that maps the concatenated statistics to a fixed-size vector as the input to the subsequent prediction layer.

The vectors μ and σ respectively summarize first- and second-order structures across nodes and are robust against changes in the number of nodes. The term $\bar{z}$ adds nonlinear distribution information through a shift-invariant kernel approximation and separates cases that share similar means and variances but differ in higher order structure. The matrix $W \in \mathbb{R}^{D\times m}$ and the phase $b \in [0, 2\pi)^m$ can be fixed for a stable baseline or learned so the effective kernel adapts to the dataset. The dimension m is the number of random Fourier features and equals the output dimension of $z(h)$. A larger m captures finer distribution details at higher cost. All operations are averages over nodes which makes the representation permutation invariant and stable when N changes. A light normalization on r^{id} before the classifier improves conditioning when batches are small. The cost is $\mathcal{O}(ND)$ for statistics and $\mathcal{O}(NDm)$ for random features.

Prediction layers

Each readout is followed by a small multilayer perceptron that outputs logits and softmax probabilities

$$p_{\text{act}} = \text{softmax}\big(f_{\text{act}}(r^{\text{act}})\big), \qquad p_{\text{id}} = \text{softmax}\big(f_{\text{id}}(r^{\text{id}})\big).$$

2.3. Training and inference

2.3.1. Objective

The objective is to learn activity recognition and user identification from the same backbone. We train the shared backbone and the two heads together in an end-to-end manner.

Let p_{act} and p_{id} be the softmax probabilities from the two classifiers. For a mini batch of B graphs with activity labels $y_{\text{act}}^{(g)}$ and identity labels $y_{\text{id}}^{(g)}$, the task losses are

$$\mathcal{L}_{\text{act}} = \frac{1}{B}\sum_{g=1}^{B} -\log p_{\text{act}}^{(g)}, \quad \mathcal{L}_{\text{id}} = \frac{1}{B}\sum_{g=1}^{B} -\log p_{\text{id}}^{(g)}. \tag{9}$$

Then, the joint objective is

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \lambda \, \mathcal{L}_{\text{id}}, \tag{10}$$

where λ is a weighting factor. Each head comprising readout and classifier is optimized by its own task loss. The shared backbone is optimized by the joint loss in (10). Gradients from both tasks update the backbone which encourages features that transfer across tasks.

2.3.2. Inference

A single forward pass computes the two predictions at the same time. We construct the directed graph and then run the shared-backbone network to obtain node embeddings. The activity head assigns attention coefficients to the nodes and aggregates them into r^{act} before producing activity logits. The identity head computes $(\mu, \sigma, \bar{z})$ then forms r^{id} and produces identity logits.

3. Evaluation

3.1. Experiment setup

1. **Dataset** We use the MiliPoint [17] and Pantomime [18] datasets for evaluation. MiliPoint contains recordings from 11 users performing 49 activities in an indoor environment, from which we use a total of 29,400 samples. Pantomime includes 41 users performing 21 gestures in two environments (Office and Open), and we use 9010 samples. For both datasets, we report results using 5-fold cross-validation.
2. **Baselines** We adopt four point cloud-based deep learning baselines designed for efficient point cloud processing. All baselines receive the same node features $(x, y, z, \phi(t))$ and are trained with the same protocol and hyperparameters described in the methodology section to control the capacity and optimization effects.
   (a) **P+G** We combine PointNet++ [19] with a gated recurrent unit (GRU) [20], leveraging a lightweight PointNet-style encoder to reduce the number of floating-point operations (FLOPs).
   (b) **D+G** We combine DGCNN [21] with a GRU. DGCNN captures local geometric relations through dynamic edge convolutions and the GRU models temporal dynamics.
   (c) **FastHAR** FastHAR [22] is designed for computationally efficient activity recognition from point clouds.
   (d) **PTV3** Point Transformer V3 (PTV3) [23] is a Transformer-based model that can handle variable-size point clouds.

   Baselines that require a fixed number of points per frame are resampled based on sampling procedure in [24].

3.2. Experimental results

1. **Computational Complexity**
   Table 1 summarizes latency and computational cost. *Pre.* denotes preprocessing time from point clouds to the model input. *ACT* and *ID* denote inference latency for activity and identity recognition, respectively. *Tot.* is the end-to-end latency, computed as *Pre.*+*ACT*+*ID*. FLOPs denote the floating-point operations of the neural network forward pass required to produce both activity and identity outputs (excluding preprocessing). Experiments are

Table 1
Computational complexity comparison (latency (ms) and FLOPs (M)) on the MiliPoint dataset.

Method	Pre.	ACT	ID	Tot.	FLOPs
P+G [19]+[20]	6.60	1.88	1.87	10.35	8
D+G [21]+[20]	6.60	18.07	17.97	42.64	888
FastHAR [22]	6.60	5.39	5.53	17.52	124
PTV3 [23]	–	63.14	63.23	126.37	24 560
Proposed (S)	2.40	1.36	0.96	4.72	30
Proposed (D)	2.40	1.61		4.01	19

Table 2
Accuracy (%) comparison on the MiliPoint and Pantomime datasets.

Method	MiliPoint		Pantomime	
	ACT	ID	ACT	ID
P+G [19]+[20]	91.48	98.37	95.52	91.44
D+G [21]+[20]	97.54	99.85	97.67	88.85
FastHAR [22]	97.34	99.85	95.45	78.36
PTV3 [23]	94.28	99.36	95.17	95.89
Proposed (S)	98.87	99.85	98.56	99.33
Proposed (D)	98.65	99.85	98.24	98.60

run on a desktop with an i5-12400 CPU, an NVIDIA RTX 3050 GPU, and 32 GB RAM, and the same environment is used for all methods.

The *Proposed (S)* row reports the results of our architecture when activity recognition and user identification are performed separately, and the total inference time is obtained by summing the two inference times. The *Proposed (D)* runs both heads jointly and produces activity and identity in one pass.

Note that P+G, D+G, and FastHAR operate on fixed-size inputs and require resampling during preprocessing, incurring non-negligible overhead. In terms of total latency, the proposed method is the fastest among all compared approaches, even compared to P+G. Moreover, compared to the single-task variant, the dual-task variant reduces FLOPs by 36.7%. These results indicate that our approach is well suited for latency-critical applications and has strong potential for deployment on edge devices.

2. **Activity Recognition**
For evaluation, we perform 5-fold cross-validation. In each fold, all baselines are trained from scratch using the same random seed for initialization.
In Table 2, ACT denotes activity-recognition accuracy (49 activities in MiliPoint and 21 gestures in Pantomime). Although P+G is computationally light (smallest FLOPs), it achieves inferior performance, suggesting limited representational capacity to capture the data variability. While the remaining baselines achieve strong performance, the proposed method consistently attains the best ACT accuracy across both datasets. Notably, the dual-task variant achieves ACT performance comparable to the single-task variant, despite simultaneously inferring identity, indicating that sharing the encoder does not compromise activity recognition.
3. **User Identification**
In Table 2, ID denotes identity-recognition accuracy (11 user classes in MiliPoint and 41 user classes in Pantomime). On MiliPoint, the relatively larger number of samples together with the small number of user classes leads to near-saturated performance, and thus all schemes achieve very high ID accuracy. In contrast, Pantomime has fewer samples and a substantially larger number of user classes, which results in noticeably lower ID accuracy for several baselines. In this setting, capturing structural patterns is important for user identification; accordingly, PTV3 and the proposed method achieve higher ID accuracy on Pantomime.

Finally, the dual-task variant remains comparable to the single-task identity model. This indicates that the shared-backbone network effectively encodes information beneficial for both activity recognition and user identification, without sacrificing identity performance.

In addition, our ablation study on MilliPoint shows that, over the sum+mean pooling readout baseline in [12], the proposed temporal and kernel-based readouts improve activity-recognition and user-identification accuracy by 0.32% and 0.20%, respectively.

4. Conclusion

We have presented a shared-backbone graph network with task-specific heads for joint activity recognition and user identification using mmWave FMCW radar point clouds. Directed graphs captured geometric relations and temporal evolution, and the shared backbone processed these graphs. The activity head leveraged gated temporal aggregation, and the identity head used statistical pooling with random Fourier feature kernel mean embeddings, enabling variable-sized inputs without resampling. Experiments on public datasets show that the proposed dual-task design achieves performance comparable to its single-task counterparts, while reducing computational cost by more than threefold through parameter sharing. Future work includes extending the framework to multi-person settings and incorporating signal-level processing and sensor-fusion approaches, along with corresponding comparisons under these more challenging scenarios.

CRediT authorship contribution statement

Jun Yong Eom: Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **Daewon Seo:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that there is no conflict of interest in this paper.

Acknowledgment

This work was supported by the InnoCORE program of the Ministry of Science and ICT (MSIT), Republic of Korea [grant number 25-InnoCORE-01].

References

[1] S. Ahmed, S.H. Cho, Machine learning for healthcare radars: Recent progresses in human vital sign measurement and activity recognition, IEEE Commun. Surv. Tutor. 26 (1) (2024) 461–495.
[2] S. Deep, X. Zheng, C. Karmakar, D. Yu, L.G.C. Hamey, J. Jin, A survey on anomalous behavior detection for elderly care using dense-sensing networks, IEEE Commun. Surv. Tutor. 22 (1) (2020) 352–370.
[3] M.V. Karthikeyan, M.F. M, J. R, Public human assault prediction using human activity recognition with AI, in: Proc. Int. Conf. Adv. Data Eng. Intell. Comput. Syst., ADICS, 2024, pp. 1–5.
[4] C.I. Nwakanma, G.O. Anyanwu, L.A.C. Ahakonye, J.-M. Lee, D.-S. Kim, A review of thermal array sensor-based activity detection in smart spaces using AI, ICT Express 10 (2) (2024) 256–269.
[5] Y. Kim, W.S. Jeon, D.G. Jeong, AmFall: WiFi CSI amplitude-based fall detection using denoised scalograms, IEEE Internet Things J. 12 (18) (2025) 37988–38003.
[6] A.D. Aishwarya, R.I. Minu, Edge computing based surveillance framework for real-time activity recognition, ICT Express 7 (2) (2021) 182–186.
[7] A. Memon, Q. Arain, N. Pirzada, A. Shaikh, A. Sulaiman, M.S.A. Reshan, H. Alshahrani, A. Shaikh, Prior-free 3D human pose estimation in a video using limb-vectors, ICT Express 10 (6) (2024) 1266–1272.
[8] K. Shrestha, G.B. Pradhan, T. Bhatta, S. Sharma, S. Lee, H. Song, S. Jeong, J.Y. Park, Intermediate nanofibrous charge trapping layer-based wearable triboelectric self-powered sensor for human activity recognition and user identification, Nano Energy 108 (2023) 108180.
[9] C. Cao, Y. Ding, M. Dai, W. Gong, X. Zhao, Real-time cross-domain gesture and user identification via COTS WiFi, IEEE Trans. Mob. Comput. (2025) 1–13.

[10] C. Yu, Z. Xu, K. Yan, Y.-R. Chien, S.-H. Fang, H.-C. Wu, Noninvasive human activity recognition using millimeter-wave radar, IEEE Syst. J. 16 (2) (2022) 3036–3047.
[11] L. Xu, K. Wang, C. Gu, X. Guo, S. He, J. Chen, GesturePrint: Enabling user identification for mmWave-based gesture recognition systems, in: Proc. IEEE Int. Conf. Distrib. Comput. Syst., ICDCS, 2024, pp. 1074–1085.
[12] J.Y. Eom, W.S. Jeon, Gesture recognition with mmWave FMCW radar using a trajectory-aware graph encoder, IEEE Internet Things J. 12 (24) (2025) 54320–54334.
[13] C. Iovescu, S. Rao, The Fundamentals of Millimeter Wave Sensors, Texas Instruments, Dallas, TX, USA, 2017, pp. 1–8.
[14] V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in: Proc. ICML 2010, Haifa, Israel, Jun. 2010, pp. 807–814.
[15] A. Rahimi, B. Recht, Random features for large-scale kernel machines, in: Adv. Neural Inf. Process. Syst., vol. 20, Vancouver, Canada, 2007, pp. 1177–1184.
[16] K. Muandet, K. Fukumizu, B. Sriperumbudur, B. Schölkopf, Kernel mean embedding of distributions: A review and beyond, Found. Trends Mach. Learn. 10 (1–2) (2017) 1–141.
[17] H. Cui, S. Zhong, J. Wu, Z. Shen, N. Dahnoun, Y. Zhao, MiliPoint: A point cloud dataset for mmwave radar, in: Proc. NeurIPS 2023, New Orleans, USA, 2023, pp. 62713–62726.
[18] S. Palipana, D. Salami, L.A. Leiva, S. Sigg, Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds, ACM Interact. Mob. Wearable Ubiquitous Technol. 5 (1) (2021) 1–27.
[19] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: Deep hierarchical feature learning on point sets in a metric space, in: Proc. NeurIPS 2017, Long Beach, USA, 2017.
[20] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, arXiv preprint arXiv:1412.3555.
[21] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph CNN for learning on point clouds, ACM Trans. Graph. 38 (5) (2019) 146.
[22] T. Shao, Z. Du, C. Li, T. Wu, M. Wang, Fast human action recognition via millimeter wave radar point cloud sequences learning, in: Proc. ACM CIKM 2024, Boise, USA, 2024, pp. 2024–2033.
[23] X. Wu, et al., Point transformer V3: Simpler faster stronger, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, Seattle, WA, USA, 2024, pp. 4840–4851.
[24] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, A. Geissbuhler, Learning from imbalanced data in surveillance of nosocomial infection, Artif. Intell. Med. 37 (1) (2006) 7–18.