IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 7, 2021, accepted May 22, 2021, date of publication May 26, 2021, date of current version June 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3083980

Real-Time People Tracking and Identification From Sparse mm-Wave Radar Point-Clouds

JACOPO PEGORARO[1], (Graduate Student Member, IEEE), AND MICHELE ROSSI[1,2], (Senior Member, IEEE)

[1]Department of Information Engineering, University of Padova, 35131 Padova, Italy
[2]Department of Mathematics Tullio Levi-Civita, University of Padova, 35131 Padova, Italy

Corresponding author: Jacopo Pegoraro (pegoraroja@dei.unipd.it)

This work was supported in part by the Italian Ministry of Education, University and Research (MIUR) through the Initiative Departments of Excellence under Law 232/2016.

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

ABSTRACT Mm-wave radars have recently gathered significant attention as a means to track human movement and identify subjects from their gait characteristics. A widely adopted method to perform the identification is the extraction of the *micro-Doppler signature* of the targets, which is computationally demanding in case of co-existing multiple targets within the monitored physical space. Such computational complexity is the main problem of state-of-the-art approaches, and makes them inapt for real-time use. In this work, we present an end-to-end, low-complexity but highly accurate method to track and identify multiple subjects in real-time using the sparse point-cloud sequences obtained from a low-cost mm-wave radar. Our proposed system features an extended object tracking Kalman filter, used to estimate the position, shape and extension of the subjects, which is integrated with a novel deep learning classifier, specifically tailored for effective feature extraction and fast inference on radar point-clouds. The proposed method is thoroughly evaluated on an edge-computing platform from NVIDIA (Jetson series), obtaining greatly reduced execution times (reduced complexity) against the best approaches from the literature. Specifically, it achieves accuracies as high as 91.62%, operating at 15 frames per seconds, in identifying three subjects that concurrently and freely move in an unseen indoor environment, among a group of eight.

INDEX TERMS mm-wave radar, person identification, point-clouds, multi-target tracking, convolutional neural networks.

I. INTRODUCTION

The use of mm-wave radars for physical environment sensing is a fast growing research area [1], [2]. They can be used to infer the position of the surrounding obstacles and humans, with high precision, by transmitting an interrogation signal and analyzing the modifications in the received reflected waves [3]. These systems represent an effective means to monitor indoor environments, inferring key information about the movement of people, without capturing any visual image of the scene, which could rise privacy concerns. In addition, in contrast with camera surveillance systems, radars are insensitive to poor light conditions, to the presence of smoke [4], and are also energy efficient and low-cost as compared to other technologies such as LIDARs [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Yi Zhang.

The high sensitivity of mm-waves to the frequency shifts caused by the Doppler effect makes them suitable to infer the movement patterns of humans. A widely adopted method of analysis is to extract features from the so-called *micro-Doppler signature* (μD) of the subject, which contains time-frequency information about the induced Doppler shift, including the contribution of small-scale movements [5], [6]. μD signatures have been used in the challenging task of distinguishing subjects from their way of walking (*gait*), which is the aim of the present work.

Human gait has been classified as a *soft biometric* [7], meaning it is unique for each person. Differently from *hard biometrics*, however, such as fingerprints or DNA, it cannot be used in high-stakes settings or to uniquely identify subjects among very large groups, e.g., more than $1,000$ people. Despite this, gait is difficult to fake, and it can be effectively analyzed even at distance and without requiring the subjects to collaborate. For these reasons, mm-wave radar based

gait recognition can be a good option to identify subjects in scenarios such as surveillance systems or individually-tailored smart home applications, where the number of people involved is in the order of a few tens, replacing or augmenting traditional camera systems.

Using a *deep learning* (DL) classifier on raw micro-Doppler spectrograms has proven to be robust and effective for identifying subjects in the case of single- and multi-target scenarios [8]–[11]. The extraction of representative features from μD signatures is often accomplished via neural networks (NN) and DL methods. These, are becoming the tools of choice due to the high randomness of mm-wave propagation, which makes a full mathematical modelling of the involved dynamics very difficult [12]–[14].

In the present work, we design and validate a realtime *multi-target* tracking and identification system running on constrained edge-computing devices[1] equipped with hardware accelerators (last generation GPUs). Instead of working on the raw data obtained from the backscattered mm-wave signal, as commonly done in the literature, we use *sparse point-clouds*. This makes it possible to implement our system on resource limited edge-computing devices. Point-clouds carry information about the three-dimensional spatial coordinates of the reflecting points, their velocity and the reflected power, and are obtained by employing detection algorithms at the radar processing unit, thus avoiding the need for transferring the *full raw data* from the radar to the edge computer. Due to their much lower data size, they bring advantages in terms of communication and computation at the connected processing device. Nonetheless, these advantages entail a *more challenging person identification task*: the sparsity of radar point-cloud data can be a source of inaccuracy and *standard DL architectures are inapt for learning from them*, as they rely on the reciprocal ordering of their input elements [15]. As a solution, we present a novel DL classifier, called temporal convolution point-cloud network (TCPCN), which allows extracting meaningful order-invariant features from sparse point-cloud data.

The proposed system sequentially performs person tracking and identification, estimating the positions and the identities of humans as they freely move in an indoor space. For that, we use a low-cost Texas Instruments IWR1843BOOST mm-wave, frequency-modulated continuous-wave (FMCW), multiple-input multiple-output (MIMO) radar and implement the required processing functions in real-time on a commercial edge-computing node (NVIDIA Jetson series). To carry out the person identification task, we combine standard tracking techniques, i.e., Kalman filter, with DL methods. This combined use of filtering and DL makes it possible to effectively capture the time evolution of the point-cloud representing each subject. Our main contributions are:

1) We build an end-to-end tracking and identification system that reliably operates in real-time at over 15 fps on a commercial edge-computing device paired with a low-cost mm-wave radar. The approach reaches an accuracy of 91.62% in identifying up to three subjects (among a group of eight) freely and concurrently moving in a new indoor space, i.e., not seen at training time.
2) We propose a novel DL classifier, called *temporal convolution point-cloud network* (TCPCN), that is tailored on mm-wave radar point-cloud sequences and that is both accurate and fast. TCPCN contains a feature extraction block that obtains global information from the radar output at each time-frame and a block that exploits causal dilated convolutions [16] to recognize meaningful patterns in the temporal evolution of the features. Our model significantly outperforms state-of-the-art neural networks in this field in terms of classification accuracy and inference time.
3) The tracking phase of our system employs a converted-measurements Kalman filter (CM-KF) that, in addition to estimating the position of the targets in Cartesian coordinates, also estimates the extension of the subject in the horizontal plane ($x - y$), considering him/her as an *extended object* rather than an ideal point-shaped reflector. This provides useful additional information that could be exploited by, e.g., occupancy or proximity based applications. In fact, knowing the extension of the subjects would be valuable for *(i)* smart-home applications that perform occupancy detection in certain areas, *(ii)* security systems in industrial settings, to estimate how close a person is to some dangerous area or machinery, *(iii)* detection systems (e.g., for automatic gates) that could quickly discern between cars, adults, kids or pets from their size. To the best of our knowledge, no earlier work uses *extended object tracking* (EOT) within a point-cloud based tracking and identification system.

The novelty of the proposed solution stems from the following main points: the design and implementation of a novel DL-based neural network classifier working on time sequences of sparse point-cloud data, that is at the same time *highly accurate* and *fast*, the integration of tracking and identification phases, that in the literature on the subject are usually dealt with separately, the implementation and validation of the solution on a commercially available edge-computing platform with limited capacity.

The rest of the paper is structured as follows. In Section II, the literature on person identification using mm-wave radars is reviewed, underlining the novel aspects in our approach. In Section III, the FMCW MIMO radar signal model is outlined, by also describing the procedure to extract the point-clouds. Our proposed framework is presented in Section IV. In Section V, experimental results are shown, while concluding remarks are given in Section VI.

II. RELATED WORK

In the last few years, person identification from backscattered mm-wave radio signals has attracted a considerable and growing interest. Most of the research attention has been

[1] As an example, see the NVIDIA Jetson series.

paid to processing human micro-Doppler (μD) signatures as a means to distinguish among subjects, usually employing deep learning classifiers, applied to the μD spectrogram [8]–[11], [17]–[20]. Although this approach is robust and accurate, it presents some drawbacks. First, the extraction of μD signatures in case of multiple targets is a rather complex endeavour, and most of the above referenced solutions only work for a single-subject. In very few works, e.g., [11], the authors devised methods to single-out the contribution from multiple concurrent targets, obtaining the individual μD signatures. However, in the interest of obtaining highly accurate signatures, these previous algorithms dealt with *non-sparse* radar range-azimuth-Doppler (RDA) maps that require a large communication bandwidth to transfer the raw radio data from the radar to the processing device, preventing their implementation on low-cost embedded boards.

Only a few works so far have considered point-clouds obtained from a low-cost mm-wave MIMO radar device. The sparsity of radar point-cloud data makes the identification task more challenging, as the specific features that identify each subject are more difficult to extract, and more sensitive to external disturbances. In [21], a recurrent neural network with long short-term memory (LSTM) cells is used for the identification. The overall accuracy obtained for 12 subjects is around 89%, and evidence that the system is able to distinguish between two concurrently walking subjects is provided. However, no evaluation of the accuracy is conducted when more than 2 subjects share the same physical space, nor by testing it in a different indoor environment (e.g., a new room) after its training. In addition, the point-cloud nature of the radar data is not fully exploited: the velocity and the received power are not used, and the classifier network requires the input data to be mapped onto a 3D voxel representation, which is inefficient and computationally expensive. The authors of [22] proposed a deep learning model that outperforms the bi-directional LSTM in [21] on their dataset. Two radar devices are used, transmitting and receiving simultaneously, leading to an increased field of view in case of blockage. However, robust methods are neither provided for tracking multiple subjects, e.g., Kalman or particle filtering [23], nor to reliably associate the detections (user identities) with trajectories. This seriously impacts the identification performance when multiple targets freely move in the monitored environment. In [22], it is in fact reported that the accuracy drops to 45% in a multi-target setting.

With the present work, we fill a literature gap, by designing a system that performs accurate tracking of multiple subjects from their point-clouds. Extended object tracking based on Kalman filtering is exploited in conjunction with a fast and novel domain-specific deep learning classifier. A tight integration of the tracking and identification modules is sought, towards enhancing the identification robustness and avoiding wrong identity associations and trajectory swaps. Moreover, and to the best of our knowledge, we are the first to provide an empirical study on the feasibility of operating the system in realtime on commercial edge-computing devices, and low-cost mm-wave radars.

III. mmWave RADAR SIGNAL PROCESSING

A *frequency-modulated continuous wave* (FMCW) radar allows the joint estimation of the distance and the radial velocity of the target with respect to the radar device. This is achieved by transmitting sequences of linear *chirps*, i.e., sinusoidal waves with frequency that is linearly increased over time, and measuring the frequency shift of the reflected signal at the receiver. The frequency of the transmitted chirp signal is increased from a base value f_o to a maximum f_1 in T seconds. Defining the bandwidth of the chirp as $B = f_1 - f_o$, bandwidth B and chirp duration T are related through $\zeta = B/T$, and the transmitted signal is expressed as

$$s(t) = \exp\left[j2\pi\left(f_o + \frac{\zeta}{2}t\right)t\right], \quad 0 \le t \le T. \tag{1}$$

The chirps are transmitted every T_{rep} seconds in sequences of L chirps each, so that the total duration of a transmitted (TX) sequence is LT_{rep}. A full sequence, termed *radar frame*, is repeated with period Δt. At the receiver, a mixer combines the received signal (RX) with the one transmitted, generating the intermediate frequency (IF) signal, i.e., a sinusoid whose instantaneous frequency corresponds to the difference between those of the TX and RX signals. Each chirp is sampled with sampling period T_f (referred to as *fast time* sampling) obtaining M points, while L samples, one per chirp from adjacent chirps, are taken with period T_{rep} (*slow time* sampling).

The use of multiple-input multiple-output (MIMO) radar devices allows the additional estimation of the angle-of-arrival (AoA) of the reflections, by computing the phase shifts between the receiver antenna elements due to their different positions (i.e., their different distances from the target). This is referred to as *spatial* sampling, and enables the localization of the targets in the physical space. The radar device used in this work has $N_{\text{TX}} = 3$ transmitter and $N_{\text{RX}} = 4$ receiver antennas, that are equivalent to a virtual receiver array of $N_{\text{TX}}N_{\text{RX}} = 12$ antennas. The transmitting elements are arranged along two spatial dimensions, which we refer to as azimuth (AZ) and elevation (EL), and are used to transmit the chirp sequences according to a time-division multiplexing (TDM) scheme. This enables the estimation of the EL and AZ angles of the reflecting points. In Section III-A, we first consider one of the receiver elements, referring to it as *reference antenna*, and describe how the range and velocity of the subjects are estimated. In Section III-B, we extend the discussion to multiple receiver antennas, showing how the AZ and EL AoAs are computed.

A. RANGE AND DOPPLER INFORMATION

Next, we show how to extract the range and velocity information from the received signal, focusing on the reference antenna. The signal reflected by a target is an attenuated version of the transmitted waveform with a delay τ

that depends on the distance between the target and the radar and on their relative radial velocity.

Denoting by c the speed of light, and letting R and v respectively be the range and velocity of the target with respect to the radar device, the reflected signal delay is

$$\tau = \frac{2(R + vt)}{c}. \tag{2}$$

After mixing and sampling, the IF signal is expressed as [2]

$$y(m, l) = \alpha \exp\left[j\varphi(m, l)\right] + \mathrm{w}(m, l), \tag{3}$$

where m and l represent the sampling indices along the fast and slow time, respectively, α is a coefficient accounting for the attenuation effects due to the antenna gains, path loss and radar cross section (RCS) of the target and $\mathrm{w}(m, l)$ is a Gaussian noise term. The phase $\varphi(m, l)$ depends on the fast time and slow time sampling indices. By neglecting the terms giving a small contribution, an approximate expression for $\varphi(m, l)$ is written by introducing the quantities $f_d = 2f_o v/c$ and $f_b = 2\zeta R/c$, which respectively represent the Doppler frequency and the *beat* frequency of the reflected signal,

$$\varphi(m, l) \approx 2\pi \left[\frac{2f_o R}{c} + f_d l T_{\mathrm{rep}} + (f_d + f_b)\, m T_f\right]. \tag{4}$$

Samples of $y(m, l)$ can be arranged into an $M \times L$ matrix containing all the information provided by a single antenna for a given time frame. The frequency shifts of interest, which reveal the range and velocity of each reflector, can be extracted after applying a bi-dimensional discrete Fourier transform (DFT) along the fast time and slow time dimensions, followed by taking the square magnitude of each obtained complex value. The result of this process is often referred to as radar *range-Doppler map* (RD), and represents the received power distribution along the range of distances and velocities of interest.

The detection of the main reflecting points is performed using the *cell-averaging constant false alarm rate* (CA-CFAR) algorithm on the range-Doppler maps [24], which consists in applying a dynamic threshold on each RD value (or *bin*), depending on the power of nearby *training* values. The use of an adaptive threshold introduces sparsity in the resulting set of detected points, as a point is retained (i.e., selected) only if its power is sufficiently larger than the average power of its neighbors.

In addition, a processing step is required to remove the reflections from static objects, i.e., the *clutter*. This operation is performed using a *moving target indication* (MTI) high pass filter that removes the reflections with Doppler frequency values close to zero [24].

The detection and MTI processing steps return a sparse RD map containing N^{det} detected reflecting points: the position of each value along the fast time reveals the corresponding frequency in the IF signal $f_d + f_b \approx f_b$, while the peak along the slow time reveals the Doppler frequency f_d. For each detected point, the *observed* desired quantities are then expressed as follows (we indicate with the symbol Δ the corresponding resolution)

$$\tilde{R} = \frac{f_b c}{2\zeta}, \quad \Delta\tilde{R} = \frac{c}{2B}, \tag{5}$$

$$\tilde{v} = \frac{f_d c}{2f_o}, \quad \Delta\tilde{v} = \frac{c}{2f_o L T_{\mathrm{rep}} N_{\mathrm{TX}}}. \tag{6}$$

Additionally, from the RD map we obtain the reflected, received power from each detection, denoted by P^{RX}.

B. AZIMUTH AND ELEVATION ANGLES ESTIMATION

The complex-valued RD map of the radar illuminated range, before taking the square magnitude, is computed at all the receiving antenna elements, and presents a different phase shift at each antenna, due to its different distance from the target. This fact is referred to as *spatial diversity* of the receiver array, and can be exploited to estimate the azimuth and elevation angles of the targets.

Denote by d the distance between two subsequent antennas along the azimuth and elevation dimensions and by ψ_{AZ} and ψ_{EL} the corresponding experienced phase shifts, respectively. Moreover, let θ and ϕ be the AZ and EL angles of a reflecting point, while $\lambda = c/f_o$ is the base wavelength of the transmitted chirps. The following relations hold

$$\begin{aligned} \psi_{\mathrm{AZ}} &\approx \frac{2\pi}{\lambda} d \cos\phi \sin\theta, \\ \psi_{\mathrm{EL}} &\approx \frac{2\pi}{\lambda} d \sin\phi. \end{aligned} \tag{7}$$

To compute the phase shift values, two DFTs across the samples taken at the azimuth and elevation antennas in the virtual receiver array are computed, extracting the peak positions similarly to what described in Section III-A for beat and Doppler frequency. Finally, the Cartesian coordinates of each detected point are obtained using Eq. (7) as

$$\begin{aligned} \tilde{x} &= \tilde{R}\cos\phi\sin\theta = \tilde{R}\frac{\lambda\psi_{\mathrm{AZ}}}{2\pi d}, \\ \tilde{y} &= \sqrt{\tilde{R}^2 - \tilde{x}^2 - \tilde{z}^2}, \\ \tilde{z} &= \tilde{R}\sin\phi = \tilde{R}\frac{\lambda\psi_{\mathrm{EL}}}{2\pi d}. \end{aligned} \tag{8}$$

The vector describing a single detected reflecting point, $\boldsymbol{p}_r$, $r = 1, \ldots, N^{\mathrm{det}}$, has five components, containing the information on its Cartesian coordinates, its velocity and the reflected power: $\boldsymbol{p}_r = \left[\tilde{x}_r, \tilde{y}_r, \tilde{z}_r, \tilde{v}_r, P_r^{\mathrm{RX}}\right]^T$.

IV. SYSTEM DESIGN

The proposed system operates on discrete time steps, indicized by variable k, whose duration corresponds to the radar inter frame time Δt. At each frame, a set of N_k^{det} reflecting points $\boldsymbol{p}_r$ are obtained through the signal processing steps of Section III. Our system sequentially performs the following operations on such points, see Fig. 1.

1) **Clustering and extension observation:** a density-based clustering algorithm is used to group the points

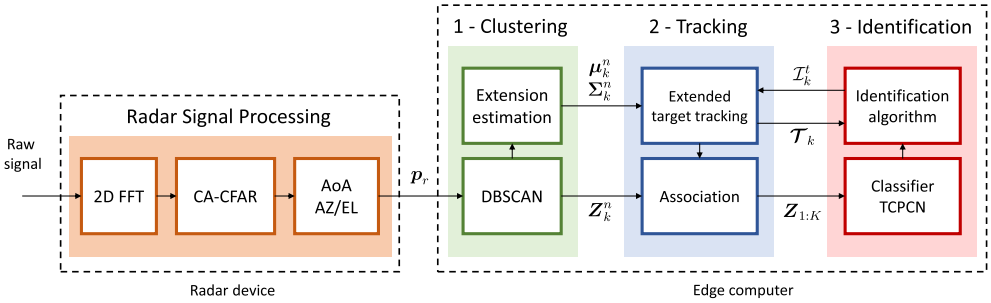


FIGURE 1. Block diagram of the proposed signal processing workflow: the raw radar data is processed on the radar device, extracting the sparse point-cloud representation of the environment, i.e., points p_r, then (1) a clustering module groups the points p_r into the contributions from the different targets and estimates their position and extension, (2-3) tracking, data association and identification are *jointly* performed through an identification algorithm.

detected by CA-CFAR into several clusters, each corresponding to a different subject present in the environment, see Section IV-A. The points associated with the different targets are then used to obtain *observations* of the subject's state, which according to our design includes his/her Cartesian position and *extension* in the horizontal plane ($x-y$). The extension is modeled as an ellipse, that is determined by the spread (covariance) of the points in each cluster, Section IV-B.

2) **Tracking and data association:** a CM-KF [25] is used to estimate the position, velocity and extension of the subjects in a multi-target tracking (MTT) framework, processing the observations outputted by the previous step, Section IV-C. A set of trajectories, each corresponding to a human subject, are maintained and sequentially updated. The MTT association between new observations and trajectories is achieved using an approximation of the *nearest-neighbors joint probabilistic data association* (NN-JPDA) algorithm, see Section IV-D.
3) **Identification:** a deep NN classifier is applied to a temporal sequence of K subsequent point-clouds associated with each trajectory, with the objective of discerning among a set of Q pre-defined subject identities. The employed NN is called *temporal convolution point-cloud network* (TCPCN), and is inspired by the popular PointNet architecture used for 3D point-cloud classification and segmentation [15]. TCPCN extends PointNet to the radar domain, by adding the velocity and received power information to the input and accounting for an additional block that handles the extraction of temporal features. Also, TCPCN is used in conjunction with an identification algorithm, which includes an exponential moving-average smoother and the Hungarian method, to jointly output a unique label for each trajectory: this combined use greatly improves the identification accuracy of the framework.

A. POINT-CLOUD CLUSTERING – DBSCAN

Density-based clustering algorithms, as opposed to *distance*-based ones, group input samples according to their local density. One of the most widely used algorithms belonging to this category is DBSCAN [26], which has been successfully applied to cluster radar point clouds in [11], [21], [22], [27]. The algorithm operates a sequential scanning of all the data points, expanding a cluster until a certain density connectivity condition is no longer met. The algorithm takes two input parameters, ε and m_{pts}, respectively representing a radius around each point and the minimum number of other points that must be inside such radius to meet the density condition. DBSCAN is only applied to the $x-y$ components of the detected points $\boldsymbol{p}_r$, namely, the Cartesian coordinates on the horizontal plane, as the different body parts of a subject can have very different velocity and reflected power values. We denote by $\left\{\boldsymbol{Z}_k^n\right\}_{n=1,\dots,D_k}$ the D_k clusters obtained at time step k by grouping the N_k^{det} detected points. In principle, there should be a distinct cluster for each human subject present in the environment, but due to several phenomena such as noise, imperfect clutter cancellation and blockage of the signal, a subject can go undetected even for several consecutive frames. DBSCAN was chosen for the following reasons: it is an unsupervised algorithm, i.e., the number of clusters (subjects) does not have to be known beforehand, it has a noise rejection quality that, together with its density-based clustering mechanism, allows a reliable and automatic separation of the reflections from distinct subjects, it has a low computational complexity, of about $\mathcal{O}\left(N_k^{\text{det}} \log N_k^{\text{det}}\right)$.

B. SUBJECT POSITION AND EXTENSION OBSERVATIONS

Due to the high spatial resolution of mm-wave radars, human subjects are detected as clusters containing tens of reflecting points. In the literature, the typical approach to their tracking has been to ignore the spatial extension of the targets, considering them as ideal point-shaped reflectors. In the present work, given a cluster of points $\boldsymbol{Z}_k^n$ selected by the DBSCAN clustering algorithm at time k, we instead obtain an estimate of the extension of the subject in the $x-y$ plane. As a first step, we define $\tilde{\boldsymbol{p}}_r = \left[\tilde{x}_r, \tilde{y}_r\right]^T$ and we normalize the received power values, P_r^{RX}, of the detected points in $[0, 1]$. The spread of the points within each cluster around the cluster centroid provides a measure of the subject's extension. The centroid

represents a noisy observation of the true position of the person, and is obtained as

$$\boldsymbol{\mu}_k^n = \sum_{r:\tilde{\boldsymbol{p}}_r \in \boldsymbol{Z}_k^n} P_r^{\mathrm{RX}} \tilde{\boldsymbol{p}}_r, \tag{9}$$

where $\boldsymbol{\mu}_k^n = [\mu_{x,k}^n, \mu_{y,k}^n]^T$ and the received normalized powers P_r^{RX} act as weights. The covariance matrix, $\boldsymbol{\Sigma}_k^n$, contains information on the dimensions of the ellipse representing the extension of cluster n, and is obtained through the weighted sample covariance estimator,

$$\boldsymbol{\Sigma}_k^n = \sum_{r:\tilde{\boldsymbol{p}}_r \in \boldsymbol{Z}_k^n} P_r^{\mathrm{RX}} \left(\tilde{\boldsymbol{p}}_r - \boldsymbol{\mu}_k^n\right)\left(\tilde{\boldsymbol{p}}_r - \boldsymbol{\mu}_k^n\right)^T. \tag{10}$$

The norms of the eigenvectors of matrix $\boldsymbol{\Sigma}_k^n$, denoted by $\tilde{\ell}_k^n$ and $\tilde{w}_k^n$ provide the axes lengths of the ellipse, while the orientation, $\tilde{\xi}_k^n$, has the same direction of the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{\Sigma}_k^n$.

C. EXTENDED OBJECT TRACKING – CONVERTED MEASUREMENTS KALMAN FILTER

With the tracking step, we perform a sequential estimation of the *state* of the subjects present in the environment from their observed positions and extensions. To this end, we use a set of CM-KFs to establish a so-called *track* for each subject. A new KF model is initialized for each detected cluster in the first frame received by the radar, while in successive frames, the tracks are maintained through the KF predict-update steps [23]. We denote by $\mathcal{T}_k^t$ the track with index t at time k, by $\boldsymbol{\mathcal{T}}_k$ the set of currently maintained tracks, i.e., $\boldsymbol{\mathcal{T}}_k = \left\{\mathcal{T}_k^t\right\}_{t=1,\ldots,T_k}$, and by T_k its cardinality. We define the state of $\mathcal{T}_k^t$ as $\mathbf{x}_k^t = \left[x_k^t, y_k^t, \dot{x}_k^t, \dot{y}_k^t, \ell_k^t, w_k^t, \xi_k^t\right]^T$, which contains the true (and unknown) user's position (x_k^t and y_k^t), velocity ($\dot{x}_k^t$ and $\dot{y}_k^t$), extension (ℓ_k^t and w_k^t) and orientation angle (ξ_k^t). Each track is then defined as a tuple, $\mathcal{T}_k^t = \left(\hat{\mathbf{x}}_k^t, \boldsymbol{P}_k^t, \boldsymbol{Z}_{k-K+1:k}^t, \mathcal{I}_k^t\right)$, containing respectively the current state estimate, $\hat{\mathbf{x}}_k^t$, the associated error covariance matrix as computed by the KF, $\boldsymbol{P}_k^t$, the collection of the last K clusters associated with the track, $\boldsymbol{Z}_{k-K+1:k}^t$, to be fed to the NN classifier, and an integer $\mathcal{I}_k^t$ representing an estimate of the identity of the associated subject, at time k. The observation vector for a detected target n at time k is $\mathbf{z}_k^n = \left[\mu_{x,k}^n, \mu_{y,k}^n, \tilde{\ell}_k^n, \tilde{w}_k^n, \tilde{\xi}_k^n\right]^T$.

The matching between any given cluster n and a corresponding track t ($n \leftrightarrow t$) is carried out using a specific procedure that will be detailed shortly in Section IV-D. For the sake of a concise notation, for the remainder of this section we drop the indices n and t, as the procedure that we describe next is carried out independently for each track (subject) once the matching $n \leftrightarrow t$ is performed.

Given the sequence of all collected measurements for a track up to time k, $\mathbf{z}_{1:k}$, the state estimation is carried out using the CM-KF. This approach assumes a posterior Gaussian distribution of the state given the sequence of measurements, i.e., $p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \mathcal{N}(\hat{\mathbf{x}}_k, \boldsymbol{P}_k)$. To update $\hat{\mathbf{x}}_k$ and $\boldsymbol{P}_k$, a KF recursion [23] is applied using the measurements transformed in Cartesian coordinates from Section IV-B.

The model of motion that is used by the Kalman filtering block is defined by two matrices, $\boldsymbol{F}$ and $\boldsymbol{H}$. $\boldsymbol{F}$ is the transition matrix, connecting the system state at time k, $\mathbf{x}_k$, to that at time $k-1$, $\mathbf{x}_{k-1}$. $\boldsymbol{H}$ is the observation matrix, which relates the observation vector $\mathbf{z}_k$ to the true state $\mathbf{x}_k$. Referring to $\boldsymbol{u}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q})$ and $\boldsymbol{r}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_k)$ as the process noise and observation noise, respectively, a dynamic model of the system is

$$\mathbf{x}_k = \boldsymbol{F}\mathbf{x}_{k-1} + \boldsymbol{u}_k, \tag{11}$$

$$\mathbf{z}_k = \boldsymbol{H}\mathbf{x}_k + \boldsymbol{r}_k. \tag{12}$$

Denoting by blkdiag[$\boldsymbol{A}, \boldsymbol{B}$] the block diagonal matrix with blocks given by matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, we have

$$\boldsymbol{F} = \mathrm{blkdiag}\left[\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \otimes \boldsymbol{I}_2, \boldsymbol{I}_3\right], \tag{13}$$

and

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I}_2 & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 3} \\ \mathbf{0}_{3\times 2} & \mathbf{0}_{3\times 2} & \boldsymbol{I}_3 \end{bmatrix}, \tag{14}$$

where $\boldsymbol{I}_n$ is an $n \times n$ identity matrix, $\mathbf{0}_{n\times m}$ is an $n \times m$ all-zero matrix and $\otimes$ refers to the Kronecker product between matrices.

We assume the process noise $\boldsymbol{u}_k$ is due to a random acceleration a_k that follows a Gaussian distribution with 0 mean and variance σ_a^2, i.e., $a_k \sim \mathcal{N}(0, \sigma_a^2)$, leading to $\boldsymbol{u}_k = \boldsymbol{g}a_k$ with $\boldsymbol{g} = \left[\Delta t^2/2, \Delta t\right]^T$. The process noise covariance matrix is obtained as

$$\boldsymbol{Q} = \mathrm{blkdiag}\left[\sigma_a^2 \boldsymbol{g}\boldsymbol{g}^T \otimes \boldsymbol{I}_2, \mathrm{diag}\left(\sigma_\ell^2, \sigma_w^2, \sigma_\xi^2\right)\right], \tag{15}$$

with $\sigma_\ell^2, \sigma_w^2, \sigma_\xi^2$ being the constant process noise variances on the extension- and orientation-related coordinates of the state. The observation noise has covariance matrix given by

$$\boldsymbol{R}_k = \mathrm{blkdiag}\left[\boldsymbol{R}'(\mathbf{x}_k), \mathrm{diag}\left(\sigma_\ell^2, \sigma_w^2, \sigma_\xi^2\right)\right], \tag{16}$$

with $\sigma_\ell^2, \sigma_w^2, \sigma_\xi^2$ being the constant observation noise variances on the extension- and orientation-related coordinates of the state. For what concerns $\boldsymbol{R}'$, as radar measurements are obtained in polar coordinates, and then converted to the Cartesian space using Eq. (8), the measurement covariance matrix is time-varying as it depends on the current target's position. The sub-matrix $\boldsymbol{R}'$ accounts for the uncertainty in the Cartesian position observations, reflecting that an error on the AoA causes a higher uncertainty in Cartesian coordinates as the distance of the subject increases, due to the non-linear mapping between polar and Cartesian coordinates. In setting the uncertainty parameters for the measurements, we use a constant measurement covariance in polar coordinates, $\boldsymbol{R}_{\mathrm{pol}} = \mathrm{diag}\left(\sigma_R^2, \sigma_\theta^2\right)$, where R and θ are the distance and azimuth AoA, respectively introduced in Section III-A and Section III-B. Hence, we use the transform $\boldsymbol{R}'(\mathbf{x}_k) = \boldsymbol{J}_{|\mathbf{x}_k} \boldsymbol{R}_{\mathrm{pol}} \boldsymbol{J}_{|\mathbf{x}_k}^T$, where $\boldsymbol{J}_{|\mathbf{x}_k}$ is the Jacobian matrix of the conversion between polar and Cartesian coordinates,

computed using the polar representation of the true subject state, $\mathbf{x}_k$, which we approximate with $\mathbf{x}_k \approx \boldsymbol{H}\hat{\mathbf{x}}_{k-1}$. Although it can be seen that our conversion to Cartesian coordinates is biased, we remark that employing the unbiased conversion proposed in [28] did not lead to significant improvements. Note that, by the structure of the model matrices in Eq. (13) and Eq. (14), the kinematic part of the subject state and the extension part are entirely decoupled and do not interact during the CM-KF operations.

As a final remark about the KF model, with our approach the extension of the subject is explicitly accounted for as part of the state, fitting the point-clouds with ellipses, similarly to [29]. Although other approaches exist, such as using random matrices [30], [31], we found that our method leads to more accurate and meaningful extension estimates of the target's shape, due to the fast variability of radar point-clouds.

D. DATA ASSOCIATION – NN-CJPDA

The association between new observations and tracks is needed *(i)* to correctly update the tracks with the observations generated by the corresponding subjects in a multi-target scenario, *(ii)* to correctly collect the sequence of the past K point-clouds associated with each subject, $\boldsymbol{Z}^t_{k-K+1:k}$.

To match tracks t to clusters n ($n \leftrightarrow t$), we use the *nearest-neighbors joint probabilistic data association* (NN-JPDA) scheme. This method consists in computing the probability of each possible association between the D_k new clusters and the previous T_k tracks. These probabilities are then arranged into a $D_k \times T_{k-1}$ matrix of scores, $\boldsymbol{\Gamma}$, and the final assignment is done considering the association leading to the maximum overall probability, computed using the Hungarian algorithm [32]. The Hungarian algorithm uses the score matrix as input and solves the problem of pairing each track with only one cluster while maximizing the total score, entailing an overall complexity $\mathcal{O}((T_{k-1}D_k)^3)$.

To compute the probability of each match, i.e., the elements of matrix $\boldsymbol{\Gamma}$, we consider the widely adopted JPDA logic, using the approximate version of [33] called cheap-JPDA (CJPDA). Exploiting the fact that the kinematic, extension and orientation parts of the state are decoupled in our framework, we apply CJPDA only using the kinematic state, as extension and orientation are more unreliable and could lead to association errors. Hence, in the following we refer to the kinematic part of the KF vectors and matrices only, i.e., to the components related to the Cartesian position and velocity of the targets.

The score matrix $\boldsymbol{\Gamma}$ is computed as follows (the time index k is omitted for a simpler notation). First, for all track-detection pairs the quantity G_{nt} is computed, which is proportional to the Gaussian function expressing the likelihood that observation n is produced by the subject corresponding to track t

$$G_{nt} = \frac{1}{\sqrt{\det \boldsymbol{S}_{nt}}} \exp\left[-\frac{1}{2}\boldsymbol{\nu}_{nt}^T \left(\boldsymbol{S}_{nt}\right)^{-1} \boldsymbol{\nu}_{nt}\right], \tag{17}$$

where $\boldsymbol{\nu}_{nt} = \hat{\mathbf{x}}^t - \boldsymbol{H}\mathbf{z}^n$ is the *innovation* brought by measurement $\mathbf{z}^n$ to the kinematic state of track t, $\hat{\mathbf{x}}^t$, and $\boldsymbol{S}_{nt} = \boldsymbol{H}\boldsymbol{P}^t\boldsymbol{H}^T + \boldsymbol{R}$ is its covariance matrix, obtained as part of the KF recursion. Second, the association probabilities for each track-detection pair are computed following [33], as

$$\Gamma_{nt} = \frac{G_{nt}}{\sum_{t=1}^{T_{k-1}} G_{nt} + \sum_{n=1}^{D_k} G_{nt} - G_{nt} + \beta}, \tag{18}$$

where the bias term β accounts for the possibility that no measurement is a good match for a specific track and is connected with the probability of missed detection. In this work, β is empirically set to $\beta = 0.01$, preventing the association of track-detection pairs with a low G_{nt} score.

E. TRACK MANAGEMENT

The proposed system is robust to subjects that randomly appear on and disappear from the monitored space: these events may happen due to blockage of the radar signal at any point in time, or because the subject has moved in or out of the radio range. Blockage is a frequent problem in mm-wave propagation and it happens frequently in multi-target scenarios, as users may block the radio signal with their own body. To deal with undetected subjects and new cluster detections which cannot be reliably associated with any existing track, while keeping the complexity of the system as low as possible, we follow a so-called `m/n` logic. In detail, a track is maintained if it received a match with any of the clusters detected by DBSCAN for at least `m` out of the last `n` frames. Similarly, cluster detections that are not associated with any existing track are initialized as new trajectories if they are detected for at least `m` out of the last `n` frames. In addition, to avoid tracks to merge when the subjects move too close to one another, the inter-track proximity is monitored. If the estimated Euclidean distance[2] between any two tracks $\mathcal{T}_k^t$ and $\mathcal{T}_k^{t'}$ becomes smaller than the DBSCAN radius parameter, ε, we remove the track having the largest determinant of the estimated error covariance, i.e., $\text{argmax}_{j\in\{t,t'\}}(\det \boldsymbol{P}_k^j)$.

F. POINT-CLOUD PRE-PROCESSING

The point cloud sequence $\boldsymbol{Z}^t_{k-K+1:k}$ obtained from each CM-KF track is pre-processed before being sent to the NN classifier. The features of the points are standardized by subtracting their mean value and dividing by their empirical standard deviation. Moreover, the point-clouds must contain a fixed number of points before being sent to the TCPCN, as the latter is a feed forward neural network processing fixed size input vectors. We chose to limit the maximum number of points for a single time step to $n_{\max} = 100$. In case the number of points is greater than such maximum value, we randomly sample $n_{\max}$ points from the point-cloud without repetitions, in case there are fewer points than $n_{\max}$, some of the points are randomly repeated to reach the maximum value. The choice of $n_{\max}$ was made by analyzing the distribution of the number of detected points for different

[2]Obtained as $d(\mathcal{T}_k^t, \mathcal{T}_k^{t'}) = ((x_k^t - x_k^{t'})^2 + (y_k^t - y_k^{t'})^2)^{1/2}$.

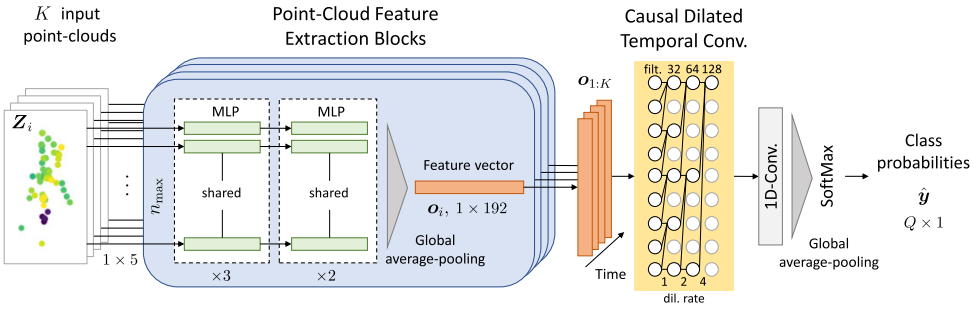


FIGURE 2. TCPCN – proposed DL-based classifier for subject identification: *(i)* a *point-cloud block* is applied to each individual time step to extract a feature vector, *(ii)* causal dilated convolutions are used to learn the temporal patterns in the sequence of feature vectors.

human subjects and empirically picking a suitable value: the selected $n_{\max}$ suffices to contain the point-clouds of all users in almost every frame in our experiments. Also, due to blockage and clutter, a subject may go undetected, especially in a multi-target scenario. If this occurs, the point-cloud data for the current frame is not collected for the blocked user and, in turn, is not sent to the NN classifier. A missed detection persisting over multiple radio frames may make the sequence of temporal features extracted for a subject by the NN less representative of his/her movement, and may ultimately degrade the identification performance of the algorithm. To ameliorate this, we propose an identification algorithm that jointly considers the outputs of the tracking block and of the classifier, as detailed in Section IV-I.

Considering that the TCPCN classifier is applied consistently to every track t at every time step k, in the following we simplify the notation denoting the pre-processed input point-cloud sequence $\boldsymbol{Z}^t_{k-K+1:k}$, of length K, by $\boldsymbol{Z}_{1:K}$.

G. IDENTIFICATION – TEMPORAL CONVOLUTION POINT-CLOUD NETWORK

The proposed classifier is designed to extract meaningful features from a temporal sequence of point-clouds, which is obtained as a result of the detection and tracking steps. The proposed architecture includes two processing blocks, termed *point-cloud* block (PC) and *temporal convolution* block (TC), and we refer to the full neural network as *temporal convolution point-cloud network* (TCPCN), see Fig. 2.

1) POINT-CLOUD BLOCK

A number K of identical (same weights) feature extraction blocks is applied to the standardized input point-clouds, $\boldsymbol{Z}_i$, $i = 1, \ldots, K$, of size $n_{\max} \times 5$, i.e., each composed of $n_{\max}$ reflecting points $\boldsymbol{p}_r$ (see Section III-B). Each of such blocks implements a function $f_{\boldsymbol{W}}(\cdot)$, obtained as the cascade of a multi-layer perceptron (MLP) [34] followed by a global average pooling operation, where $\boldsymbol{W}$ is a set of weights to be learned. Each reflecting point $\boldsymbol{p}_r$ in point-cloud $\boldsymbol{Z}_i$ (a vector of size 1×5), is fed to the first MLP layer and is independently processed from all the other $n_{\max}$ points in $\boldsymbol{Z}_i$, by one of $n_{\max}$ parallel branches. The MLPs located at the same depth share the same weights across all the points: there are 3 *fully-connected* (FC) layers with 96 units followed by 2 FC layers with 192 units. Each FC layer applies a linear transformation of the input followed by an exponential-linear unit (ELU) activation function [35]. Batch normalization is used after each linear transformation [36] and right before the following non-linearity (ELU). The output feature vector from the last MLP layer from each branch has size 1×192. Global average pooling reduces this set of features to a single feature vector, $\boldsymbol{o}_i = f_{\boldsymbol{W}}(\boldsymbol{Z}_i)$, of size 1×192, by taking the average of each element across all the 100 parallel branches. The structure of function $f_{\boldsymbol{W}}(\cdot)$ is loosely inspired by the popular PointNet [15]. The key aspect of $f_{\boldsymbol{W}}(\cdot)$ is that it uses functions that are invariant to the ordering of the input points, by sharing the weights of the MLP and using suitable pooling operations. This ensures robustness and generality, because point-clouds that only differ in how the points are ordered will result in the same output. We underline that our TCPCN significantly differs from PointNet as the latter is designed to perform end-to-end classification and segmentation of *dense* 3D point clouds, whereas our $f_{\boldsymbol{W}}(\cdot)$ performs feature extraction from *sparse* 5D point-clouds.

2) TEMPORAL CONVOLUTION BLOCK

The sequence of feature vectors $\boldsymbol{o}_{1:K} = \{f_{\boldsymbol{W}}(\boldsymbol{Z}_i)\}_{i=1:K}$, each of dimension 192, is then fed to the TC block, which operates along the temporal dimension applying a function $h_{\boldsymbol{U}}(\cdot)$, where $\boldsymbol{U}$ is another set of weights. To extract temporal features efficiently, $h_{\boldsymbol{U}}(\cdot)$ contains temporal convolutions, which are a type of convolutional neural network (CNN) layer [34] where the input is convolved with a uni-dimensional filter (or *kernel*) of learned weights in order to recognize temporal patterns. The output of the filters is organized into so called *feature maps*, which become more and more complex and abstract with the depth of the layer. In TCPCN we use *causal dilated convolutions* [16], [37]. This technique consists *(i)* in masking the filters in such a way that neurons corresponding to a certain time step only depend on neurons corresponding to past time steps, i.e., they cannot use future information, as done in [16], and *(ii)* in applying the convolution filters skipping blocks of $\delta - 1$ samples in the input,

where δ is the so-called *dilation rate*. Formally, denoting a feature map as $\mathbf{m}$ and the filter as $\mathbf{k}$, the output of a dilated convolution, $*_\delta$, between $\mathbf{m}$ and $\mathbf{k}$ is [37],

$$(\mathbf{m} *_\delta \mathbf{k})(s) = \sum_{i+\delta j=s} \mathbf{m}(i)\mathbf{k}(j). \tag{19}$$

The standard discrete convolution is obtained for $\delta = 1$. In the proposed TCPCN we employ 3 temporal convolution layers with filters of dimension 3 (also called *kernel* dimension) and dilation rates of 1, 2 and 4, respectively. The applied filters are repeated along the feature vector components of the input, obtaining 32, 64 and 128 feature maps at each layer, respectively.

The last layer of TCPCN is a temporal convolution layer that maps the extracted temporal features onto Q feature maps, each corresponding to one of the output classes. It applies a standard convolution with a kernel size of 3 and it is followed by a global average pooling to group the information from each feature map and obtain a single vector of dimension Q. Finally, a SoftMax function is applied, defined for a generic vector $\boldsymbol{x}$ as $\text{SoftMax}(\boldsymbol{x})_i = e^{x_i} / \sum_j e^{x_j}$. The vector outputted by this last layer is denoted by $\hat{\boldsymbol{y}} = \text{SoftMax}(h_{\boldsymbol{U}}(\boldsymbol{o}_{1:K})) = \text{TCPCN}(\boldsymbol{Z}_{1:K})$ and its q-th elements represents the probability that the input point-cloud sequence belongs to class q.

H. CLASSIFIER TRAINING AND INFERENCE

1) LOSS FUNCTION

The loss function used is the categorical cross-entropy, which is a standard choice in classification problems [34]. The CE compares the output of the last layer $\hat{\boldsymbol{y}}$ with the ground-truth identities of the subjects expressed in one-of-Q representation, $\boldsymbol{y}$: $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = -\sum_{q=1}^{Q} y_q \log(\hat{y}_q)$.

2) TRAINING

To train TCPCN, we used the Adam optimizer with learning rate $\eta = 10^{-4}$ [34]. The process is stopped once the loss function computed on a validation set of data stops decreasing, a technique called *early stopping*. Overfitting is a severe problem in the context of radar point-clouds: the high randomness of the detected points and the sensitivity to different environments make the learning task challenging, especially when generalization to unseen environments is required. To reduce overfitting, several strategies were utilized: *dropout* [38] was applied to the output of the PC blocks, randomly dropping components of the feature vectors with probability $p_{\text{drop}} = 0.5$, an L_2 regularization cost [34] on all network weights was considered, with parameter $\lambda_{L_2} = 10^{-4}$. The selection of the hyperparameters was carried out using a *greedy search* procedure.

3) INFERENCE

During the inference (or prediction) step, TCPCN is used to obtain classification probabilities for each maintained track, $\mathcal{T}_k^t \in \boldsymbol{\mathcal{T}}_k$, in the current time step k. We denote by $\tilde{\boldsymbol{y}}_k^t \in [0,1]^Q$ the vector that collects these probabilities. The prediction is carried out on a batch of T_k point-cloud sequences in parallel with a single pass of the data through the network, jointly obtaining $\{\tilde{\boldsymbol{y}}_k^t\}_{\mathcal{T}_k^t \in \boldsymbol{\mathcal{T}}_k}$. Moreover, we apply weight quantization, [39], to 8 bit integer values to reduce even further the inference time and the memory cost of the model. It is worth noting that, due to the use of convolutions, TCPCN has a low number of parameters: in the PC block the weights are shared among the $n_{\max}$ parallel branches, while the TC block is a *fully convolutional* neural network, with no fully connected layers. Fully convolutional networks are typically very fast in terms of training and inference time compared to fully connected or recurrent neural networks and have fewer parameters (further analysis is carried out in Section V-H).

I. IDENTIFICATION ALGORITHM

After obtaining the output probabilities for each track from TCPCN, several problems still have to be tackled: *(i)* obtaining stable classifications, robust to the fact that subjects may turn or move in unpredicted ways which do not carry their typical movement signature, *(ii)* finding a method to compensate for the missing frames when subjects go undetected, which can cause classification errors, *(iii)* dealing with the uniqueness of the subject identities, as classifying the subjects independently and solely based on $\tilde{\boldsymbol{y}}_k^t$ may lead to assigning the same identity to multiple targets. To address these problems, we devised the procedure detailed in Alg. 1, which uses both the output of the tracking procedure and the classification probabilities provided by TCPCN to estimate the identities of the subjects in a stable and reliable way. The procedure acts as follows.

1) At the first time step $k = 1$, a vector $\boldsymbol{y}_1^t$ of size Q is initialized for each track $\mathcal{T}_1^t \in \boldsymbol{\mathcal{T}}_1$, with all components equal to $1/Q$. $\boldsymbol{y}_1^t$ represents a stabilized vector of probabilities for each track.
2) At the generic time step $k > 1$, $\boldsymbol{y}_k^t$ is updated using Alg. 1, according to one of the two following rules:
   a) if track $\mathcal{T}_k^t$ was detected in the most recent $K/2$ time-steps (line 1), TCPCN is applied to the corresponding sequence of point-clouds, obtaining the probability vector $\tilde{\boldsymbol{y}}_k^t$ (line 2). Hence, an exponentially weighted moving average procedure (line 6) is applied to mediate between the previous stable estimate $\boldsymbol{y}_{k-1}^t$ and the newly computed one $\tilde{\boldsymbol{y}}_k^t$, obtaining a new stable estimate $\boldsymbol{y}_k^t$ (normalized so that its elements sum to one, see line 7).
   b) if track $\mathcal{T}_k^t$ was not detected in at least one of the most recent $K/2$ time steps (line 8), $\boldsymbol{y}_k^t$ is obtained as $\gamma \boldsymbol{y}_{k-1}^t$ with $\gamma < 1$ (line 9). In this way, we maintain the last reliable identification, but we progressively lower the confidence that we put on it over time. Note that after this step $\boldsymbol{y}_k^t$ does not longer resemble a probability distribution, as the sum of its elements is smaller than one.

Algorithm 1 Joint Identification at Time Step k

Input: Current set of tracks, $\boldsymbol{\mathcal{T}}_k$, smoothing parameter, ρ, decay parameter, γ.
Output: Identities $\mathcal{I}_k^t, \quad \forall \mathcal{T}_k^t \in \boldsymbol{\mathcal{T}}_k$.

1: Set $\boldsymbol{\mathcal{T}}_k^{(s)} = \left\{\mathcal{T}_k^t \in \boldsymbol{\mathcal{T}}_k \text{ s.t. } \mathcal{T}_k^t \text{ det. in the last } K/2 \text{ frames}\right\}$
2: $\left\{\tilde{\boldsymbol{y}}_k^t\right\}_{\mathcal{T}_k^t \in \boldsymbol{\mathcal{T}}_k^{(s)}} \leftarrow \text{TCPCN}\left(\left\{\boldsymbol{Z}_{k-K+1:k}^t\right\}_{\mathcal{T}_k^t \in \boldsymbol{\mathcal{T}}_k^{(s)}}\right)$
3: Initialize $\boldsymbol{Y}_k = \boldsymbol{0}_{T_k \times Q}$
4: **for** $\mathcal{T}_k^t \in \boldsymbol{\mathcal{T}}_k$ **do**
5: **if** $\mathcal{T}_k^t \in \boldsymbol{\mathcal{T}}_k^{(s)}$ **then**
6: $\boldsymbol{y}_k^t \leftarrow (1-\rho)\,\tilde{\boldsymbol{y}}_k^t + \rho \boldsymbol{y}_{k-1}^t$
7: normalize $\boldsymbol{y}_k^t$
8: **else**
9: $\boldsymbol{y}_k^t \leftarrow \gamma \boldsymbol{y}_{k-1}^t$
10: **end if**
11: $(\boldsymbol{Y}_k)_{t,:} \leftarrow \boldsymbol{y}_k^t$
12: **end for**
13: $\mathcal{I}_k^t \leftarrow \text{Hungarian}(\boldsymbol{Y}_k, p_{\text{conf}}), \ \forall \mathcal{T}_k^t \in \boldsymbol{\mathcal{T}}_k$

3) To assign identities to subjects without repetitions, we build a matrix of scores $\boldsymbol{Y}_k$ with all vectors $\boldsymbol{y}_k^t$ belonging to each track (line 11). We compute the best assignment of the identities using the Hungarian algorithm on $\boldsymbol{Y}_k$, which guarantees that the joint maximum score is attained with a one-to-one mapping (line 13).

To avoid associating a label to a track if the corresponding probability is very low, in the identification process we use a slightly modified version of the Hungarian algorithm, which behaves as follows: first, we compute the associations using the standard Hungarian algorithm. Hence, if the probability of a certain association is below $p_{\text{conf}} = 0.1$, we set the identity of the considered track to *unknown*. In Alg. 1, this modified Hungarian algorithm is indicated as Hungarian($\boldsymbol{Y}_k, p_{\text{conf}}$) to highlight that the result is a function of the score matrix $\boldsymbol{Y}_k$ and of the confidence threshold p_{conf}.

With Alg. 1, we jointly exploit the information from the classifier (vector $\tilde{\boldsymbol{y}}_k^t$) and the tracking step ($\boldsymbol{\mathcal{T}}_k^{(s)}$) to improve the identification performance.

Alg. 2 deals with errors in the tracking procedure, using the identity information available for each track. Tracking or association errors may happen during a blockage event involving two subjects (*blocker* and *blocked* in the following): for example a blocked subject may be erroneously associated when he/she becomes detectable again while being close to the blocker. These errors are dynamically corrected by analyzing the output of Alg. 1. Specifically, when the identity of a track $\mathcal{T}_k^t$ changes, we assume that this is an indication of a tracking error of the above-mentioned type (see line 2 of Alg. 2). In this case, this track is removed from the set of tracks that are maintained (line 4). At the same time, a new track $\mathcal{T}_k^j$ is initialized using the new identity $\mathcal{I}_k^t$, a new track index j (not yet used) and the current variables (state and covariance) associated with the old track $\mathcal{T}_k^t$ at time k (line 3). The new track $\mathcal{T}_k^j$ is then added to the set of maintained tracks

Algorithm 2 Tracking Error Correction at Time Step k

Input: Current set of tracks $\boldsymbol{\mathcal{T}}_k$.
Output: Updated set of tracks $\boldsymbol{\mathcal{T}}'_k$.

1: **for** $\mathcal{T}_k^t \in \boldsymbol{\mathcal{T}}_k$ **do**
2: **if** $\mathcal{I}_k^t \neq \mathcal{I}_{k-1}^t$ **then**
3: initialize new track $\mathcal{T}_k^j$ using $\mathbf{x}_k^t$, $\boldsymbol{P}_k^t$ and $\mathcal{I}_k^t$
4: $\boldsymbol{\mathcal{T}}'_k \leftarrow \left\{\boldsymbol{\mathcal{T}}_k \setminus \mathcal{T}_k^t\right\} \cup \{\mathcal{T}_k^j\}$
5: **end if**
6: **end for**

(line 4). Note that, the memory $\boldsymbol{Z}_{k-K+1:k}^t$ (past frames) is not attached to the new track, which is started anew.

V. EXPERIMENTAL RESULTS

In this section, we present results obtained by evaluating our tracking and identification method on

1) the mmGait dataset described in [22], available at https://github.com/mmGait/people-gait (Section V-A).
2) Our own dataset, featuring 8 subjects (Section V-B). This dataset was collected from our own measurements, implementing the proposed system on an NVIDIA Jetson TX2[3] board paired with a Texas Instruments IWR1843BOOST mm-wave radar[4] operating in the $77-81$ GHz band.

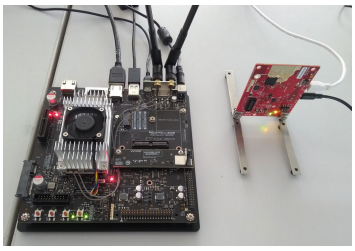
The Jetson board mounts an NVIDIA Tegra X2 GPU accelerator, the radar device is connected to it via USB and the communication is performed via UART ports, as shown in Fig. 3a. A camera was used to collect a video of the scene during the measurements and to label the dataset with the correct identities of the subjects. This setup poses some severe limitations on the amount of data that can be transferred in real-time to the NVIDIA processing device. Note that a more advanced solution such as an Ethernet connection would require additional hardware at an extra cost.[5] The full system has been implemented in Python, using the TensorFlow library for the neural network classifiers. In Tab. 2, the system parameters used in the evaluation are summarized.

A. EVALUATION ON THE mmGait DATASET

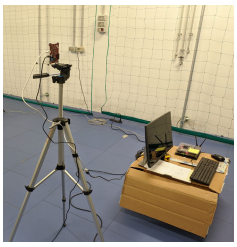
To assess the capabilities of TCPCN to effectively extract human gait features from point-cloud sequences, we test it on the publicly available mmGait dataset [22], which contains measurements from two different evaluation rooms, `room_1` and `room_2`, including respectively 23 and 31 different subjects. The dataset contains sequences where subjects are constrained to walk along straight lines in front of the radar, and other sequences where they walk freely.

Next, we present a comparison between our neural network classifier, TCPCN, and the CNN proposed by the authors of mmGait, denoted by mmGaitNet [22]. The accuracy results

[3]https://developer.nvidia.com/embedded/jetson-tx2
[4]https://www.ti.com/tool/IWR1843BOOST
[5]https://www.ti.com/tool/DCA1000EVM



(a) Jetson board (left) and radar (right).



(b) The mounted setup in the test room.

(c) Three subjects walking in the test room.

FIGURE 3. Overview of the experimental setup.

TABLE 1. Evaluation results on the mmGait dataset [22]. We report the accuracy (%) obtained by mmGaitNet according to the original paper [22] and the accuracy of our TCPCN, highlighting the best performance with a bold font. In the table, two columns show the results for linear and unconstrained motion. The dataset contains 29 subjects for `room_2` in the free motion case, and 30 in the linear motion case. The symbol "–" is used for those cases for which no accuracy value is provided in [22].

Setup		TCPCN (ours)		mmGaitNet [22]	
Room	# subj.	linear	free	linear	free
`room_1`	10	**92.07**	**70.31**	90.0	45.0
`room_1`	15	86.21	68.36	–	–
`room_1`	20	**83.37**	63.97	80.0	–
`room_2`	30/29	89.34	64.73	–	–

obtained by TCPCN on a superset of the tests conducted by the authors in [22] are shown in Tab. 1. We stress that, for the sake of a fair comparison, for these results we just compared TCPCN with the CNN of [22], without using our algorithms Alg. 1 and Alg. 2, as they would provide an additional performance increase.

For the results in Tab. 1, we consider the mmGait traces recorded by a single TI IWR6843[6] radar working in the $60 - 64$ GHz frequency band. The measurements for each subject are split according to a $80\% - 20\%$ proportion to obtain training and test sets, as done in [22].

TCPCN outperforms mmGaitNet in all the considered cases. The gap is particularly large in case the subjects walk freely: in this case, mmGaitNet reaches an accuracy of 45% on 10 subjects, as compared to an accuracy of 70.31% for TCPCN. This difference is due to the high variety of patterns that occur in the presence of unconstrained motion. TCPCN is more robust to such variability thanks to its invariance to the ordering of the points in the data cloud. The obtained performance on 30 subjects is encouraging, leading to identification accuracies as high as 89.34% and 64.73% for linear and unconstrained motion, respectively. This shows that gait-based identification systems employing mm-wave radar sensors hold the potential of scaling to scenarios where the number of users is in the order of a few tens. Finally, we point out that the accuracy with 30 subjects being higher than that with 15 and 20 is probably due to the fact that `room_2` contains subjects who are more easily distinguishable than those from `room_1`.

[6]https://www.ti.com/tool/IWR6843ISK

TABLE 2. Summary of the parameters of the proposed system.

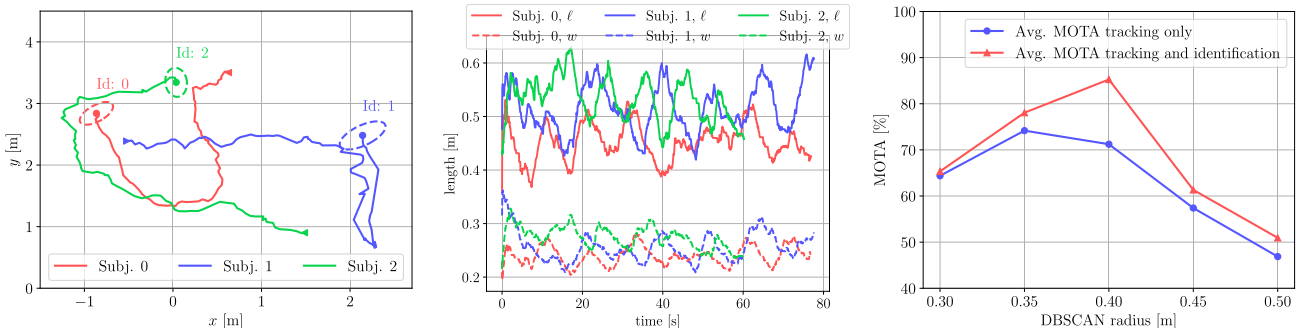
System parameters		
Antenna el. spacing	d	1.948 mm
Number of TX antennas	N_{TX}	3
Number of RX antennas	N_{RX}	4
Start frequency	f_o	77 GHz
Chirp bandwidth	B	3.072 GHz
Chirp duration	T	60 μs
Chirp repetition time	T_{rep}	68 μs
No. samples per chirp	M	256
No. chirps per seq.	L	64
Frame rate	$1/\Delta t$	14.92 fps
ADC sampling frequency	$1/T_f$	5 MHz
Range resolution	ΔR	4.88 cm
Velocity resolution	$\Delta \tilde{v}$	14.9 cm/s
DBSCAN radius	ε	0.4 m
DBSCAN min. cluster dim.	m_{pts}	10
Meas. range std	σ_R	0.03 m
Meas. az. angle std	σ_θ	$\pi/24$ rad
Meas. ext. std	$\sigma_{\tilde{\ell}}, \sigma_{\tilde{w}}$	0.05 m
Meas. orient. std	$\sigma_{\tilde{\xi}}$	$\pi/6$ m
Process noise std	σ_a	8 m/s^2
Process ext. std	σ_ℓ, σ_w	0.001 m
Process orient. std	σ_ξ	$\pi/24$ m
CJPDA bias term	β	0.01
m/n logic parameters	m/n	10/30
Max point-cloud dim.	$n_{\max}$	100
Input time-steps	K	30
Moving avg. parameter	ρ	0.99
Decay parameter	γ	0.999
Dropout probability	p_{drop}	0.5
Regularization parameter	λ_{L_2}	10^{-4}
Learning rate	η	10^{-4}

B. PROPOSED DATASET DESCRIPTION

To further validate the proposed system, we built our own dataset using four different rooms: three to collect training data and one for testing purposes. This arrangement of data and rooms was intentionally adopted to assess the generalization capabilities of the proposed system. Eight subjects were involved in the measurements, see Tab. 3.

1) TRAINING

the training rooms are two research laboratories, of size 8×8 meters and 8×3 meters, respectively, containing desks,



(a) Estimated trajectories and extensions of three subjects walking freely in the test room.

(b) Evolution of the extension estimates for the three subjects.

(c) Average MOTA using tracking only or joint tracking and identification.

FIGURE 4. Tracking system evaluation.

TABLE 3. Details on the subjects involved in the measurements.

Subject	Age	Height [m]	Sex	ℓ [cm]	w [cm]	Frames
0	26	1.63	F	43	22	33, 339
1	26	1.76	M	52	23	33, 514
2	25	1.85	M	52	24	36, 126
3	26	1.72	M	46	16	18, 668
4	28	1.69	M	45	22	19, 035
5	25	1.61	F	43	20	27, 674
6	63	1.77	M	50	24	22, 039
7	63	1.58	F	41	20	16, 925

furniture and technical equipment, and a furnished living room of size 8×5 meters. In the first room, due to space limitations, the area used for the training measurements is a rectangular space of size 3×5 meters. To collect the training data, one subject at a time walked freely for an amount of time ranging from 1 to 5 minutes. Note that, in all our measurements the subjects are allowed to cover a distance of up to 6 m from the radar, within its field-of-view of $\pm 60°$.

The measurement campaign was repeated across different days, acquiring from 20 to 40 minutes of data per subject. Taking into account different days, we aimed at reducing the effect of clothing or daily patterns in the way of walking. Prior to the actual training phase, the point-clouds data were pre-processed as described in Section IV-F and grouped into sequences of $K = 30$ consecutive frames, leaving an overlap of 20 frames between different sequences. To reduce overfitting, we artificially augmented the training data by applying random shuffling of the points in each point-cloud and adding random noise to each point, drawn from a uniform distribution in the interval $[-0.1, 0.1]$. To select the neural network hyperparameters, a portion of the training data (one sequence of approximately 2, 250 frames per target) was used as a validation set.

2) TEST

the test room is a 7×4 meters research laboratory, whose measurement area is free of furniture (see Fig. 3). We stress that, while training is performed on up to 8 *single subjects*, all our test sequences include multiple targets concurrently moving in the test environment. This leads to blockage events, i.e., when a subject occludes the line-of-sight (LoS) between the radar and another target, resulting in bursts of frames where the blocked subject goes undetected.

The measurement sequences contained in the test dataset are split as follows:

1) 10 sequences of 80 seconds (1, 200 frames) with 3 subjects. These are further split into 5 sequences where the subjects were constrained to walk following a linear movement at their preferred speed (back and forth across predefined linear paths), and 5 sequences where they could walk freely, following any trajectory in the available space, as shown in Fig. 3c. This leads to unpredictable trajectories that can cover the whole field-of-view of the radar sensor ($\pm 60°$) and distances up to 6 m. Moreover, in all our experiments user trajectories intersect frequently, leading to ambiguities in the data association, and making tracking more challenging.
2) 10 sequences of 80 seconds with 2 subjects, split into 5 sequences with a linear walking movement, and 5 sequences where the subjects walk freely.

C. TRACKING PHASE EVALUATION

In Fig. 4a, we show example trajectories followed by the three targets in one of the test sequences. In this experiment, the CM-KF succeeded in identifying and reconstructing the trajectory of each target, even in the presence of complex and strongly non-linear movement. The NN-CJPDA data association logic was found to be very robust, as long as the targets are correctly separated by DBSCAN into disjoint point-cloud clusters.

In all the test measurements the main difficulty faced by the system was that of handling blockage events that span over a large number of frames, e.g., more than $2 - 3$ second long. The number of such events increases significantly when more subjects are added to the monitored environment. We empirically assessed that, using a single radar sensor with

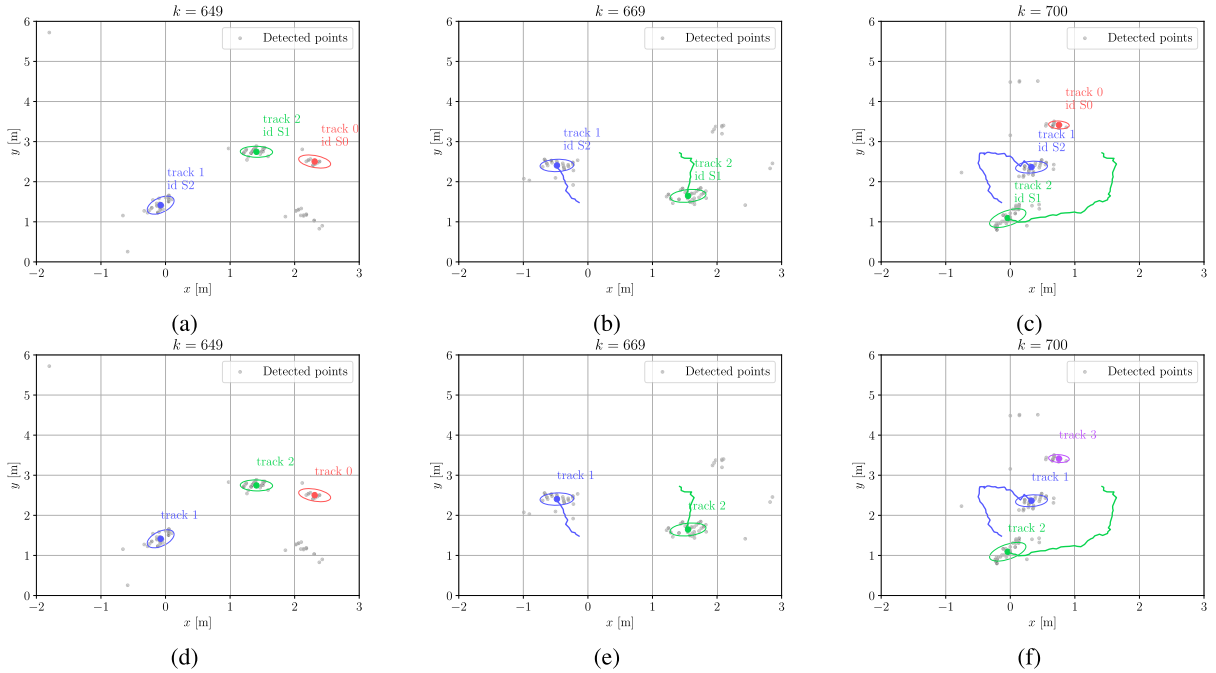


FIGURE 5. Proposed identification algorithm (a - b - c) compared to a standalone tracking approach (d - e - f) on the $x-y$ plane. Subject 0 (S0) is lost at time $k = 669$ and tracked again at time $k = 700$. By joint use of tracking and identification algorithms, the new track 3 is correctly re-associated with S0 (c), i.e., track 3 is mapped back onto track 0. Instead, the sole use of tracking would lead to the initialization of a new track for the same subject (f), causing a mismatch.

the resolution and communication capabilities considered in this work, going beyond three freely moving subjects at a time in such a small indoor environment leads to insufficient tracking and identification accuracy due to blockage. This is coherent with the findings in the literature, e.g. [22], where two radars placed in different locations were used to compensate for these facts.

Fig. 4b shows the results of the extension estimation across a full test sequence for all subjects. The expected shape enclosing a human target is correctly estimated: the ellipse axes are coherent with typical shoulder widths, ℓ, and thorax widths along the sagittal plane, w. The estimated value varies depending on the position of the target with respect to the radar: this is due to the fact that the received point-clouds contain a smaller number of points as the distance increases, due to propagation losses. Despite this fact, the average values are still proportional to the true subjects' extensions, as it can be checked by comparing Fig. 4b with Tab. 3.

To evaluate the capability of the proposed system towards tracking human subjects and the improvement brought by combining tracking and identification algorithms, we use the popular multiple object tracking accuracy (MOTA) metric [40]. The MOTA conveniently summarizes the ratio of missed targets (miss), false positives (fp) and track mismatches (mm), over the number of ground truth targets (gt) in each time frame k of the test sequence, formally,

$$\text{MOTA} = 1 - \frac{\sum_k \left(\text{miss}_k + \text{fp}_k + \text{mm}_k\right)}{\sum_k \text{gt}_k}. \qquad (20)$$

The value of gt_k was obtained from a reference video, as mentioned at the beginning of Section V.

In Fig. 4c, we show the MOTA obtained for different values of the DBSCAN radius, ε, for the NN-JPDA algorithm and our method, where NN-JPDA is used in conjunction with Alg. 1 and Alg. 2 (subject identification and label correction). Note that, with the standard NN-JPDA tracking algorithm, when a track is deleted and re-initialized, it is counted as a mismatch in Eq. (20), significantly lowering the MOTA. Moreover, data association errors can lead to track swaps when the trajectories of two subjects intersect. The MOTA obtained in this case is plotted as a blue curve in Fig. 4c. The red curve instead represents the improved MOTA, obtained by *(i)* merging together all the tracks associated with the same subject's identity, as described in Section IV-I, and *(ii)* correcting track swaps using Alg. 2. For the sake of clarity, in Fig. 5 we exemplify step *(i)*, which significantly improves the results by mitigating the effect of losing and re-initializing tracks.

From Fig. 4c, we see that for the optimal value $\varepsilon = 0.4$ m, the integration of tracking and identification provides an improvement of almost 20% in terms of MOTA. Remarkably, this is obtained at almost no additional complexity, by just feeding back the identity information to the tracking block.

D. ACCURACY RESULTS

In Tab. 4, we report the person identification accuracy obtained with the proposed method on the test sequences described in Section V-B. For the unconstrained walks,

TABLE 4. Accuracy and MOTA obtained with 2 and 3 subjects moving in the test room. We report the results both when the subjects follow linear trajectories ("linear") and when they move freely ("free"). With "Test x sub. train y" we denote the fact that the TCPCN used for the identification was trained on the single-target measurements of y subjects and tested on multi-target sequences containing x subjects simultaneously.

	Test 2 sub. train 3			Test 3 sub. train 3			Test 3 sub. train 8		
	linear	free		linear	free		linear	free	
[%]	Id. acc.	Id. acc.	MOTA	Id. acc.	Id. acc.	MOTA	Id. acc.	Id. acc.	MOTA
Seq. 1	98.67	99.61	98.71	100	99.67	99.06	96.95	92.35	99.06
Seq. 2	100	99.75	98.71	100	99.91	84.14	99.81	96.17	84.14
Seq. 3	95.26	96.91	86.42	100	91.79	94.11	100	88.14	90.19
Seq. 4	99.54	100	99.62	90.43	100	76.36	90.43	89.46	76.36
Seq. 5	99.34	100	97.96	99.37	92.44	72.61	97.04	92.02	72.61
Average	**98.56**	**98.98**	**96.28**	**97.96**	**96.76**	**85.26**	**96.85**	**91.62**	**84.47**

we also report the corresponding MOTA. The per-subject identification accuracy is computed using the time-steps in which the subject is correctly tracked, and is defined as the fraction of time-steps where a subject, besides being tracked, is also correctly identified. The final accuracy on a test sequence is obtained by taking the average accuracy on each subject, weighted by the total number of frames in which he/she is detected and tracked by the system.

In our tests, the number of subjects used for training is set as either 3 or 8 to assess how the system performs with an increasing number of targets. In Tab. 4, this is indicated with "Test x sub. train y", where x and y respectively refer to the number of subjects in the training set and those who are simultaneously present in the test data. The accuracy ranges from a maximum of 98.98% down to 91.62%, with the latter achieved for the most challenging case where 3 concurrent subjects have to be identified among a set of 8.

Differently from the results on mmGait (see Section V-A), there are no significant deviations in the identification performance between linear and unconstrained motion. This is due to the proposed identification algorithm, which lowers the effect of turns and non-linear movements that are likely to impact the classification accuracy. The MOTA is instead significantly lower with three targets, because of the more frequent blockage events (more misses and mismatches).

Fig. 6 shows the average accuracy obtained over the free-walking test sequences by *(i)* using the proposed solution (Alg. 1 and Alg. 2), *(ii)* using Alg. 1 only, *(iii)* using Alg. 1 without the Hungarian method, and *(iv)* identifying each subject at each time step k by solely using the point-cloud data at time k, and estimating the identity as $\arg\max \tilde{\mathbf{y}}_k^t$. For this evaluation the TCPCN was trained on 3 subjects. Note that with 2 subjects *(i)* and *(ii)* lead to about the same performance, but Alg. 2 leads to a slight improvement with 3 subjects, as tracking errors caused by track swaps due to blockage are more frequent in this case.

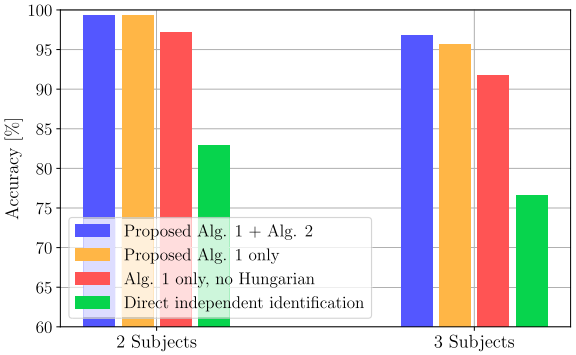


FIGURE 6. Accuracy of the proposed identification algorithm.

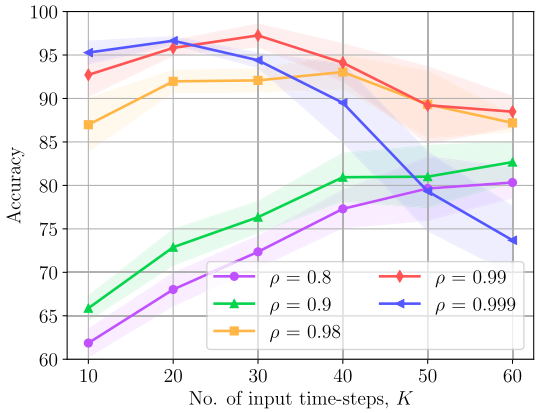


FIGURE 7. Effect of varying K and ρ on the identification accuracy.

E. IMPACT OF TEMPORAL FILTERING PARAMETERS

Now, we analyze the impact of K and ρ, i.e., the number of input time steps and the moving average smoothing parameter, respectively. These parameters are intimately connected, as they both control the dependence of the current output on past frames. In Fig. 7, we show the average accuracy computed on 10 different trainings of TCPCN with $Q = 3$ subjects, when tested on 3 subjects moving freely. The shaded areas represent 95% confidence intervals. In the abscissa, we vary K, plotting a different curve for several selected values of ρ. Lower values of ρ, e.g., 0.8 or 0.9, lead to a lower performance, as the memory of the moving average filter in these cases is too short to introduce stability in the classification (it corresponds to 5 and 10 time steps for $\rho = 0.8$ and 0.9, respectively) and high values of K

are required to get an accuracy beyond 80%. Increasing ρ has the effect of moving the point of maximum accuracy towards lower values of K. From our results, we recommend using $K = 30$ (two seconds of radar readings) and $\rho = 0.99$, as these values lead to the best average accuracy while keeping the system sufficiently reactive, with a moving average memory of approximately 100 time steps (between 6 and 7 seconds).

TABLE 5. TCPCN accuracy (no Alg. 1 and Alg. 2) on 8 single targets using: all the point-cloud features in p_r (all), selectively leaving out the received power information (no-P), the velocity (no-v), the z coordinate (no-z) or the $x-y$ coordinates (no-xy).

	all	no$-P$	no$-v$	no$-z$	no$-xy$
Acc. [%]	82.08	79.66	65.89	66.53	65.52

F. IMPORTANCE OF POINT-CLOUD FEATURES

In Tab. 5 we show the accuracy results of the sole TCPCN (no Alg. 1 and Alg. 2) considering 8 single targets, by leaving out some of the point-cloud features in $\boldsymbol{p}_r$. Specifically, we trained and tested the NN by selectively leaving out the received power (no-P), the velocity (no-v), the z coordinate (no-z) or the $x-y$ coordinates (no-xy). This evaluation provides insights on the importance of each of these features towards identifying the subjects. In particular, removing the velocity, $x-y$ or z coordinates led to the largest reduction in accuracy, suggesting that these carry the most useful information. In addition, Tab. 5 proves that our method mostly relies on movement-related features rather than on the reflectivity of the target (related to the received power). We remark that this is key to gain robustness to reflectivity changes due to different clothing or other environmental factors, and the lower importance of certain features is enforced by the learning procedure, which has automatically learned it by processing data from the same subjects across different days (wearing different clothes, etc.) and environments.

G. REAL-TIME IMPLEMENTATION REQUIREMENTS

Operating the proposed system in real-time poses constraints on the execution time of each processing block, and on the choice of the size and structure of the NN classifier. We measured the computation time needed by each block, respectively denoting by t_p the time needed to run the point-cloud extraction module running on the radar device (including the chirp sequence transmission, three DFTs along the fast time, slow time and angular dimension and the CA-CFAR detector), by t_c the time to transmit the data using the UART port, by t_t the execution time of the DBSCAN clustering algorithm, the CM-KF tracking step and the data association, and by t_i the inference time of the classifier. We found that while t_p is stable and strictly lower than 10 ms, t_c is highly variable, mostly because of the variable number of detected points in the scene, and ranges between 0 ms (when no points are detected) and 25 ms (with 3 subjects). The clustering and tracking take on average $t_t = 12$ ms with 3 subjects, with very low variance. Being the radar frame duration $\Delta t \approx 67$ ms, the identification step has meet the inequality $t_i < \Delta t - \max t_p - \max t_c - t_t \approx 20$ ms. In the next section, we present a comparison between the proposed approach and two works from the literature in terms of accuracy and inference time, taking these considerations into account.

H. COMPARISON WITH STATE-OF-THE-ART SOLUTIONS

Out of the two other approaches from the literature (see Section II), [22], does not obtain good results when subjects move freely, as neither a robust tracking method is implemented nor the identification information is used to improve the tracking performance, while [21] performs the identification in an *offline* fashion. In addition, they use different datasets. For these reasons, we chose to implement the classifiers from [22] and [21] and evaluate them on our multi-target test dataset using $K = 30$ input time steps and the same training data. As a baseline, we consider a model similar to TCPCN, but using a recurrent neural network (RNN) instead of temporal convolutions after the point-cloud feature extraction block. We refer to this model as PN + GRU in the following, as it is obtained combining a feature extraction block similar to PointNet with a *gated recurrent unit* layer (GRU) [41], which is capable of learning long-term dependencies. GRU cells maintain a hidden state across time, processing it together with the current input vector to learn temporal features in the input sequence (see [41] for a detailed description of GRU cells). In our implementation, we use a GRU layer with 128 hidden units.

TABLE 6. Comparison between TCPCN and other models from the literature in terms of training time and number of parameters.

Model	Training time [min]	No. of parameters
TCPCN (Ours)	13	153,711
PN + GRU	19	218,115
mm-GaitNet [22]	32	178,595
bi-LSTM [21]	63	3,237,379

In Tab. 6, we compare the learning models in terms of training time and number of parameters. This evaluation has been conducted on an NVIDIA RTX 2080 GPU for all the models. The training time is affected by the processing speed of each NN model and by the convergence time of the training process (number of training epochs). We note that the processing time of convolutional models (TCPCN and mm-GaitNet) is lower than that of recurrent ones (PN + GRU and bi-LSTM). However, training is significantly faster for the two models featuring the proposed point-cloud feature extractor (TCPCN and PN + GRU) due to faster convergence.

A comparison of accuracy and inference time, measured on the NVIDIA Jetson board, is presented in Fig. 8. The most accurate models in identifying the subjects are our TCPCN and PN + GRU. This shows the superiority of using a point-cloud feature extractor, due to its invariance to the ordering of the input points. TCPCN proves to be

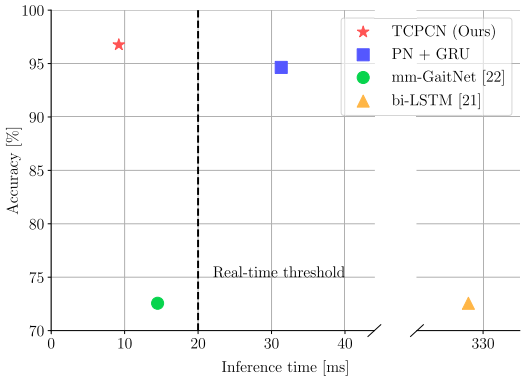


FIGURE 8. Performance comparison of the proposed TCPCN model against mm-GaitNet [22] and the bidirectional LSTM from [21]. As a baseline, we also evaluate a network similar to TCPCN that uses a GRU layer (PN + GRU) instead of temporal convolutions.

slightly better than PN + GRU, meaning that dilated temporal convolutions do not only improve the inference and training times but are also more effective in extracting temporal features. Through a vertical dashed line, we mark the maximum inference time for the algorithms to run in real-time on the Jetson device, i.e., 20 ms (see Section V-G): only two models satisfy this constraint, namely the proposed TCPCN and mm-GaitNet [22], which both exploit convolutions, as opposed to the RNN-based PN + GRU and bi-LSTM. In particular, TCPCN is the fastest model in making predictions, with an average inference time of 9.21 ± 2.12 ms.

VI. CONCLUDING REMARKS

In this work, we proposed a novel system that performs real-time person tracking and identification on an edge computing device using sparse point-cloud data obtained from a low-cost mm-wave radar sensor. The raw signal undergoes several processing steps, including detection, clustering and Kalman filtering for position and subject extension estimation in the $x - y$ plane, followed by a fast neural network classifier based on a point-cloud specific feature extractor and dilated temporal convolutions. Our system significantly outperforms previous solutions from the literature, both in terms of accuracy and inference time, being able to reliably run in real-time at 15 fps on an NVIDIA Jetson TX2 board, identifying up to three subject among a group of eight with an accuracy of almost 92%, while simultaneously moving in an unseen indoor environment.

Future research directions include the extension of the system to multiple radar devices, to deal with the frequent blockage events that can happen at mm-wave frequencies when multiple subjects move in the same physical environment. This would allow covering bigger spaces, while also getting better results in the presence of occlusions.

REFERENCES

[1] S. A. Shah and F. Fioranelli, "RF sensing technologies for assisted daily living in healthcare: A comprehensive review," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 11, pp. 26–44, Nov. 2019.

[2] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Process. Mag.*, vol. 34, no. 2, pp. 22–35, Mar. 2017.

[3] N. Knudde, B. Vandersmissen, K. Parashar, I. Couckuyt, A. Jalalvand, A. Bourdoux, W. De Neve, and T. Dhaene, "Indoor tracking of multiple persons with a 77 GHz MIMO FMCW radar," in *Proc. Eur. Radar Conf. (EURAD)*, Nuremberg, Germany, Oct. 2017, pp. 61–64.

[4] C. X. Lu, S. Rosa, P. Zhao, B. Wang, C. Chen, J. A. Stankovic, N. Trigoni, and A. Markham, "See through smoke: Robust indoor mapping with low-cost mmWave radar," in *Proc. 18th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, Toronto, ON, Canada, Jun. 2020, pp. 14–27.

[5] V. C. Chen, F. Li, S.-S. Ho, and H. Wechsler, "Micro-Doppler effect in radar: Phenomenon, model, and simulation study," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 1, pp. 2–21, Jan. 2006.

[6] V. C. Chen, "Analysis of radar micro-Doppler with time-frequency transform," in *Proc. 10th IEEE Workshop Stat. Signal Array Process.*, Pocono Manor, PA, USA, Aug. 2000, pp. 463–466.

[7] A. Nambiar, A. Bernardino, and J. C. Nascimento, "Gait-based person re-identification: A survey," *ACM Comput. Surv.*, vol. 52, no. 2, pp. 1–34, May 2019.

[8] B. Vandersmissen, N. Knudde, A. Jalalvand, I. Couckuyt, A. Bourdoux, W. De Neve, and T. Dhaene, "Indoor person identification using a low-power FMCW radar," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, pp. 3941–3952, Jul. 2018.

[9] Y. Yang, C. Hou, Y. Lang, G. Yue, Y. He, and W. Xiang, "Person identification using micro-Doppler signatures of human motions and UWB radar," *IEEE Microw. Wireless Compon. Lett.*, vol. 29, no. 5, pp. 366–368, May 2019.

[10] Z. Chen, G. Li, F. Fioranelli, and H. Griffiths, "Personnel recognition and gait classification based on multistatic micro-Doppler signatures using deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 669–673, May 2018.

[11] J. Pegoraro, F. Meneghello, and M. Rossi, "Multiperson continuous tracking and identification from mm-wave micro-Doppler signatures," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 2994–3009, Apr. 2021.

[12] A.-K. Seifert, M. G. Amin, and A. M. Zoubir, "Toward unobtrusive in-home gait analysis based on radar micro-Doppler signatures," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 9, pp. 2629–2640, Sep. 2019.

[13] M. S. Seyfioğlu, A. M. Özbayoğlu, and A. Z. Gürbüz, "Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 4, pp. 1709–1723, Aug. 2018.

[14] F. Jin, R. Zhang, A. Sengupta, S. Cao, S. Hariri, N. K. Agarwal, and S. K. Agarwal, "Multiple patients behavior detection in real-time using mmWave radar and deep CNNs," in *Proc. IEEE Radar Conf. (RadarConf)*, Boston, MA, USA, Apr. 2019, pp. 1–6.

[15] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 652–660.

[16] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," in *Proc. 9th ISCA Speech Synth. Workshop*, Sunnyvale, CA, USA, Sep. 2016, pp. 1–15.

[17] P. Cao, W. Xia, M. Ye, J. Zhang, and J. Zhou, "Radar-ID: Human identification based on radar micro-Doppler signatures using deep convolutional neural networks," *IET Radar, Sonar Navigat.*, vol. 12, no. 7, pp. 729–734, Jul. 2018.

[18] S. Abdulatif, F. Aziz, K. Armanious, B. Kleiner, B. Yang, and U. Schneider, "Person identification and body mass index: A deep learning-based study on micro-Dopplers," in *Proc. IEEE Radar Conf. (RadarConf)*, Boston, MA, USA, Apr. 2019, pp. 1–6.

[19] A. Jalalvand, B. Vandersmissen, W. De Neve, and E. Mannens, "Radar signal processing for human identification by means of reservoir computing networks," in *Proc. IEEE Radar Conf. (RadarConf)*, Boston, MA, USA, Apr. 2019, pp. 1–6.

[20] V. Polfliet, N. Knudde, B. Vandersmissen, I. Couckuyt, and T. Dhaene, "Structured inference networks using high-dimensional sensors for surveillance purposes," in *Proc. Int. Conf. Eng. Appl. Neural Netw. (EANN)*, Crete, Greece, May 2018, pp. 16–27.

[21] P. Zhao, C. X. Lu, J. Wang, C. Chen, W. Wang, N. Trigoni, and A. Markham, "MID: Tracking and identifying people with millimeter wave radar," in *Proc. 15th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, Santorini Island, Greece, May 2019, pp. 33–40.

[22] Z. Meng, S. Fu, J. Yan, H. Liang, A. Zhou, S. Zhu, H. Ma, J. Liu, and N. Yang, "Gait recognition for co-existing multiple people using millimeter wave sensing," in *Proc. AAAI Conf. Artif. Intell.*, New York, NY, USA, Feb. 2020, pp. 849–856.
[23] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
[24] M. A. Richards, J. Scheer, W. A. Holm, and W. L. Melvin, *Principles of Modern Radar*. Raleigh, NC, USA: Scitech Publishing, 2010.
[25] D. Lerro and Y. Bar-Shalom, "Tracking with debiased consistent converted measurements versus EKF," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, no. 3, pp. 1015–1022, Jul. 1993.
[26] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, Portland, OR, USA, Aug. 1996, pp. 226–231.
[27] T. Wagner, R. Feger, and A. Stelzer, "Radar signal processing for jointly estimating tracks and micro-Doppler signatures," *IEEE Access*, vol. 5, pp. 1220–1238, Feb. 2017.
[28] S. Bordonaro, P. Willett, and Y. Bar-Shalom, "Decorrelated unbiased converted measurement Kalman filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 2, pp. 1431–1444, Apr. 2014.
[29] G. Gennarelli, G. Vivone, P. Braca, F. Soldovieri, and M. G. Amin, "Multiple extended target tracking for through-wall radars," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6482–6494, Dec. 2015.
[30] J. W. Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 3, pp. 1042–1059, Jul. 2008.
[31] M. Feldmann, D. Franken, and W. Koch, "Tracking of extended objects and group targets using random matrices," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1409–1420, Apr. 2011.
[32] H. W. Kuhn, "The hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.
[33] R. J. Fitzgerald, "Development of practical PDA logic for multitarget tracking by microprocessor," in *Proc. Amer. Control Conf.*, Seattle, WA, USA, Jun. 1986, pp. 889–898.
[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[35] D. A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Juan, Puerto Rico, May 2016, pp. 1–14.
[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Lille, France, Jul. 2015, pp. 448–456.
[37] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent.*, San Juan, Puerto Rico, May 2016, pp. 1–13.
[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jun. 2014.
[39] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 2704–2713.
[40] K. Bernardin, A. Elbs, and R. Stiefelhagen, "Multiple object tracking performance metrics and evaluation in a smart room environment," in *Proc. 6th IEEE Int. Workshop Vis. Surveill., Conjunct With ECCV*, Graz, Austria, May 2006, p. 91.
[41] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1–15.

JACOPO PEGORARO (Graduate Student Member, IEEE) received the B.Sc. degree in information engineering and the M.Sc. degree in ICT for internet and multimedia engineering from the University of Padova, Padua, Italy, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree with the SIGNET Research Group, Department of Information Engineering. His research interests include deep learning and signal processing with applications to radio frequency sensing, and mm-wave radar sensing.

MICHELE ROSSI (Senior Member, IEEE) is currently a Full Professor of telecommunications with the Department of Information Engineering (DEI), University of Padova (UNIPD), Italy, teaching courses within the master's degree in ICT for internet and multimedia. He also sits on the Directive Board of the master's degree in data science offered by the Department of Mathematics (DM), UNIPD, for which he teaches machine learning and neural networks targeting the analysis of human data. From 2016 to 2018, he has been involved in the design of the IoT protocols exploiting cognition and machine learning, as part of INTEL's Strategic Research Alliance (ISRA) Research and Development Program. Since 2017, he has been the Director of the DEI/IEEE Summer School of Information Engineering. His research was supported by the European Commission through several H2020 projects, such as SCAVENGE under Grant 675891 on green 5G networks, MINTS under Grant 861222 on mm-wave networking and sensing, and GREENEDGE under Grant 953775 on green edge computing for mobile networks (project coordinator). His research interests include wireless sensing systems, green mobile networks, and edge and wearable computing. In recent years, he has been involved in several EU projects on the IoT technology, such as IOT-A under Project 257521 and has collaborated with companies, such as DOCOMO (compressive dissemination and network coding) and Worldsensing (IoT solutions for smart cities). In 2014, he was a recipient of the SAMSUNG GRO award with a project entitled Boosting Efficiency in Biometric Signal Processing for Smart Wearable Devices. He was a recipient of the seven best paper awards from the IEEE and currently serves on the Editorial Boards of the IEEE TRANSACTIONS ON MOBILE COMPUTING, and the Open Journal of the Communications Society.