Radar Signal Processing for Human Identification by Means of Reservoir Computing Networks

Azarakhsh Jalalvand*, Baptist Vandersmissen*, Wesley De Neve*‡, Erik Mannens*
*IDLab, Ghent University – imec, Belgium
‡Center for Biotech Data Science, Ghent University Global Campus, Korea

Abstract—**Along with substantial advances in the area of image processing and, consequently, video-based surveillance systems, concerns about preserving the privacy of people have also deepened. Therefore, replacing conventional video cameras in surveillance systems with less-intrusive and yet effective alternatives, such as micro-wave radars, is of high interest. The aim of this work is to explore the application of Reservoir Computing Networks (RCNs) to the problem of identifying a limited number of people in an indoor environment, leveraging gait information captured by micro-wave radar measurements. These measurements are done using a commercial low-power linear frequency-modulated continuous-wave (FMCW) radar. Besides the low quality of the outputs of such a radar sensor, walking spontaneously as opposed to controlled situations adds another level of complexity to the targeted use case. In this context, RCNs are interesting tools, given that they have shown a high effectiveness in capturing temporal information and handling noise, while at the same time being easy to setup and train. Using Micro-Doppler features as inputs, we follow a structured procedure towards optimizing the parameters of our RCN-based approach, showing that RCNs have a great potential in processing the noisy features provided by a low-power radar.**

I. Introduction

Surveillance systems typically make use of conventional sensors such as cameras to collect information about their surroundings, given that cameras are small, easy to install and inexpensive. Moreover, surveillance systems making use of cameras can benefit from decades of extensive research on the topic of image processing. Nevertheless, these systems are still not fully functional in low-light conditions or smoky rooms. Furthermore, increasing concerns about privacy are another reason to look for alternatives that are less intrusive. As an example, radar-based systems can still perform well in many situations where cameras fail, and thanks to recent advances in hardware development and design, they are becoming as interesting and popular as conventional sensors.

Using radar for the detection and identification of human targets and their activities is currently a topic of great interest, coming with many relevant applications. Autonomous cars, search and rescue operations, intelligent environments, border control, and building security are only some of the possible use cases. In recent years, many algorithms have been studied to process micro-Doppler (MD) signatures and to extract the most informative features for detecting and recognizing humans and their actions [1]. For example, to classify human actions, Kim *et al.* [2] have presented an approach using manual feature extraction and support vector machines (SVM). Furthermore, given the spectrogram shape of MD signatures, it is not surprising to observe that several convolutional neural network-based approaches have also been introduced for different use cases [3], [4], [5]. Also, in [6], different human actions have been modeled and classified by hidden Markov models (HMMs) in combination with principal component analysis (PCA).

With regard to identifying individual persons, a scenario has been proposed in [7], in which MD signatures have been used to identify thirteen subjects (seven males and six females), walking on a treadmill positioned in front of a radar device. Based on k-means clustering and k-nearest neighbors (k-NN), an accuracy of 100% is achieved on identifying these targets. In addition, the authors report an accuracy of 92.4% on gender classification. Moreover, in [3], a deep convolutional neural network (DCNN) is designed for identifying five persons. Using original and pre-processed radar data, an accuracy is obtained of 53.17% and 73.35%, respectively.

The aim of our work is to setup and employ RCNs for identifying humans based on their gait in an indoor environment, using MD signatures provided by a low-power FMCW radar. All of our training and validation experiments have been conducted on IDRad [3], a publicly available radar data set that consists of five persons walking spontaneously and individually in an indoor environment.

The rest of this paper is organized as follows. Section II gives an overview of reservoir computing networks. Next, Section III describes the characteristics of the MD signatures used in our work. The architecture of an RCN-based radar signal processing system, along with its setup, is described in Section IV. Section V presents our experimental setup and discusses the performance of radar and RCNs in terms of identifying humans. Our paper ends with a brief conclusion and ideas for future work.

II. Reservoir Computing Network (RCN)

In its simplest form, an RCN is a neural network with two different computational layers: (1) a *reservoir* of recurrently interconnected non-linear neurons, driven by inputs and by delayed feed-backs of its outputs and (2) an output layer of linear neurons, driven by the reservoir outputs (Fig. 1). A fundamental point is that the input weights and the recurrent connection weights are initialized randomly, and only the output weights are trained for solving the targeted problem.

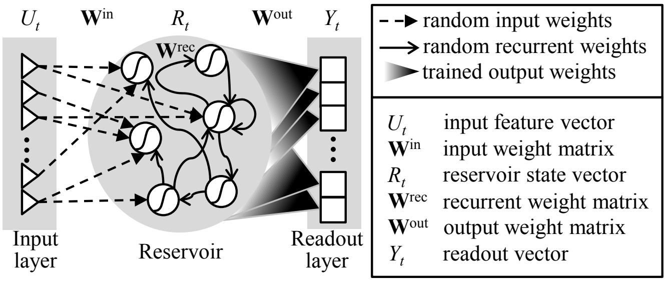

U_t
$\mathbf{W}^{\text{in}}$
R_t
$\mathbf{W}^{\text{out}}$
Y_t
$\mathbf{W}^{\text{rec}}$
random input weights
random recurrent weights
trained output weights
U_t input feature vector
$\mathbf{W}^{\text{in}}$ input weight matrix
R_t reservoir state vector
$\mathbf{W}^{\text{rec}}$ recurrent weight matrix
$\mathbf{W}^{\text{out}}$ output weight matrix
Y_t readout vector
Input layer
Reservoir
Readout layer

Fig. 1. A basic RCN consists of a reservoir and a readout layer. The reservoir is composed of interconnected non-linear neurons with fixed random weights. The readout layer consists of linear neurons with trained weights.

If U_t, R_t and Y_t represent the reservoir inputs, the reservoir outputs and the outputs (called *readouts*) at time t, then the RCN equations can be written as follows:

$$R_t = (1-\lambda)R_{t-1} + \lambda\, f_{res}(\mathbf{W}^{in}U_t + \mathbf{W}^{rec}R_{t-1}) \quad (1)$$
$$Y_t = \mathbf{W}^{out}R_t \quad (2)$$

with λ being a constant between 0 and 1, with f_{res} being the non-linear activation function of the reservoir neurons (we used *hyperbolic tangent* units in this work) and with $\mathbf{W}^{in}$, $\mathbf{W}^{rec}$ and $\mathbf{W}^{out}$ being the input, recurrent and output weight matrices, respectively. The constant λ is called the leak rate because Equation (1) represents a leaky integration of the neuron activation (if one makes abstraction of f_{res}).

The input features are normalized so that they have a zero mean and unit variance over the training examples. The weights of the hidden neurons are fixed by means of a random process that is characterized by four parameters [8]: (1) α_U, the maximal absolute eigenvalue of the input weight matrix $\mathbf{W}^{in}$, (2) ρ, also known as *spectral radius*, the maximal absolute eigenvalue of the recurrent weight matrix $\mathbf{W}^{rec}$, (3) K^{in}, the number of inputs driving each reservoir neuron and (4) K^{rec}, the number of delayed reservoir outputs driving each reservoir neuron. The first two parameters control the relative importance of the inputs and the delayed reservoir outputs in the reservoir neuron activation. The latter two control the sparsity of the input and the recurrent weight matrices. Any effective reservoir should at least have the so-called echo state property, stating that with time, the reservoir should forget the initial state it was in. It was shown in [9] that the echo state property holds if ρ, the spectral radius of the recurrent weight matrix, is smaller than 1.

The aim of the training is to find the output weights that minimize the mean squared difference between the readouts Y_t and their desired values D_t across N^{frm} available training examples. Introducing the matrices $\mathbf{R}$ and $\mathbf{D}$ with columns R_t and D_t, respectively, the output weights are the solution of a regularized Tikhonov regression problem [10]:

$$\hat{\mathbf{W}}^{out} = \underset{\mathbf{W}^{out}}{\arg\min}\left(\frac{1}{N^{frm}}\left|\left|\mathbf{W}^{out}\mathbf{R}-\mathbf{D}\right|\right|^2 + \epsilon\ \left|\left|\mathbf{W}^{out}\right|\right|^2\right) \quad (3)$$

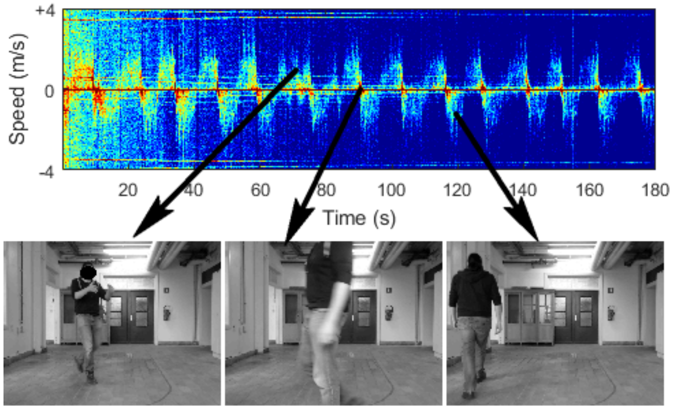


Fig. 2. Example recorded MD signatures. The signatures have been thresholded for better visualization.

with ϵ being the regularization parameter, preventing overfitting to the training data. The solution is obtained in a closed-form [11] as

$$\hat{\mathbf{W}}^{out} = (\mathbf{R}\mathbf{R}^T + \epsilon\, \mathbf{I})^{-1}(\mathbf{D}\,\mathbf{R}^T), \quad (4)$$

with $\mathbf{I}$ representing the identity matrix and $\mathbf{A}^{-1}$ the Moore-Penrose pseudo inverse of $\mathbf{A}$ [12].

III. Micro–Doppler

We conduct experiments on a dataset that has been collected with a 77-GHz multi-channel LFMCW radar platform. This radar platform is based on Infineon's monolithic microwave integrated circuits (MMICs) [13] and on baseband technology from Inras GmbH [14]. Such radars are currently mainly used for automotive purposes. The advantage of these radars is that they can be produced at a low cost and that they are relatively power efficient. This power efficiency usually comes at the cost of having a low signal-to-noise ratio (SNR), which is one of the challenges faced in this study. Fig. 2 shows a number of example MD signatures, clearly displaying a low SNR due to the combination of a low-power radar with the low radar cross-section of humans.

For each of the five persons in the dataset used, Fig. 3 contains an example MD signature of 10 seconds. The different distinguishable signals represent the body, the arms and the legs swinging of each person. Also, the zero Doppler bins contain the reflections of all static objects in the room, with these objects typically producing a stronger activation than a human in the room.

IV. Methodology

In this section, we explain the architecture of the proposed RCN-based model for processing the MD signatures. Specifically, we describe the main components of this model, namely (1) developing the input features, (2) the reservoir topology and (3) the readout layer.

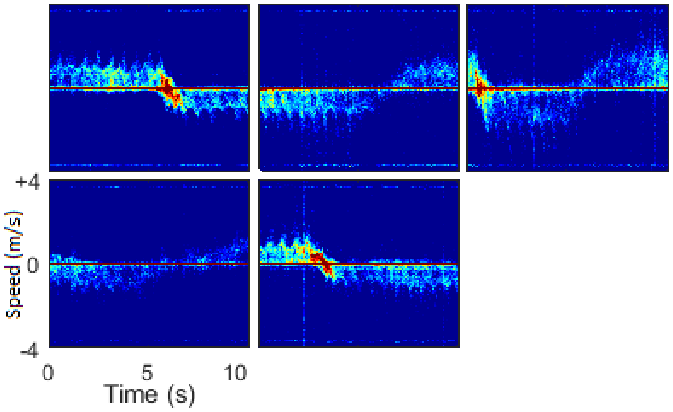


Fig. 3. Five example MD signatures with a duration of 10 seconds each (one randomly selected MD signature is shown for each person in the dataset used).

A. Pre-processing the Input Features

An FMCW radar produced by INRAS is used in SISO mode for creating the IDRad dataset [3]. A micro-Doppler map is calculated by first determining a range-Doppler map using a two-dimensional Fourier transform and by subsequently summing the absolute values over the range dimension. This leads to a vector of 256 Doppler features at each time step (called frames).

In order to boost the dynamic nature of the RCN and to enrich the temporal information, we can stack several subsequent frames or add the first and second derivatives of the inputs. These approaches enlarge the dimension of the input vectors. However, since the input weights are initialized randomly, thus not trained, increasing the size of the input vectors has no influence on the complexity of the training step.

B. Reservoir Topology

We aim at developing a multi-layer RCN that is obtained by stacking multiple RCNs. The input of each layer is the output of the previous layer and the layers are trained one after another using the same desired outputs in every layer.

The argument for cascading layers is that a new layer can learn and correct some of the mistakes made by the preceding layers because it offers additional temporal modeling capacity and a new inner space in which to perform the classification.

One could also benefit from bi-directional processing [15], by using two reservoirs in the RCN. As illustrated in Fig. 4, the forward reservoir processes the inputs $U_{1\to T}$, whereas the backward reservoir processes the inputs $U_{T\to 1}$. The outputs of the latter reservoir are then time-reversed before combining them with the outputs of the forward reservoir.

C. Readouts

The readout layer of the system encompasses N_p nodes corresponding to the number of people whose data are used to develop the model. The score of a each person is then obtained by accumulating the corresponding readout over time.

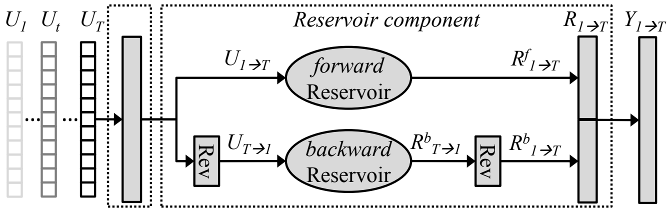


Fig. 4. Architecture of an RCN leveraging bi-directional processing in the reservoir component.

V. Experimental Results

In this section, we assess the performance of our RCN-based approach as a function of different input setups, reservoir size and the depth of the reservoir. Moreover, we also investigate whether or not the bidirectional scanning of the inputs is beneficial. In order to perform a fair comparison between the uni- and bidirectional reservoirs, a bidirectional reservoir of size N actually encompasses two independent reservoirs of size $N/2$ working in parallel. The number of trainable parameters of an RCN can be calculated by the number of persons in the dataset (N_p) times $(N+1)$, in which the extra 1 represents a bias for each readout node.

The performance is evaluated using the Frame Error Rate (FER; in %), which is the sum of the number of misclassified micro-Doppler columns (frames) divided by the total number of frames in the evaluation dataset.

A. IDRad Dataset

When creating the IDRad dataset, one major goal was to develop and publish samples of walking in an indoor environment that are as real as possible [3]. Therefore, the data recording has been spread out over multiple days and rooms, so to be able to take into account the effect of contextual influences like different static objects, clothing and shoes. By not focusing on a single recording per user in a single room, a robust system can be developed that is capable of dealing with different environments.

In a first stage, the random walking of five persons was recorded in a room for five consecutive minutes. Two weeks later, the same people were again recorded in the same room for 15 consecutive minutes. In a second stage, a different room has been used to create the validation and test set. Again, the recordings for both the validation and the test set contain five minutes of continuous walking for all targets and were created with two weeks in between. Fig. 5 shows the training and validation/test room used. It is worth mentioning that the presence of a metallic and wooden closet in the test room potentially decreases the SNR.

All subjects are males between 23 and 32 years old with comparable postures. Their weights range from 60 to 99 kilograms and their heights range from 178 to 185 cm.

As mentioned before, each target is recorded in a continuous matter. Therefore, the recordings also contain other movements than regular walking, including turns, short stops and accidental moves.

Fig. 5. Pictures of the rooms that have been used to record the MD signatures for training (left) and evaluation (right).

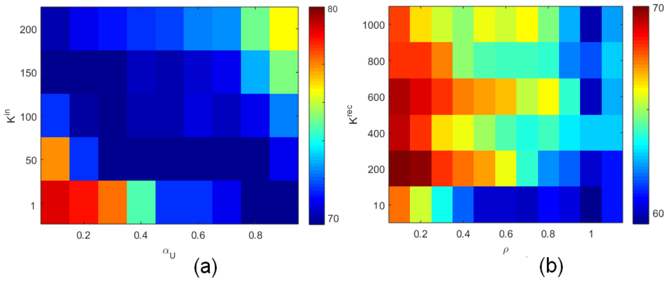


Fig. 6. FER (in %) on the validation set (a) as a function of α_U and K^{in} and (b) as a function of ρ and K^{rec}.

A video camera simultaneously recorded the walking targets. The video data are only used as a visual reference for the analysis of the MD signatures and the performance of the developed models.

B. RCN Hyperparameter Setup

The set of hyperparameters that need to be optimized consists of $(K^{in}, K^{rec}, \alpha_U, \rho, \lambda)$, along with the reservoir size. Setting up a suitable RCN has been studied in detail in [15] for the task of speech recognition. The empirical findings of that work show that there are simple and comprehensible rules which allow to design a reservoir in a structured manner, rather than performing a naive and time-consuming grid search over all parameter values:

1) In order to optimize the hyperparameters, one can begin with a rather small-sized reservoir.
2) The input and recurrent weight matrices ($\mathbf{W}^{in}$ and $\mathbf{W}^{rec}$) can be very sparse. In particular, 5 to 10 elements per node are enough, regardless of the size of the reservoir and the input feature vector.
3) ρ and α_U together control the relative importance of the input and recurrent neuron activation. Therefore, they can be tuned based on prior knowledge about the relation between these two types of activation (e.g., current input needs more weight than the past information) or by making use of a plain grid search.
4) The leak rate λ must be tuned based on $T_{\min}$, the minimum time (in scan steps) the reservoir output is expected to remain constant:
$$\ln(1-\lambda) = -1/T_{\min} \tag{5}$$
5) The last parameter to optimize is the size of the reservoir. In general, the randomly initialized weight matrices are an effective asset to avoid the over-fitting. Therefore, the maximum size of the reservoir depends on the size of the training data, hardware limitations and the ideal processing time.

C. Input Setup

In order to optimize the input layer, we begin with a moderately sized baseline RCN of 1000 nodes supplied with a 256-D normalized micro-Doppler feature vector (i.e., a frame), targeting the classification of five persons. Fig. 6(a) shows the performance of an RCN without recurrent connections as a function of α_U and K^{in}. The results confirm that the density of the input connections is indeed not a bottleneck as long as the input weight scale is chosen correctly. Also, the linear relation between these parameters suggests that, by changing one of them, the other one can be adjusted without sweeping.

Without going into the details, we also would like to mention that adding the first and second derivatives did not lead to significant performance improvements. The same finding holds true when we supplied the reservoir with a window of frames instead of only one frame. This means that these dynamic features do not contain new information that is significant in nature (that is, information that the RCN cannot derive itself from the original inputs).

D. Reservoir Setup

This sections covers the setup of the reservoir parameters, including the topology of the recurrent connections, as well as the selection of simple or leaky integration neurons.

As demonstrated by the previous experiments, the sparsity of the input weights does not significantly influence the performance, as long as the weight scale is chosen correctly. Fig. 6(b) depicts the outcome of a similar experiment on the recurrent weights. According to these results, (1) adding recurrent connections clearly improves the performance of the RCN and (2), regardless of the density of the reservoir, a spectral radius smaller but close to one is a good choice. In particular, it is interesting to observe that the sparser the network, the more stable it is against ρ. Indeed, in a sparse reservoir, the neurons are in general less influenced by recurrent activations.

λ can be estimated based on the average time interval in which the readouts are supposed to be constant. This can be interpreted as the minimum length the input should have for a meaningful classification. For the use case at hand, person identification is primarily performed based on walking style. Therefore, we claim that the minimum time needed to collect enough information is one complete period of walking, i.e., two walking steps. The average human walking speed at crosswalks is about $1.4\ m/s$, with an average step length of 0.76m [16]. Hence, a period of walking takes about 1.08s. Given the fact that the sampling rate of the radar device is 15fps, $T_{min} \approx 17 frames$, resulting in the following value for λ:

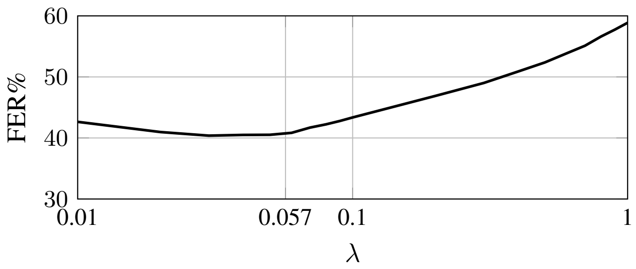


Fig. 7. FER (in %) on the validation set as a function of λ.

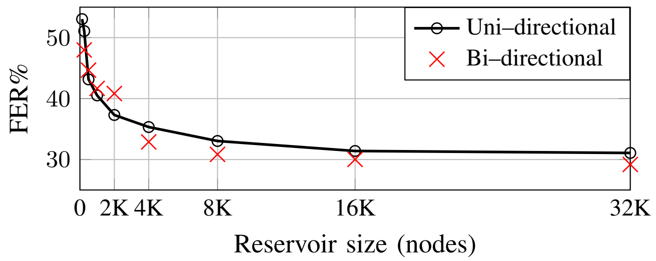
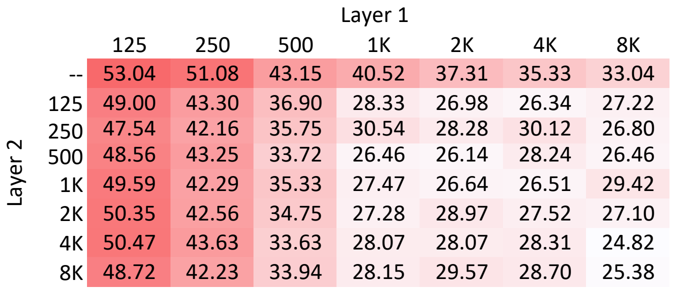


Fig. 8. FER (in %) as a function of the reservoir size (for a single-layer RCN), in both a uni-directional and bi-directional setup.



		Layer 1						
		125	250	500	1K	2K	4K	8K
Layer 2	--	53.04	51.08	43.15	40.52	37.31	35.33	33.04
	125	49.00	43.30	36.90	28.33	26.98	26.34	27.22
	250	47.54	42.16	35.75	30.54	28.28	30.12	26.80
	500	48.56	43.25	33.72	26.46	26.14	28.24	26.46
	1K	49.59	42.29	35.33	27.47	26.64	26.51	29.42
	2K	50.35	42.56	34.75	27.28	28.97	27.52	27.10
	4K	50.47	43.63	33.63	28.07	28.07	28.31	24.82
	8K	48.72	42.23	33.94	28.15	29.57	28.70	25.38

Fig. 9. FER (in %) on the validation set as a function of the reservoir size in the first and second layer.

TABLE I

FERs (in %) obtained with a 2-layer uni-directional RCN with 8K-500 nodes. The training, validation and test sets contain 99K, 22K and 24K frames, respectively.

Layer	Train set	Time (S)	Train	Valid	Test
1	Train	132	0.20	33.04	26.79
	Train + valid	155	0.30	0.06	23.63
2	Train	47	0.23	26.46	24.35
	Train + valid	59	0.27	0.13	22.76

$$\lambda = 1 - e^{(-1/T_{min})} \approx 0.057 \tag{6}$$

Fig. 7 shows the performance of our RCN as a function of λ, confirming that $\lambda = 0.057$, as suggested by the optimization strategy described above, is indeed a valid estimation.

Fig. 8 presents the performance of our RCN as a function of the reservoir size, in both a uni-directional and bi-directional setup. The results show that:

- Increasing the size of the reservoir constantly improves the performance, but the relative improvement becomes less significant in networks larger than 4K nodes.
- There is no sign of overfitting, even in a system with 32K nodes (≈160K trainable parameters).
- There seems to be no significant benefit in utilizing the bi-directional reservoir.

In order to study the impact of the second layer, we devised experiments with different reservoir sizes in the first and second layer. Given the experimental results shown in Fig. 9, we conclude that:

- Any single-layer model benefits from adding a second layer (on average 20%, relative), even if the extra layer is as small as only 125 nodes.
- In order to achieve promising results, the size of the first layer plays a role that is more crucial than the second layer.
- A continuous trend of improvement can be observed from a small model of 1×125 nodes up to a large model of 2×8K nodes, which confirms the resistance of RCNs against the overfitting trap.

Considering a trade-off between the performance and the complexity of the model, we opted for the 8K-500 RCN as our final model. Table I lists the performance of this system when it is trained only on training data, as well as the merger of the training and validation set. With a training error rate of less than 1%, it is obvious that this model is nicely capable of learning the given information. Comparing the validation error rate when the validation data are/are not included in the training set, one can conclude that an RCN succeeds in generalizing to the new environment, even though the validation set is as small as one-fourth of the training set. However, the error rate obtained for the test set does not drop as drastically as the error rate obtained for the validation set. This points to the need for a more in-depth investigation of the differences between the recorded data in the validation set and the recorded data in the test set.

In order to give the reader an idea about the time needed to train an RCN, Table I also depicts the training time (in seconds) of the first and the second layer using a single core of an Intel® Core$^{\text{TM}}$ i7–5930K CPU. According to Equation 4, the training procedure consists of (1) executing the reservoir on the training data to collect the activations (R), so to be able to calculate R^TR and R^TD and (2) solving the Moore-Penrose pseudo inverse of $(R^TR + \epsilon I)$. In that regard, the training time is linearly proportional to the size of the training set and quadratically to the size of the reservoir.

Table II compares the performance of the proposed method with some conventional approaches such as a Random Forest- and SVM-based model, along with a deep convolutional neural network, all trained on the same dataset [3]. We can observe that an RCN outperforms a deep CNN when the original MD signatures are used. It is interesting to observe that, only after significant feature engineering (that is, thresholding the input features empirically (Thr) and removing the static and

TABLE II

MODEL COMPARISON ON THE VALIDATION AND TEST SETS. THE INPUTS COLUMN DENOTES THE DIFFERENT INPUT MICRO-DOPPLER FEATURES, NAMELY, ORIGINAL, AFTER THRESHOLDING (THR), AFTER REMOVING SOME CHANNELS (RM), OR AFTER APPLYING PCA.

Model	Inputs	Valid	Test
D-CNN	Original	51.16	-
D-CNN	RM	46.35	-
D-CNN	Thr	28.46	-
D-CNN	Thr+RM	24.70	21.54
Random Forest	Thr+PCA	48.86	38.59
SVM	Thr+PCA	49.20	38.52
RCN (proposed)	Original	26.46	24.35

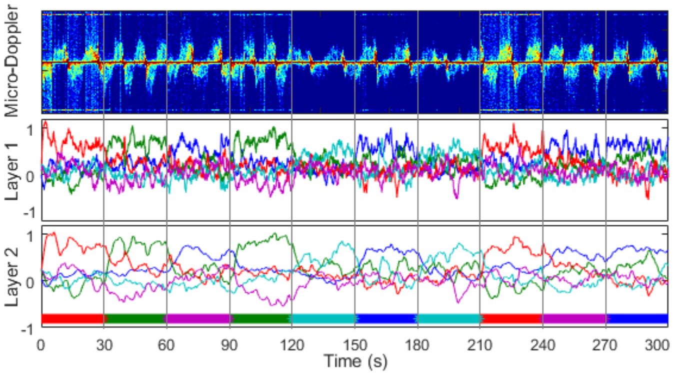


Fig. 10. Outputs of the first and the second layer of the RCN when the input is a random concatenation of signatures of the people in the dataset. The correct targets have been indexed by colors beneath the activation.

outer Doppler channels (RM)), the deep CNN model can outperform the RCN that has been trained on the original features. Moreover, we would like to mention that the proposed RCN model needs less than four minutes for training on this dataset.

Finally, Fig. 10 shows the readout of the first and the second layer when the input is a *concatenation* of 30-second chunks of examples *randomly chosen* from the test dataset. The plots show that, although the designed reservoir has a rather long memory ($\lambda = 0.057$), it reacts quickly to the change of the input from one person to another. Also, it shows that the first layer has some degree of uncertainty, although the correct readout is significantly higher than the others. The main role of the second layer seems to reduce this uncertainty, as the outputs are smoother than the first layer.

VI. CONCLUSION AND FUTURE WORK

In this paper, we investigated the application of reservoir computing networks (RCNs) for the purpose of radar signal processing. In particular, our focus was on the identification of people by leveraging human gait information available in micro-Doppler features. We described the important factors in designing our model and we showed that RCNs can effectively handle the low SNR radar data. Moreover, their capability in processing the temporal information, along with the easy training procedure, make them an appealing tool for this kind of time-series data processing.

Given that this work was a first step in the area of designing RCN-based models for processing micro-Doppler data, we believe that there is room to extend the work. For instance, it would be of interest to train an RCN-based model with range-Doppler information instead of micro-Doppler information. Also, combining Convolutional Neural Networks with RCNs, modeling the gait with Hidden Markov Models (HMM) and using an RCN to provide the HMM state probabilities are other directions for future work.

ACKNOWLEDGMENT

This research described in this paper was funded by Ghent University, imec, and the European Union.

REFERENCES

[1] R. M. Narayanan and M. Zenaldin, “Radar micro-Doppler signatures of various human activities,” *IET Radar, Sonar Navigation*, vol. 9, no. 9, pp. 1205–1215, 2015.

[2] Y. Kim and H. Ling, “Human activity classification based on micro-Doppler signatures using a support vector machine,” *IEEE Trans. Geoscience and Remote Sensing*, vol. 47, no. 5, pp. 1328–1337, 2009.

[3] B. Vandersmissen, N. Knudde, A. Jalalvand, I. Couckuyt, A. Bourdoux, W. D. Neve, and T. Dhaene, “Indoor person identification using a low-power fmcw radar,” *IEEE Trans. Geoscience and Remote Sensing*, vol. 56, no. 7, pp. 3941–3952, 2018.

[4] J. Park, R. J. Javier, T. Moon, and Y. Kim, “Micro-Doppler based classification of human aquatic activities via transfer learning of convolutional neural networks,” *Sensors*, vol. 16, no. 12, p. 1990, 2016.

[5] Y. Kim and T. Moon, “Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 8–12, 2016.

[6] M. O. Padar, A. E. Ertan, and Ç. ĝ. Candan, “Classification of human motion using radar micro-Doppler signatures with hidden markov models,” in *2016 IEEE Radar Conference (RadarConf)*, 2016, pp. 1–6.

[7] G. Garreau, C. M. Andreou, A. G. Andreou, J. Georgiou, S. Dura-Bernal, T. Wennekers, and S. Denham, “Gait-based person and gender recognition using micro-doppler signatures,” in *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2011, pp. 444–447.

[8] M. Lukoševičius, “A practical guide to applying echo state networks,” in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7700, pp. 659–686.

[9] H. Jaeger, “The ‘‘echo state’’ approach to analysing and training recurrent neural networks - with an erratum note,” GMD Report 148, German National Research Center for Information Technology, Tech. Rep., 2001. [Online]. Available: http://www.faculty.jacobs-university.de/hjaeger/pubs/EchoStatesTechRep.pdf

[10] C. M. Bishop, “Training with noise is equivalent to tikhonov regularization,” *Neural Computation*, vol. 7, no. 1, pp. 108–116, 1994.

[11] D. W. Marquardt and R. D. Snee, “Ridge regression in practice,” *The American Statistician*, vol. 29, no. 1, pp. 3–20, 1975.

[12] R. Penrose, “A generalized inverse for matrices,” *Proc. The Cambridge Philosophical Society*, vol. 51, pp. 406–413, 1955.

[13] H. J. Ng, R. Stuhlberger, L. Maurer, T. Sailer, and A. Stelzer, “Low phase noise 77-ghz fractional-n pll with dll-based reference frequency multiplier for FMCW radars,” in *European Microwave Integrated Circuit Conference*, 2011, pp. 196–199.

[14] A. Haderer, “INRAS products–radarbook,” INRAS GmbH, Tech. Rep., 2014.

[15] A. Jalalvand, F. Triefenbach, K. Demuynck, and J.-P. Martens, “Robust continuous digit recognition using reservoir computing,” *Computer Speech and Language*, vol. 30, no. 1, pp. 135 – 158, 2015.

[16] B. J. Mohler, W. B. Thompson, S. H. Creem-Regehr, H. L. Pick, and W. H. Warren, “Visual flow influences gait transition speed and preferred walking speed,” *Experimental Brain Research*, vol. 181, no. 2, pp. 221–228, 2007. [Online]. Available: https://doi.org/10.1007/s00221-007-0917-0