Quantifying Uncertainty in Real Time with Split BiRNN for Radar Human Activity Recognition

Lorin Werthen-Brabants*, Geethika Bhavanasi, Ivo Couckuyt, Tom Dhaene, Dirk Deschrijver
IDLab, Ghent University - imec, Belgium
*lorin.werthenbrabants@ugent.be

***Abstract*— Radar systems can be used to perform human activity recognition in a privacy preserving manner. Deep Neural Networks are able to effectively process the complex radar data and make predictions. Often these networks are large and do not scale well when processing a large amount of radar streams at once, for example when monitoring multiple rooms in a hospital. This work proposes Bayesian Split Bidirectional Recurrent Neural Network for Human Activity Recognition. Using this technique the processing of data is split in two parts, one part on-premise (low-power, low-cost device), and the other off-premise (high power device). The proposed approach leverages the power of the off-premise device to quantify its uncertainty, and to gain more information on its epistemic and its aleatoric parts. Results indicate the proposed approach is able to correctly identify parts of a prediction that either need more training data for better predictions (epistemic uncertainty), or are inherently hard to classify by the model (aleatoric uncertainty).**

***Keywords*— radar, human activity recognition, bidirectional rnn, edge-cloud interaction, uncertainty quantification, bayesian neural networks**

I. Introduction

Real-time activity recognition in a hospital room or nursing home is important, because it can help to detect troublesome events, such as the fall of a patient, as soon as possible. This is most meaningful for geriatric patients [1], [2] that are more likely to suffer lasting injuries from a fall, especially if treatment is delayed. Using radar, a privacy-preserving method to detect falls can be established. A popular technique to perform fall detection, or human activity recognition is *Deep Learning* (DL). Data-driven DL models require high training times and powerful machines at prediction time. When this is deployed at scale to handle many radars at once, the cost of using DL increases significantly.

Split BiRNNs [3] were introduced to combat the requirements needed to handle many streams radars at once. A two-staged model, designed to work on separate devices with distinct Micro-Doppler [4] streams, facilitates live monitoring of human activities in many different environments simultaneously, for example in a hospital with many rooms. The two-stage design flow is as follows:

1. An edge device computes the class predictions on a stream of incoming radar frames, using a lightweight model. This enables streaming of results in real-time to those who need immediate feedback from the model.
2. Another more capable device (able to process large amounts of data in a batch) applies a backward model on intermediate computations made by the edge devices and improves the predictions. Any inaccurate predictions made by the edge device are rectified by this more capable device.

In this work, the backward model is extended using Monte Carlo Dropout (MCD) [5] to predict a distribution of possible activities, rather than a single activity at any given time. This way, both *epistemic* and *aleatoric* uncertainty can be calculated. The epistemic uncertainty describes how uncertain the model is about a given prediction, while the aleatoric uncertainty expresses the uncertainty inherent to the data. These can be used to evaluate the model at prediction time, and to identify highly uncertain events, or to detect when the model needs retraining if many uncertain events keep occurring.

II. Background

A. *Split BiRNN*

The Split BiRNN technique [3] consists of two linked networks called *forward branch* and *backward branch.* To update predictions as new data flows into the forward branch, the unidirectional nature of Recurrent Neural Networks (RNN) is used. These feed a state from one time step to the next, and only require the previous state and the current input to calculate the next state. The state of a forward RNN is denoted as $\overrightarrow{\mathbf{h}}^{(t)}$, with t denoting the time step corresponding to this hidden state. f is the non-linear function (LSTM [6], GRU [7], …) applied to the input $\mathbf{x}^{(t)}$ at time t, parametrized by $\boldsymbol{\theta}$ and n refers to the corresponding layer.

$$\overrightarrow{\mathbf{h}}_{n+1}^{(t)} = f(\mathbf{x}^{(t)}, \overrightarrow{\mathbf{h}}_{n}^{(t-1)}; \boldsymbol{\theta}). \tag{1}$$

This means that given time t, state $\mathbf{h}^{(t-1)}$ is the only state necessary to calculate state $\mathbf{h}^{(t)}$. This is different when making use of a Bidirectional RNN (BiRNN). In this case, an RNN layer is added that processes the input data in reverse. The resulting hidden states of the forward and backward RNN layers are then concatenated. This means the output is dependent on previous and following states when using a bidirectional RNN. The concatenation function between vectors is denoted as $+\!\!+$. Concatenation is used in this work to maximize the amount of data fed into the backward branch, but another addition operator could be used instead to save memory, or could be a learned operator. The backward RNN state is denoted as $\overleftarrow{\mathbf{h}}^{(t)}$. $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$ are used to highlight the different parametrization between the forward and backward RNNs.

$$\mathbf{h}_n^{(t)} = f(\mathbf{h}_{n-1}^{(t)}, \overrightarrow{\mathbf{h}}_n^{(t-1)}; \boldsymbol{\theta}_\mathbf{1}) \mathbin{+\!\!+} f(\mathbf{h}_{n-1}^{(t)}, \overleftarrow{\mathbf{h}}_n^{(t+1)}; \boldsymbol{\theta}_\mathbf{2}). \quad (2)$$

In the Split BiRNN model, the calculation of the forward and the backward state is split. The general deep case is noted as $\overrightarrow{\mathbf{h}}_0^{(t)} = \overleftarrow{\mathbf{h}}_0^{(t)} = \mathbf{x}^{(t)}$, while the parameters $\boldsymbol{\theta}$ are omitted for notational convenience.:

$$\overrightarrow{\mathbf{h}}_n^{(t)} = f(\overrightarrow{\mathbf{h}}_{n-1}^{(t)}, \overrightarrow{\mathbf{h}}_n^{(t-1)}) \quad (3)$$

$$\overleftarrow{\mathbf{h}}_n^{(t)} = f(\overrightarrow{\mathbf{h}}_{n-1}^{(t)} \mathbin{+\!\!+} \overleftarrow{\mathbf{h}}_{n-1}^{(t)}, \overleftarrow{\mathbf{h}}_n^{(t+1)}) \quad (4)$$

$$\mathbf{h}_n^{(t)} = \overrightarrow{\mathbf{h}}_n^{(t)} \mathbin{+\!\!+} \overleftarrow{\mathbf{h}}_n^{(t)}. \quad (5)$$

The forward branch of the network follows the regular structure of a deep RNN with a fully connected layer for predictions, as seen in Figure 1.

The backward branch differs from regular deep BiRNN networks because it is run separately from the forward branch, but shares the same weights in the final fully connected layer. In a regular BiRNN, the output of the forward and the backward RNNs would be concatenated and fed into the next layer. The model differs in that the forward RNN always operates without knowledge of the output of the backward layer. This has the advantage that both branches can be jointly trained, reducing the full model size. This is not necessary, as a separate fully connected layer could be trained instead.

The states $\overrightarrow{\mathbf{h}}_n^{(t)}$ calculated in the forward branch are reused when calculating the backward branch. Figure 2 shows how they are concatenated, as described in (5).

B. *Uncertainty Quantification using Monte Carlo Dropout*

Uncertainty of a prediction consists of two major components: the *epistemic* uncertainty and the *aleatoric* uncertainty. The epistemic uncertainty encodes *how many* models fit the data, given the model architecture. Hence, low (high) entropy describes high (low) agreement between different models and their predictions, respectively. The entropy of a sampled distribution is calculated as

$$H[p(x)] = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x), \quad (6)$$

describing how *surprising* the distribution is.

a) Epistemic Uncertainty: The epistemic uncertainty displayed by a model parameterized by $\boldsymbol{\theta}$ and trained on dataset $\mathcal{D}$ is measured by the posterior probability distribution: $p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$, as derived from Bayes' Rule. This posterior distribution exhibits high entropy if a model was not trained on enough data to explain the observation. This type of uncertainty is therefore also referred to as *reducible*, as more training data reduces this uncertainty. As a consequence, with a sufficiently complex model, a highly entropic posterior distribution for a certain input likely means the data point is out of distribution. Applying this to Human Activity Recognition (HAR), this uncertainty will be high when making a prediction on actions with diverse ways of execution (e.g. cleaning, cooking, ...), where not all possible variations of the activity are captured in the dataset. This may also occur when little samples of a given class are available in the dataset.

b) Aleatoric Uncertainty: The aleatoric uncertainty expressed by a model is encoded in the conditional distribution of a class, given an input and a set of weights: $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$. It describes the inherent ambiguity in the data and cannot be explained away by introducing more data. An example of aleatoric uncertainty arising is when the model learns that some data points are inherently hard to classify. This may be the case when activities are similar in nature, such as sitting down on a bed or sitting down on a chair.

c) Monte Carlo Dropout on Split BiRNN: The backward branch of the aforementioned Split BiRNN can be adapted to perform MCD [5]. This means the output of the network is no longer a single point, but rather an ensemble of models describing a predictive posterior distribution $p(\mathbf{y}|\mathbf{x})$. Following [8], the total predictive uncertainty of an ensemble, such as the one obtained by training and performing predictions with dropout [9], is found by averaging the predictions of the ensemble and calculating the resulting entropy, where M describes the amount of samples from the MCD distribution:

$$u_t(\mathbf{x}) = -\sum_{y \in \mathbf{y}} \left(\frac{1}{M} \sum_{i=1}^{M} p(y|\boldsymbol{\theta}_i, \mathbf{x}) \right) \log_2 \left(\frac{1}{M} \sum_{i=1}^{M} p(y|\boldsymbol{\theta}_i, \mathbf{x}) \right). \quad (7)$$

The resulting uncertainty estimate $u_t(\mathbf{x})$ also includes the epistemic uncertainty about the network weights $\boldsymbol{\theta}$. Fixing a single set of weights removes this uncertainty. Therefore, the expectation over the entropies of these distributions for an ensemble model

$$u_a(\mathbf{x}) = -\frac{1}{M} \sum_{i=1}^{M} \sum_{y \in \mathbf{y}} p(y|\boldsymbol{\theta}_i, \mathbf{x}) \log_2 p(y|\boldsymbol{\theta}_i, \mathbf{x}). \quad (8)$$

is a measure of aleatoric uncertainty. Finally, the epistemic uncertainty is simply calculated as the difference

$$u_e(\mathbf{x}) = u_t(\mathbf{x}) - u_a(\mathbf{x}). \quad (9)$$

III. Experiments

A. *Dataset and Model*

a) Dataset: The model is trained and evaluated on the PARRad dataset[1] [10], [3]. In total, 22 hours of radar data are split into two subsets: Homelab and Hospital, both (simulating) hospital rooms. This work focuses on the Hospital subset contained in PARrad, containing 13359 activities spread over 9 combined classes, see Table 1. The Hospital dataset is comprised of 20 test subjects performing different activities in 4 different 10-minute sessions.

This dataset contains Micro-Doppler (MD) signatures recorded with an off the shelf Texas Instruments (TI) xWR14xx radar with center frequency of 77 GHz, and a TI xWR68xx radar with center frequency of 60 GHz. In this paper, MD

[1] Publically available at https://sumo.intec.ugent.be/radar

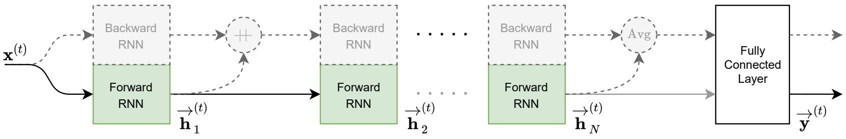
$\mathbf{x}^{(t)}$
Backward RNN
Forward RNN
Avg
Fully Connected Layer
$\overrightarrow{\mathbf{h}}_1^{(t)}$
$\overrightarrow{\mathbf{h}}_2^{(t)}$
$\overrightarrow{\mathbf{h}}_N^{(t)}$
$\overrightarrow{\mathbf{y}}^{(t)}$

Fig. 1. Graphical model of the forward step. The backward step is grayed out.

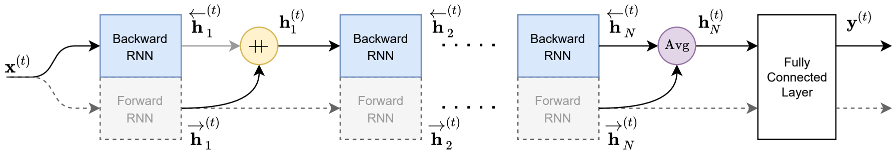
$\mathbf{x}^{(t)}$
Backward RNN
Forward RNN
$\overleftarrow{\mathbf{h}}_1^{(t)}$
$\mathbf{h}_1^{(t)}$
$\overleftarrow{\mathbf{h}}_2^{(t)}$
$\overleftarrow{\mathbf{h}}_N^{(t)}$
Avg
$\mathbf{h}_N^{(t)}$
Fully Connected Layer
$\mathbf{y}^{(t)}$
$\overrightarrow{\mathbf{h}}_1^{(t)}$
$\overrightarrow{\mathbf{h}}_2^{(t)}$
$\overrightarrow{\mathbf{h}}_N^{(t)}$

Fig. 2. Graphical model of the backward step. The grayed out lines and blocks in the forward step indicate that this step is no longer necessary because the values are pre-computed in the forward step, and can be reused.

Table 1. Classes available in PARrad, including the transformations made by combining different classes.

Activity	Samples
Walk	149880
Fall on the floor	33435
Stand up	99050
Sit down	55010
Get in bed	29665
Lie in bed (stop moving)	23040
Roll in bed	30555
Sit in bed (from lying down)	21120
Get out bed	32225

signatures [4] over time are used, which capture speed relative to the radar sensor. These are placed in different corners of the hospital room, and capture activities simultaneously. This means every capture session is effectively duplicated, albeit with differences in the field of view of the radar sensors.

The captured MD signatures consist of 128 Doppler bins, which were averaged from the 93 range bins in their corresponding RD maps. These MD signatures are stacked over time, so there is a datapoint $\mathbf{X} \in \mathbb{R}^{t\times 128}$ for every recording in the dataset, where t varies from recording to recording. t is commonly around $t = 6666$, which corresponds to 10 minutes real time due to the frame length being $0.09\,\mathrm{s}$.

b) Model: The model is adopted from [3], which is a hybrid CNN-GRU approach [11], [12]. The latter part of the network has been extended with three extra fully connected layers with ReLU as non-linear activation function, on which dropout is applied with a rate of 0.2.

B. Discussion

Both the aleatoric (u_a) and epistemic uncertainty (u_e) are calculated on the test set. A single recording is chosen to evaluate the merit of uncertainty. Each prediction is sampled $M = 20$ times using dropout. The resulting weighted F_1-score of the MCD model is 0.890, while that of the normal model is 0.903, which is statistically insignificant. The predictions and their corresponding uncertainties over time can be seen in Figure 3.

Both the epistemic and aleatoric uncertainty are calculated for each time step in the recording. This makes it possible to identify activities that come with high uncertainty.

High aleatoric uncertainty can be seen at the edges of activities. This is likely due to the inherent ambiguity that exists when transitioning from one activity to another, and the continuous nature of the problem. Longer events of aleatoric uncertainty may be of higher interest, such as what is seen in the beginning of the predictions. There the predictions are more erratic, and four classes seem to be considered simultaneously.

It can be seen that areas with high epistemic uncertainty are highly informative, as they point out classes and situations where the model needs more data to improve its certainty, being the bed activities. Aleatoric uncertainty however needs to further be filtered, as every transition between activities contains relatively high uncertainty values. Ignoring these short transient periods, high aleatoric uncertainty can be seen at the activities performed in bed. These activities in the PARrad dataset contain more fine-grained movements than the rest. These are all performed in a single location in the hospital room, and constitute the smallest portion of the dataset.

Thresholds could be set on both quantities of uncertainty. On the one hand, a lot of aleatoric uncertainty might inform the user that the actions are too hard to distinguish. This either because the actions themselves are not well defined, or because the sensor may not capture it accurately. On the other hand, high aleatoric uncertainty means the model needs more data. Identifying the activities which need more data would improve model accuracy.

IV. Conclusion and Future Work

In this work, a Bayesian interpretation of Split BiRNNs is proposed that uses Monte Carlo Dropout. As a result, both epistemic and aleatoric uncertainty can be quantified. It is shown that these uncertainties correspond to respectively inherent ambiguities present in the data, and the activities for which little data is available.

This work only handles the uncertainty quantification of the backward branch, but the forward branch may be equally important. However, due to the many evaluations needed for a

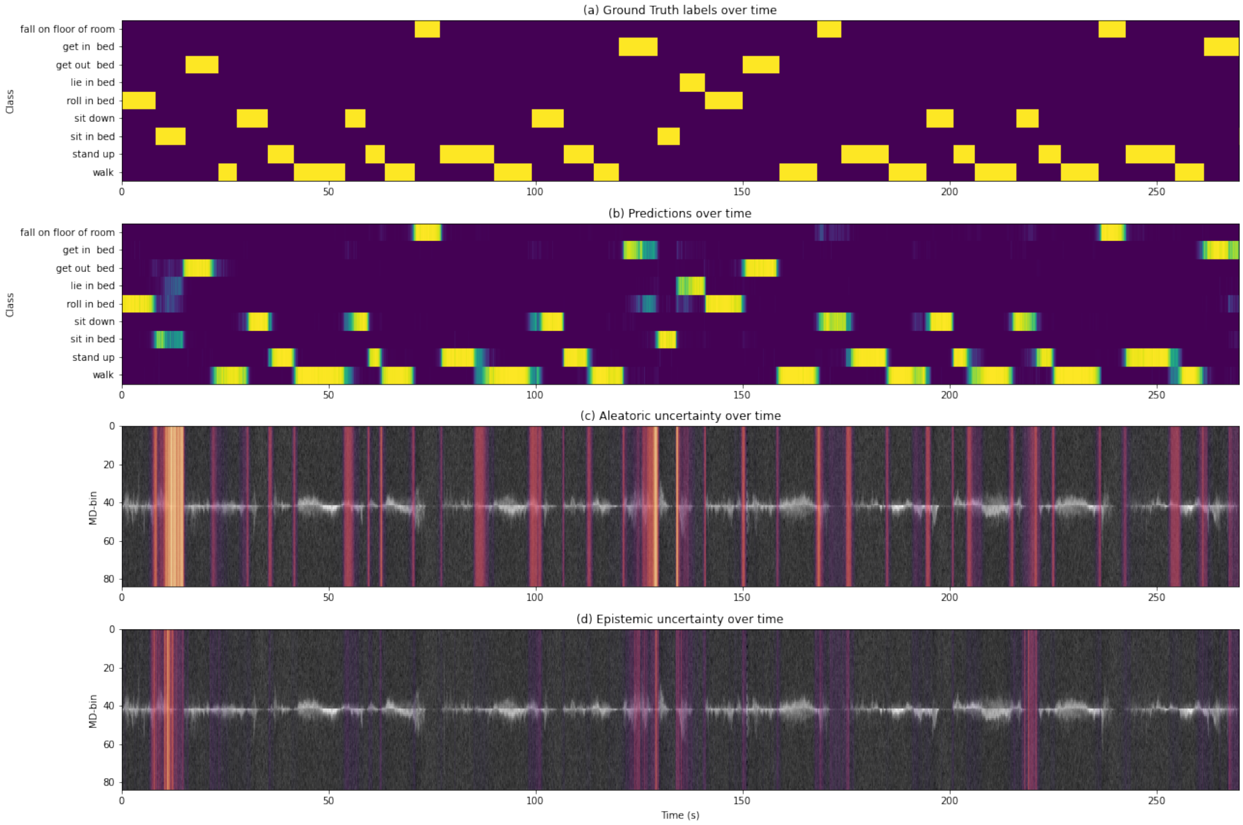

Fig. 3. (a) Ground truth labels over time, yellow regions denote the occurence of a captured activity (b) Predictions over time between 0 and 1, with the yellow shade denoting 1 (c) Aleatoric uncertainty expressed as entropy in bits overlaid on the original MD over time (d) Epistemic uncertainty expressed as entropy in bits overlaid on the original MD over time

single prediction, MCD is not suitable on the on-premise device as it increases the workload. As future work, an investigation of efficient Bayesian techniques will be conducted to allow the on-premise device to decide when to send a prediction to the off-premise device, lowering the amount of computation needed by the entire system.

Acknowledgment

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

[1] G. F. Fuller, "Falls in the Elderly," *American Family Physician*, vol. 61, no. 7, p. 2159, Apr. 2000.

[2] X. Yu, "Approaches and principles of fall detection for elderly and patient," in *HealthCom 2008 - 10th International Conference on e-Health Networking, Applications and Services*, Jul. 2008, pp. 42–47.

[3] L. Werthen-Brabants, G. Bhavanasi, I. Couckuyt, T. Dhaene, and D. Deschrijver, "Split BiRNN for Real-Time Activity Recognition using Radar and Deep Learning," *Scientific Reports*, vol. 12, no. 1, pp. 1–11, 2022.

[4] V. C. Chen, F. Li, S. S. Ho, and H. Wechsler, "Micro-Doppler effect in radar: Phenomenon, model, and simulation study," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 2–21, Jan. 2006.

[5] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of the 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, Jun. 2016, pp. 1050–1059.

[6] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[7] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Sep. 2014, pp. 1724–1734.

[8] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Machine Learning*, vol. 110, no. 3, pp. 457–506, Mar. 2021.

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[10] G. Bhavanasi, L. Werthen-Brabants, T. Dhaene, and I. Couckuyt, "Patient Activity Recognition Using Radar Sensors and Machine Learning," *Neural Computing and Applications*, pp. 1–16, 2022.

[11] B. Vandersmissen, N. Knudde, A. Jalalvand, I. Couckuyt, T. Dhaene, and W. De Neve, "Indoor human activity recognition using high-dimensional sensors and deep neural networks," *Neural Computing and Applications*, vol. 32, no. 16, pp. 12 295–12 309, Aug. 2020.

[12] J. Zhu, H. Chen, and W. Ye, "A Hybrid CNN–LSTM Network for the Classification of Human Activities Based on Micro-Doppler Radar," *IEEE Access*, vol. 8, pp. 24 713–24 720, 2020.