WaveCounter

Passage detection and direction classification using CSI time-series

Christian Gustavsson, Henrik Ahlinder, Joel Henneberg, Martin Dahl, Sebastian Andersson, Tiger Kylesten

ISY
Communication Systems
Linköping University

December 16, 2022

Table of Contents

Problem description

Solution
- Solution overview
- Hardware for CSI estimation
- Data collection
- Feature pre-processing
- Detection of passage
- Classification of passage

Usage

Results

Problem description

- **Goal:** Detect passage and classify direction of movement through door.
- **Equipment:** One pair of Pluto SDR radios.
- **Data:** Estimated CSI (Channel State Information) time series of one wireless channel.

Problem description

Pluto SDR radios

- Two SMA connectors for instrumentation and antennas,
- 300 MHz - 3.8 GHz radio frequency coverage,
- 200 kHz - 20 MHz channel bandwidth and
- USB2 port (480 Mb/s at 100% utilization) with the possible modes: network device, USB serial device, mass storage device.

Figure: Picture of the ADALM-PLUTO units.

Problem description

Experimental setup

- Door of room in ISY KS corridor at Linköping university.
- Room and corridor had furniture.

Figure: Picture of what the setup looks like.

Problem description

Principle

- A wireless channel $y[n] = h[n]x[n] + w[n]$[1], $w[n] \sim \mathcal{CN}(0, \sigma)$ [1].
- Walking through the door disturbs the channel, changing h (The CSI).
- The changes in the CSI will be different for walking in compared to walking out.
- This difference in CSI can be modelled using machine learning.

[1] For notational simplicity we assume no channel delay

Problem description

Example of CSI estimation time series

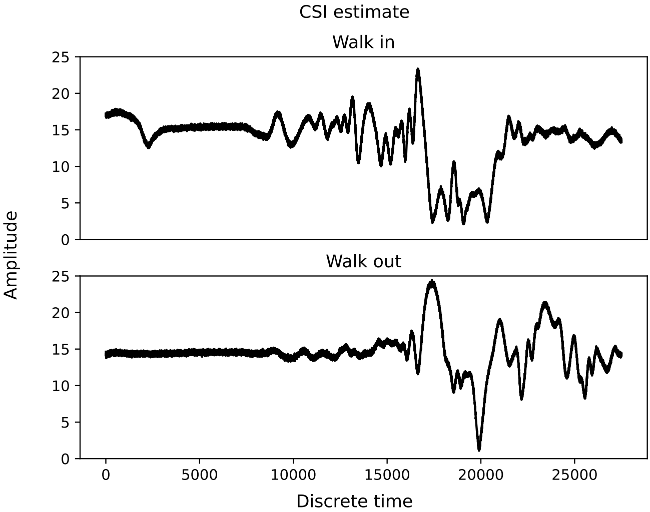

Figure: Output of hardware subsystem, 27500 estimates of the CSI.

Solution overview

1. Setup of hardware (Pluto Devices) for CSI estimation
2. Data collection, walking in and out 4508 times ...
3. Passage detection of CSI live-feed
4. Direction classification of CSI snippet
5. Integrate with graphical user interface

Hardware for CSI estimation

Setup

- Center frequency $f_c = 915\text{MHz}$
- Sample frequency $f_s = 550\text{khz}$
- Both Pluto devices connected to computer for synchronization

Hardware for CSI estimation

1. **CSI estimation**
   1. Transmitter sends $x[n] = 1$ for $n_0 \leq n \leq n_0 + 99$.
   2. Every 100 samples are used to estimate $|h|$ (CSI absolute value), this gives an estimation frequency of $f_s/100 = 5500\text{Hz}$.
   3. Explicitly, $|h|$ is estimated as in Equation 1 (note that $x[k] = 1$).

$$|\hat{h}|[n] = \frac{1}{100} \sum_{k=100n+1}^{100(n+1)} \left| \frac{y[k]}{x[k]} \right| \tag{1}$$

2. **Bias calibration**
   1. Receiver calculates $\hat{h}$ for 1 second worth of data.
   2. The static mean h_0 is calculated as the arithmetic mean of $\hat{h}$: $h_0 = \overline{\hat{h}}$

The final output is the time series $h_0[n] := |\hat{h}|[n] - |h_0|$, since the CSI in the meaning of the channel is not of actual interest.

Hardware for CSI estimation

Practical imperfections:

- Changes in surroundings can create serious issues with the model predictions.
- Sometimes the Pluto devices seemed to become warm after a few minutes of usage, which might cause an increase measurement noise. This means that data samples collected in the end of a collection series might be more noisy than the ones in the start.

Data collection

- A dataset $\mathcal{D} = \{(x, y) : x \in R^{1 \times d}, y \in \{0, 1\}\}$, where $|\mathcal{D}| = 4508$, $d = 27500$ was collected using the hardware sub-system.
- Specifically, each sample x is a 5 second time-series of CSI estimation at 5500 Hz, giving 27500 time-samples.
- y denotes the class, where 0 is in and 1 is out, with $\mathcal{D}$ having 2257 and 2251 samples of each, respectively.

Feature pre-processing

- Before input of samples $(x, y) \in \mathcal{D}$, the 27500 features in x needs to be pre-processed.
- The following tools were used for pre-processing: Butterworth low-pass filtering, Min-Max normalization, Z-score normalization, downsampling, Shift-Time DWT.
- The tools were used in two different combinations depending if model required 1D or 2D input.
- Note: From now on "CSI" refers to the CSI estimate time series.

Feature pre-processing

Unprocessed CSI

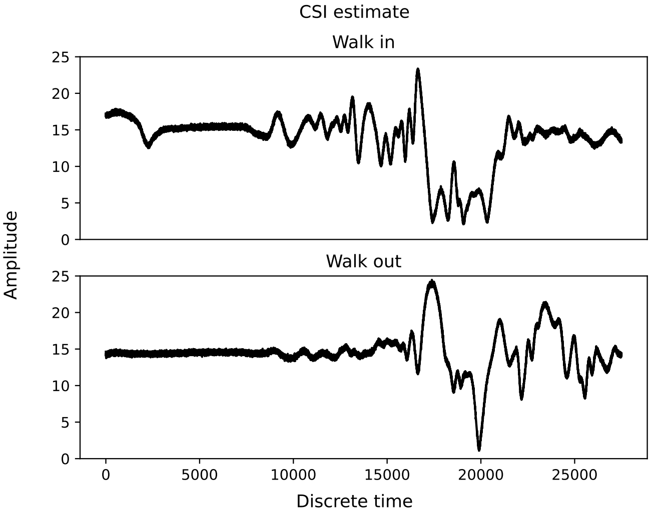

Figure: Output of hardware subsystem, 27500 estimates of the CSI.

Feature pre-processing

Butterworth filtered CSI

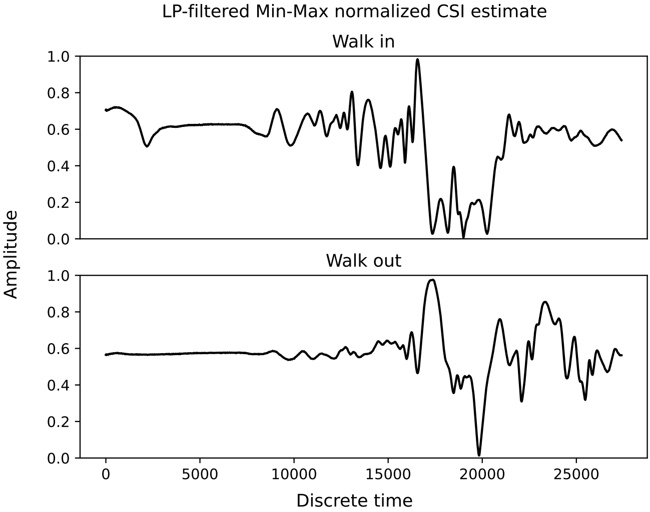

Figure: Low-pass filtered CSI. The CSI is also Min-Max normalized.

Feature pre-processing

Min-Max normalization
The Min-Max normalization of $x \in R^{n \times 1}, n \in N$ is described in Equation 2

$$\bar{x} = \frac{x - \min_i(x_i)}{\max_i(x_i) - \min_i(x_i)}, \; 1 \leq i \leq n. \tag{2}$$

Feature pre-processing

Z-Score normalization
The Z-score normalization of $x \in R$ is

$$\bar{x} = \frac{x - \hat{\mu}}{\hat{\sigma} + \varepsilon}, \tag{3}$$

where $\hat{\mu}$, $\hat{\sigma}$ are empirical estimates of the mean and standard deviation of the underlying distribution of x assumed to be normal.

Feature pre-processing

Downsampling

- Downsampling is used to decrease the dimensionality of the CSI for inputs with 1D input.
- The result, where the CSI has been downsampled from 27500 to 110 features can be seen in Figure 6.

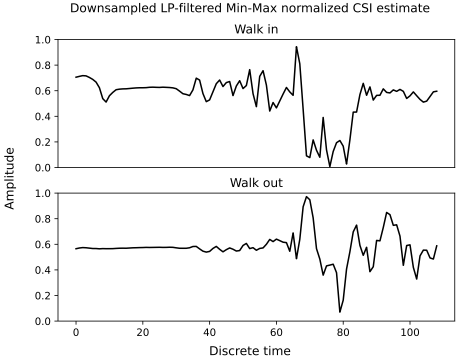

Figure: Downsampled, low-pass filtered CSI.

Feature pre-processing

DWT

- DWT is the Discrete Wavelet Transform.
- A Shift-Time DWT is used to transform the 1D 27500 dimensional CSI to a 12×25 size input for models requiring 2D input. We refer to this as "DWT".

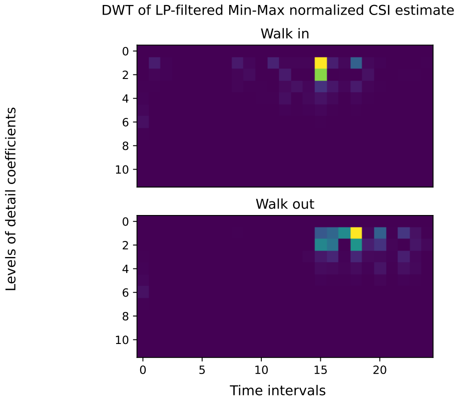

Figure: Shift-Time DWT of low-pass filtered CSI.

Detection of passage

Threshold trigger

- Passage detection was not used on samples in $\mathcal{D}$, but live detection mode.
- The following procedure was used to detect passage:

1. The signal is low-pass filtered using a Butterworth filter.
2. The variance of the low-pass filtered signal in 1 second windows is calculated.
3. If the variance is above a threshold a detection is triggered.

On detection, the system sends a 5 second window of CSI starting 1 second before the window that triggered the detection for classification.

Classification of passage

Models for classification

- 1DCNN: A convolutional neural network where the convolutional layers are 1 dimensional, to process 1D data
- 2DCNN: An ordinary convolutional neural network, to process 2D data.
- Other models such as a fully connected neural network, LSTM neural network, SVM and Hidden Markov Model were tested but not included in the final product.

Classification of passage

Feature pre-processing for 1DCNN and 2DCNN

1DCNN

1. Low-Pass filter
2. Min-Max normalize
3. Downsample

2DCNN

1. Low-Pass filter
2. Min-Max normalize
3. DWT
4. Z-Score normalize over all $R^{12 \times 25}$ features with empirical estimates saved from training set

Usage

Graphical user interface

In the GUI, a user can:

- Collect data for training[2]
- Start a live detection and classification of passages

[2] $\mathcal{D}$ was created using this feature.

Live mode

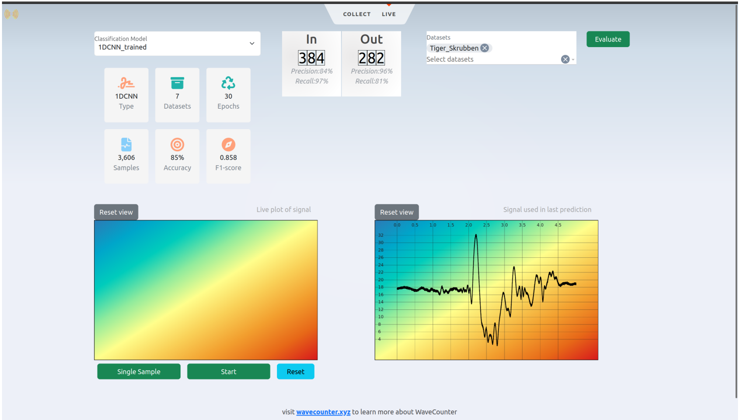

Figure: A view of the Live-page in the user interface.

Results

Data exploration

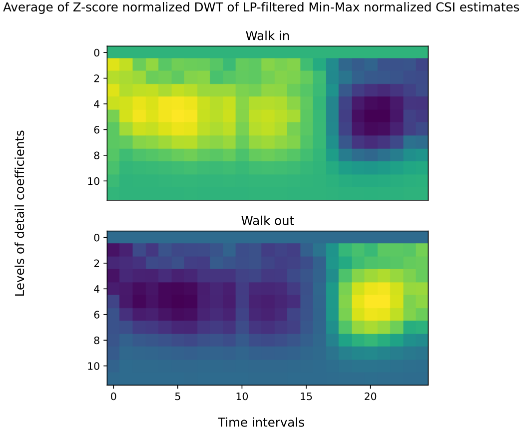

Figure: Average of all samples with DWT and Butterworth low-pass filtering.

Results

Data exploration

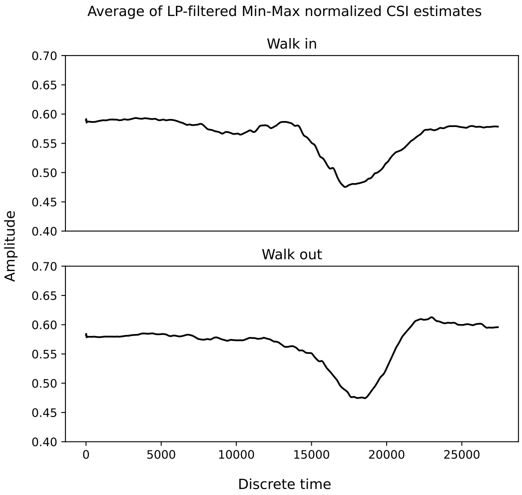

Figure: Average of all samples with Butterworth low-pass filtering.

Results

Model performance

- Models have been trained and validated on subsets of $\mathcal{D}$. While that is left out of this presentation it can be found in the Technical Documentation.
- Here we show the performance for the live mode:

Live mode performance on 50 passages:

- 1DCNN: 90% classification accuracy
- 2DCNN: 72% classification accuracy
- Passage detection: About 5% of passages are missed, these are not included in the 50 passages above.

Demonstration and live test

References

Channel state information. [Online]. Available: https://en.wikipedia.org/wiki/Channel_state_information