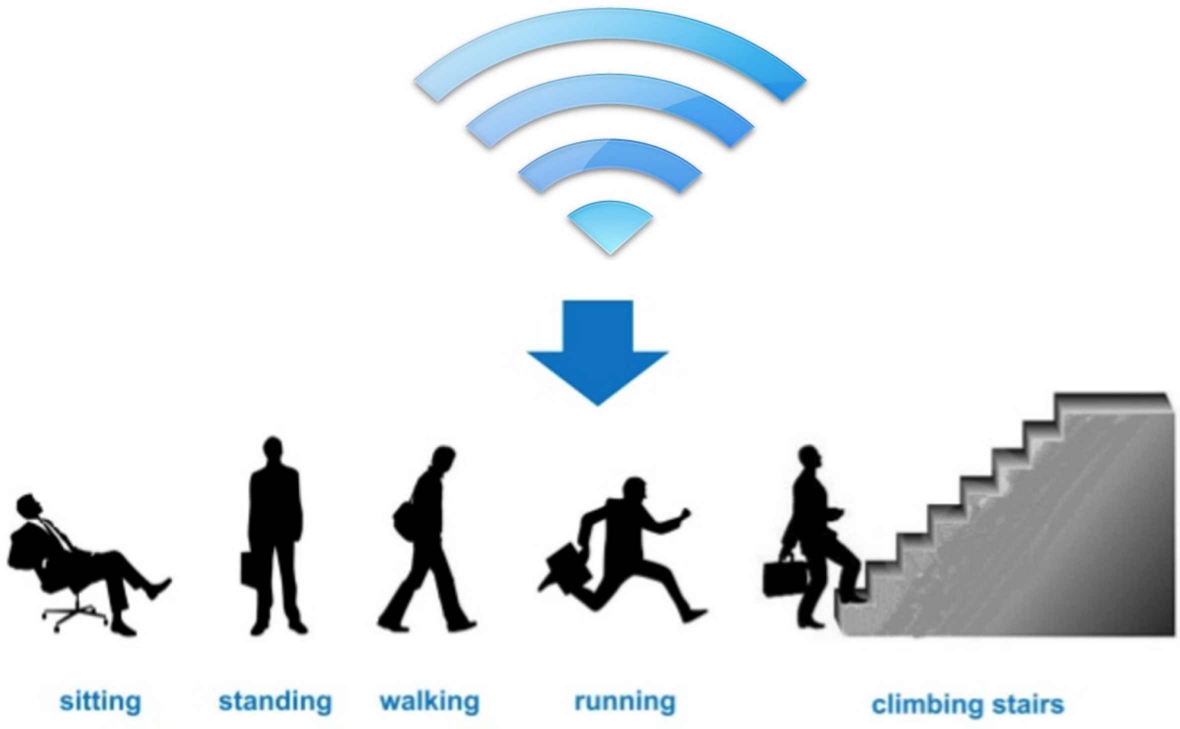
Activity Recognition using B-CSI data

Sanket Dige (110815356) Raunaq Kochar (110944843)



Introduction

This project deals with recognition of different activities performed by humans using Wifi (backscatter-CSI) data in lieu of the traditional methods of wearables, RSSIs or radars. The power of CSI data over plain RSSI values is that it provides us with both RSSI as well as Phase information for all subcarriers in the WiFi channel. “A human body can greatly attenuate WiFi signal and that can act as a "fingerprint" to identify the presence or absence of a person”, is the principle we will use for implementation of this project. Now instead of just using the CSI data onto which lot of research has been done and there has been a dead-end to the effect the technology can be leveraged for activity detection, here we are using back-scatter CSI data also known to be B-CSI data. A infrastructure named as RIBBN, developed at Stony Brook university, shows details about how exactly the backscatter principle works and you gather the data in a different format as compared to CSI data.

Application POV

From an application point of view, hospitals can use this technology to differentiate between a patient's accidental fall or voluntary lying down. Since this will use CSI data, cameras will not be necessary, thus reducing the effort for continuous video monitoring and also protecting the privacy of the patient.

Literature Survey

Papers Consulted

[1] Emotion Recognition Using Wireless Signals
[2] Keystroke Recognition Using WiFi Signals
[3] Understanding and Modeling of WiFi Signal Based Human Activity Recognition

a. Positive
    i. No external sensors needed to be carried by user/subject in question.
    ii. The noise filtering steps in each of the aforementioned papers is very detailed, sophisticated and specific to the filtering of CSI data.
    iii. Adequate techniques are introduced to analyze the accuracy of our training model with the given data.

b. Negative
    i. The environments in which the experiments have been conducted have not been thoroughly documented, hence making it difficult to reproduce these results.

Setup

We collected training samples for 10 different activities in our lab, which is 9m of length and 6m in width, as shown in figure 1. We collected total 2640 samples for the testing activities from 6 research participants. The participants included 5 males and 1 female graduate students with ages in the range of 25-34.
We evaluated the recognition accuracy of *B-CSI* through two sets of experiments, one is in the trained environments and the other is in the untrained environments. We use the lab where we collected the training environments.

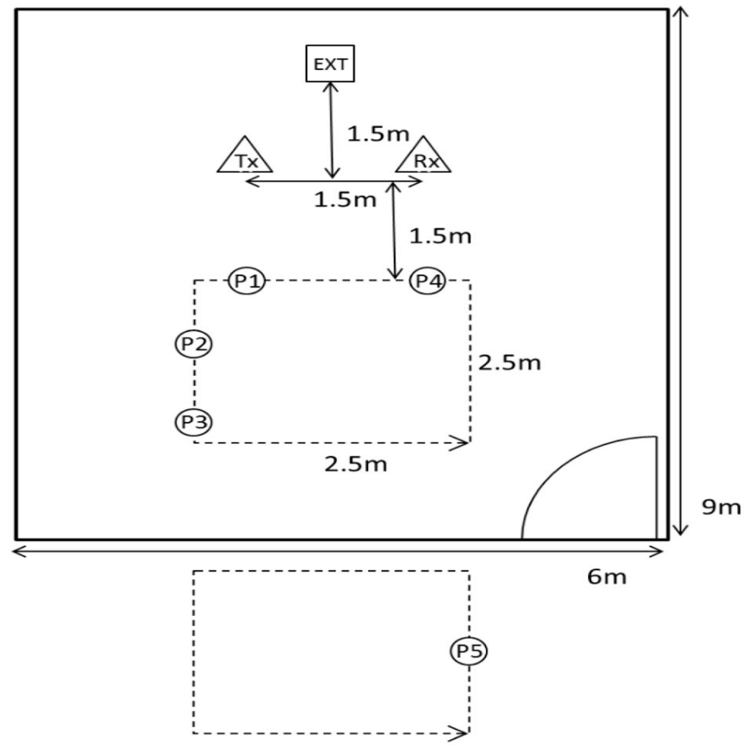


Figure 1: floor plan

The activities for which we collected training samples are listed in figure 2, along with abbreviations and number of samples for each activity. We collected samples for each activity. For walking on path and running on path, our participants followed the path with a dashed line at the center of the lab. Other activities were conducted at the five different locations which marked as circle. The location five is exceptionally conducted outside the lab. Tx and Rx represent the location of the tag node and sink node. EXT represent the location of exciter. The total time for our activity dataset with 2640 samples was data collection 1 hour 50 min on a desktop with Intel i5-3470 CPU, as shown in figure 2.

Activity	Samples	Dataset time
(FA) Falling	300	12m30s
(RP) Running on path	120	5m
(RS) Running on spot	300	12m30s
(SI) Sitting	300	12m30s
(SD) Sitting down	300	12m30s
(ST) Standing	300	12m30s
(TX) Texting	300	12m30s
(WP) Walking on path	120	5m
(WS) Walking on spot	300	12m30s
(WR) Writing	300	12m30s

Figure 2: Summary of activity dataset

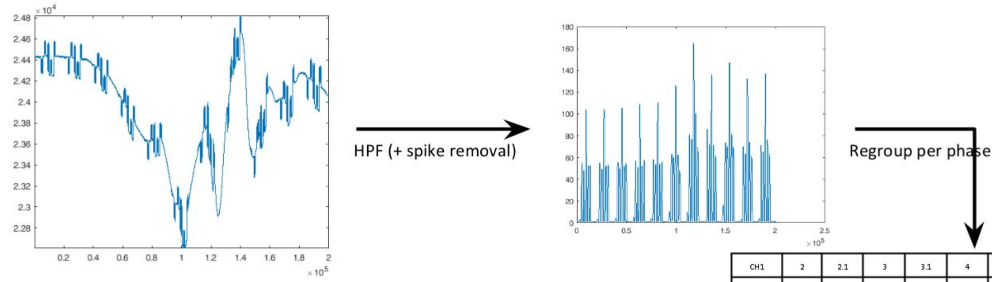
Experiment / Implementation

Our data set could be summarized basically as below -

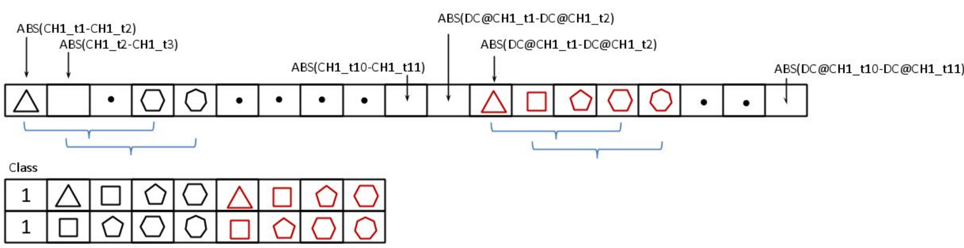
- 6 research participants
- 10 repetitions of activity on 10 activities
- 600+ collections of 2.5 sec each set
- 50%+ as training and 50%- as testing, even sets of collection divided into 50/50 for training and testing, more training for odd sets of the collection.

Our feature extraction could be summarized as shown in below figure -

Feature extraction



CH1	2	2.1	3	3.1	4	7	6.8	2	16	3	3.5
CH2	55	54	57	58	59	63	76	83	68	65	66
CH3	49	51	50	50.5	50.9	60	75.6	70	66.5	62.5	63
CH4-6	NOT counted										



First, with just the raw B-CSI data, we do a high-pass filter (+ spike removal) so as to get the 6 different channel data for 10/11 data-set in each sample file (sample activity recorded for 2.5 seconds). Now, for feature representation, we regroup per phase data as follows - CH1 [10/11], CH2 [10/11], CH3 [10/11], CH4 [10/11], CH5 [10/11] and CH6 [10/11]. We discard the use of data from CH4, CH5 and CH6 as they are a kind of combination of CH1, CH2 and CH3 itself.

We also use a low pass filter on the main raw CSI data and shrink it down to get a DC value of timeseries of 10/11 datasets i.e DC [10/11].

Thus, a single sampled activity data in B-CSI format will be represented as below -

DC1	DC2	DC3	DC4	DC5	DC6	DC7	DC8	DC9	DC10	DC11
CH1 1	CH1 2	CH1 3	CH1 4	CH1 5	CH1 6	CH1 7	CH1 8	CH1 9	CH1 10	CH1 11
CH2 1	CH2 2	CH2 3	CH2 4	CH2 5	CH2 6	CH2 7	CH2 8	CH2 9	CH2 10	CH2 11
CH3 1	CH3 2	CH3 3	CH3 4	CH3 5	CH3 6	CH3 7	CH3 8	CH3 9	CH3 10	CH3 11

Now for vector creation, we use a window-kind mechanism. We choose a window of 3, say the one colored in above figure, and create a feature vector of dimension 12.

DC1	DC2	DC3	CH1 1	CH1 2	CH1 3	CH2 1	CH2 2	CH2 3	CH3 1	CH3 2	CH3 3

Then, we move the window by 1 on right side and create a new feature vector -

DC2	DC3	DC4	CH1 2	CH1 3	CH1 4	CH2 2	CH2 3	CH2 4	CH3 2	CH3 3	CH3 4

This, is how we created the training data with its respective label of activity.

Once we created the feature representation, we choose different ML classifiers to classify the activities. We started by testing it with Logistic-Regression but it was giving very poor classification accuracy (less than 20%). The, we shifted to SVM.
The Support Vector Machine (SVM) [] with a linear kernel function for mapping input samples into high dimensional space was used. *B-CSI* achieved an validation accuracy of 98.1% across all activities.

B-CSI achieves an accuracy of higher than 98.2% for the research participant who is not included on training set.

Run the project

To predict live activity on incoming data, we have set up an infrastructure with the help of google drive and a command line utility named ‘gdrive’. We have created a client-server architecture where a client will push a sample_data that needs to be predicted for an activity by server.
On the google drive, we have a specific file ‘sample_data.csv’ which is specified by a given file_id once we create a shareable link from it.
Using this file_id, we override this sample_data.csv file with a new incoming live data sample using update_gdrive.py script. At client side, he should have the client folder which includes the gdrive_data folder and update_gdrive.py script. Inside the gdrive_data folder, we have a sample_data.csv file which needs to have a sample dataset for prediction. One we have that; client PC can push this file using update_gdrive.py script.
On server side, we have a live prediction system (predict_activity.py) which is constantly checking if any new data is uploaded on google-drive at that specific file_id (which is

sample_data.csv pushed to google-drive using the above scenario). If an updated version of file is seen, the prediction system, we download the sample_data.csv file, predict the activity from it and got to sleep for 10 seconds to check for any new data. If no updated file is seen, the prediction system will simply wait for 10 secs to check back again for any new data.
Now prior to this prediction, we should train our prediction system with some training set and stored the trained model. This is done by the 'SVM.py' script which takes in the sample data set from data folder and trains using SVM classifier. This trained model is then exported to 'activity_recognizer_model.pkl' file. This trained model file is then used by the prediction system to predict for any incoming sample test data-set.

Note : For live prediction, we only support prediction of first 6 activities namely -

1. Falling
2. Running on path
3. Running on the spot
4. Sitting
5. Sitting down on floor
6. Standing

Analysis and Evaluation:

To calculate the accuracy of our model, we have taken the following steps:

1. Divided the collected dataset as 50-50.
2. Trained the model on the training data and predicted results on the test data, appending the results as the last column.
3. To evaluate the model, we checked the majority of the number of predicted activities and the actual value of the activity. We considered the final predicted activity to be the value with the maximum frequency among all the values.

Consider the following diagram:

0	1	2	3	4	5	6	7	8	9	10	11	12	predicted_output
1	24.872	-53.399	95.591	-1.8518	-13.264	-12.84	1.2986	6.4897	6.8864	-13.079	-13.264	-12.84	1
1	-53.399	95.591	-101.28	-13.264	-12.84	-12.738	6.4897	6.8864	6.6295	-13.264	-12.84	-12.738	1
1	95.591	-101.28	-61.195	-12.84	-12.738	-15.283	6.8864	6.6295	-10.323	-12.84	-12.738	-15.283	1
1	-101.28	-61.195	935.95	-12.738	-15.283	10.277	6.6295	-10.323	-8.3498	-12.738	-15.283	10.277	1
1	-61.195	935.95	-212.94	-15.283	10.277	-33.499	-10.323	-8.3498	-12.417	-15.283	10.277	-33.499	1
1	935.95	-212.94	927.04	10.277	-33.499	-4.9806	-8.3498	-12.417	-13.623	10.277	-33.499	-4.9806	1
1	-212.94	927.04	-1899.2	-33.499	-4.9806	3.1629	-12.417	-13.623	1.921	-33.499	-4.9806	3.1629	1
1	927.04	-1899.2	1158.4	-4.9806	3.1629	-0.98631	-13.623	1.921	-1.4051	-4.9806	3.1629	-0.98631	1
1	-92.395	-31.307	-87.689	-15.664	-14.065	-7.1463	-9.1655	-0.14073	3.2421	7.9233	8.9705	9.1188	1
1	-31.307	-87.689	1051.5	-14.065	-7.1463	-28.727	-0.14073	3.2421	6.5482	8.9705	9.1188	-10.773	1
1	-87.689	1051.5	-373.21	-7.1463	-28.727	-16.108	3.2421	6.5482	-10.778	9.1188	-10.773	-20.184	1
1	1051.5	-373.21	-193.87	-28.727	-16.108	17.37	6.5482	-10.778	8.1483	-10.773	-20.184	-11.274	1
1	-373.21	-193.87	-1083.3	-16.108	17.37	-1.5609	-10.778	8.1483	-13.088	-20.184	-11.274	9.277	1
1	-193.87	-1083.3	1091	17.37	-1.5609	10.891	8.1483	-13.088	3.9516	-11.274	9.277	0.11962	1
1	-1083.3	1091	-617.92	-1.5609	10.891	1.8685	-13.088	3.9516	1.2023	9.277	0.11962	0.66076	1
1	1091	-617.92	-175.51	10.891	1.8685	-0.40317	3.9516	1.2023	3.6168	0.11962	0.66076	-0.044759	1
1	18.079	206.83	390.84	13.118	11.989	4.9008	0.20732	-0.54183	-1.542	-0.49231	2.8858	5.3049	1
1	206.83	390.84	91.689	11.989	4.9008	-13.725	-0.54183	-1.542	4.3287	2.8858	5.3049	4.5518	1
1	390.84	91.689	542.96	4.9008	-13.725	-25.907	-1.542	4.3287	3.9655	5.3049	4.5518	-1.0656	1
1	91.689	542.96	-179.24	-13.725	-25.907	-53.363	4.3287	3.9655	-3.3531	4.5518	-1.0656	-19.429	1
1	542.96	-179.24	60.773	-25.907	-53.363	23.35	3.9655	-3.3531	-2.6331	-1.0656	-19.429	-4.1044	1
1	-179.24	60.773	-1036.2	-53.363	23.35	-11.93	-3.3531	-2.6331	-2.462	-19.429	-4.1044	-7.0831	1
1	60.773	-1036.2	611.48	23.35	-11.93	15.384	-2.6331	-2.462	-5.2161	-4.1044	-7.0831	4.5544	1
1	-1036.2	611.48	-462.2	-11.93	15.384	-5.0179	-2.462	-5.2161	-8.3608	-7.0831	4.5544	-1.699	1
1	-67.967	-181.26	-298.92	-14.539	-13.156	-7.633	-6.95	0.026089	3.2213	7.9387	7.739	7.8121	10
1	-181.26	-298.92	-340.05	-13.156	-7.633	2.348	0.026089	3.2213	2.9297	7.739	7.8121	-3.8652	10
1	-298.92	-340.05	1647.4	-7.633	2.348	-39.942	3.2213	2.9297	-16.219	7.8121	-3.8652	-19.497	10
1	-340.05	1647.4	-1562.1	2.348	-39.942	4.1971	2.9297	-16.219	-11.189	-3.8652	-19.497	-11.193	10
1	1647.4	-1562.1	888.13	-39.942	4.1971	-15.851	-16.219	-11.189	1.5278	-19.497	-11.193	-15.189	10

1. If we look at the last column, we see that the number 1 has maximum frequency and we thus consider the predicted output to be 1.
2. From this analysis, we get an overall accuracy of 99%.

This is depicted as:

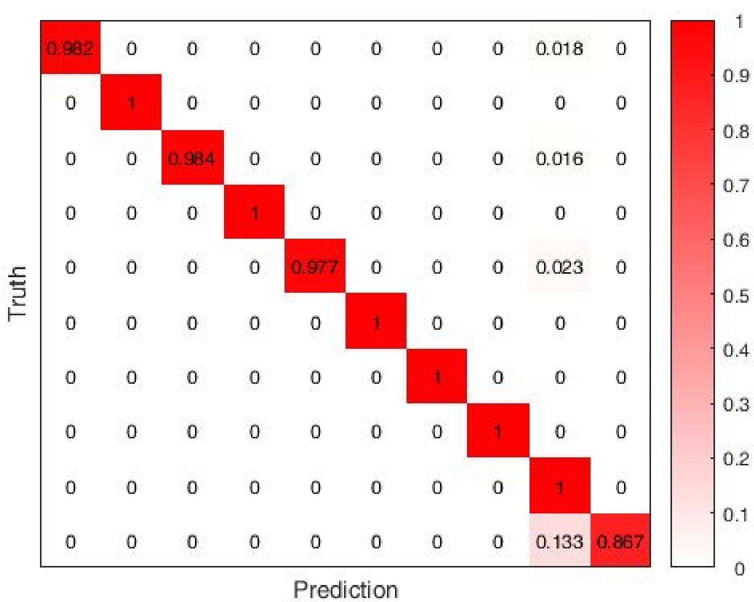


Figure: SVM Confusion Matrix for B-CSI

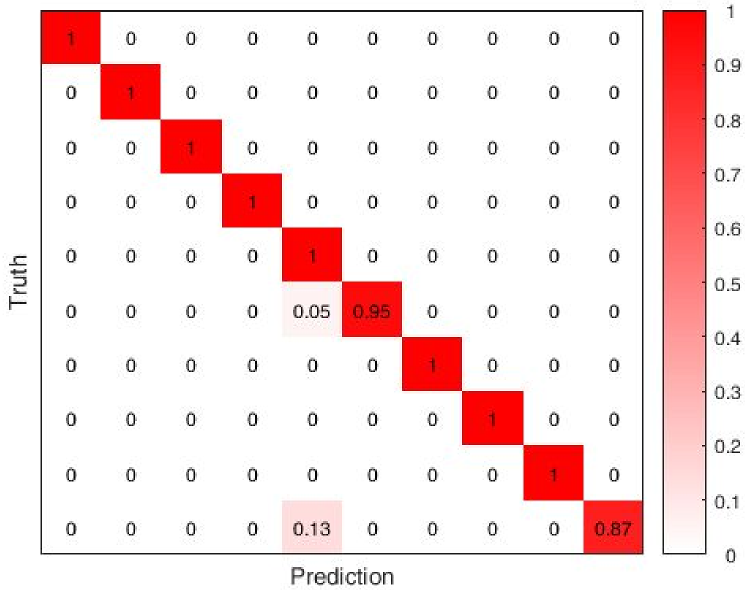


Figure: SVM Confusion Matrix for untrained participant

Link to a small video demonstration of activity detection using this model - [video link]

References

[1] Emotion Recognition Using Wireless Signals
[2] Keystroke Recognition Using WiFi Signals
[3] Understanding and Modeling of WiFi Signal Based Human Activity Recognition
[4] RIBBN - A Research Infrastructure for Backscatter-Based Networks