Title: WiPE-FaLl: Wi-Fi-based Prediction and Estimation of Fall Likelihood

This “ReadMe.pdf” file was generated on 2024-04-03 by Dr William Taylor

# **GENERAL INFORMATION**

## **Title of Dataset**

## WiPE-FaLl: Wi-Fi-based Prediction and Estimation of Fall Likelihood

## **Author Information**

Names: William Taylor, Scott Pinkerton, Muhammad Tufail Khan, Mohammad M.A. Taha,

Basel Barakat, Mahmoud A. Shawky, Qammer H. Abbasi, Muhammad Ali Imran, and

Ahmad Taha

Institution: University of Glasgow

**Address:** James Watt School of Engineering, College of Science and Engineering, University of Glasgow, Glasgow, G12 8QQ

**Emails:** [william.taylor@glasgow.ac.uk](mailto:william.taylor@glasgow.ac.uk), [2468314p@student.gla.ac.uk](mailto:2468314p@student.gla.ac.uk), [muhammadtufail.khan@uws.ac.uk](mailto:muhammadtufail.khan@uws.ac.uk), [mohammad.m.a.taha@gmail.com](mailto:mohammad.m.a.taha@gmail.com), [eng.basel\_barakat@live.com](mailto:eng.basel_barakat@live.com), [mahmoudshawky201539@gmail.com](mailto:mahmoudshawky201539@gmail.com), [qammer.abbasi@glasgow.ac.uk](mailto:qammer.abbasi@glasgow.ac.uk), [muhammad.imran@glasgow.ac.uk](mailto:muhammad.imran@glasgow.ac.uk), ahmad.taha@glasgow.ac.uk

## **Date of data collection:**

The data was collected between 21/07/2022 – 04/08/2022.

## The geographic location of data collection:

Room 511, James Watt South Building, Glasgow, G12 8QQ, United Kingdom

## Information about funding sources that supported the collection of the data:

This work is supported in parts by EPSRC EP/R513222/1, EP/T021020/1, and EP/T021063/1. This work was conducted during WT's studentship which was supported by CENSIS UK through the Scottish Funding Council in collaboration with British Telecom.. Mohammad M.A. Taha has completed this work as an independent researcher. He has since joined the Savannah River National Laboratory.

# **SHARING/ACCESS INFORMATION**

## Licenses/restrictions placed on the data:

NA

## Links to other publicly accessible locations of the data:

NA

## Was data derived from another source?

No

Recommended citation for this dataset: TBC

# **DATA & FILE OVERVIEW**

## **Details of Data Folders and Files**

The dataset is divided into 3 classes Low, Medium and High. There are 6 folders in total for training data and unseen data for each of the 3 classes (see Figure 1 and Table 1). There are 500 samples of each class with 475 used for training and 25 samples kept as unseen data. Each sample is in the form of a CSV file which contains the Channel State Information (CSI) amplitude data for each subcarrier in the OFDM communication link between two Universal Software-defined Radio Peripheral (USRP) devices.

A screenshot of a computer

Description automatically generated

*Figure 1 Data folder Structure*

*Table 1 Details of the Data Set (Folders, Files, Description, and Number of Samples)*

|  |  |  |  |
| --- | --- | --- | --- |
| Folder Name | Class/File Name | Description | Number of Samples per  Class |
| high | high | TUG test performed with a simulated high risk of falling with low balance. | 475 |
| high\_unseen | high | TUG test performed with a simulated high risk of falling with low balance. | 25 |
| low | low | TUG test performed with a simulated high risk of falling with low balance. | 475 |
| low\_unseen | low | TUG test performed with a simulated high risk of falling with low balance. | 25 |
| med | medium | TUG test performed with simulated medium risk of falling with low balance. | 475 |
| med\_unseen | medium | TUG test performed with simulated medium risk of falling with low balance. | 25 |

**METHODOLOGICAL INFORMATION**

## Description of methods used for collection/generation of data:

The dataset represents a combination of activities captured through wireless CSI, using two USRP X300 devices each equipped with the VERT2450 omnidirectional antennas. One USRP is used as the transmitter and the second USRP is used for the receiver. Each USRP was connected to an All-in-One PC that uses an Intel(R) Core (TM) i7-7700 3.60 GHz processor and 16 GB RAM. The system made use of virtual machines to provide the Ubuntu 16.04 operating system. On the Ubuntu virtual machines, Gnu Radio was used to communicate with the USRP devices. Figure 2 (a) and (b) shows the experimental setup which was used to collect the data.

|  |  |
| --- | --- |
| A diagram of a chair  Description automatically generated | (a) |

|  |  |
| --- | --- |
| A room with a diagram of a person  Description automatically generated | (b) |

*Figure 2 Experimental Setup*

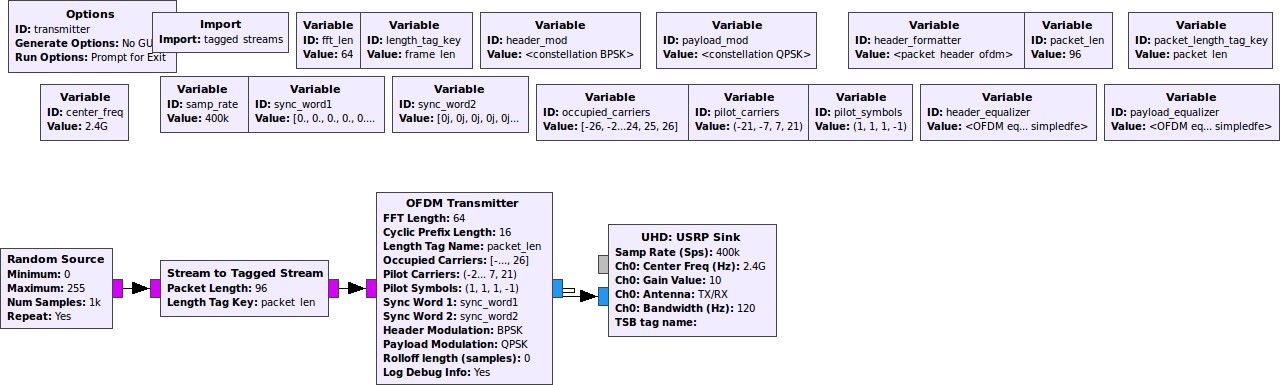
## Methods for processing the data:

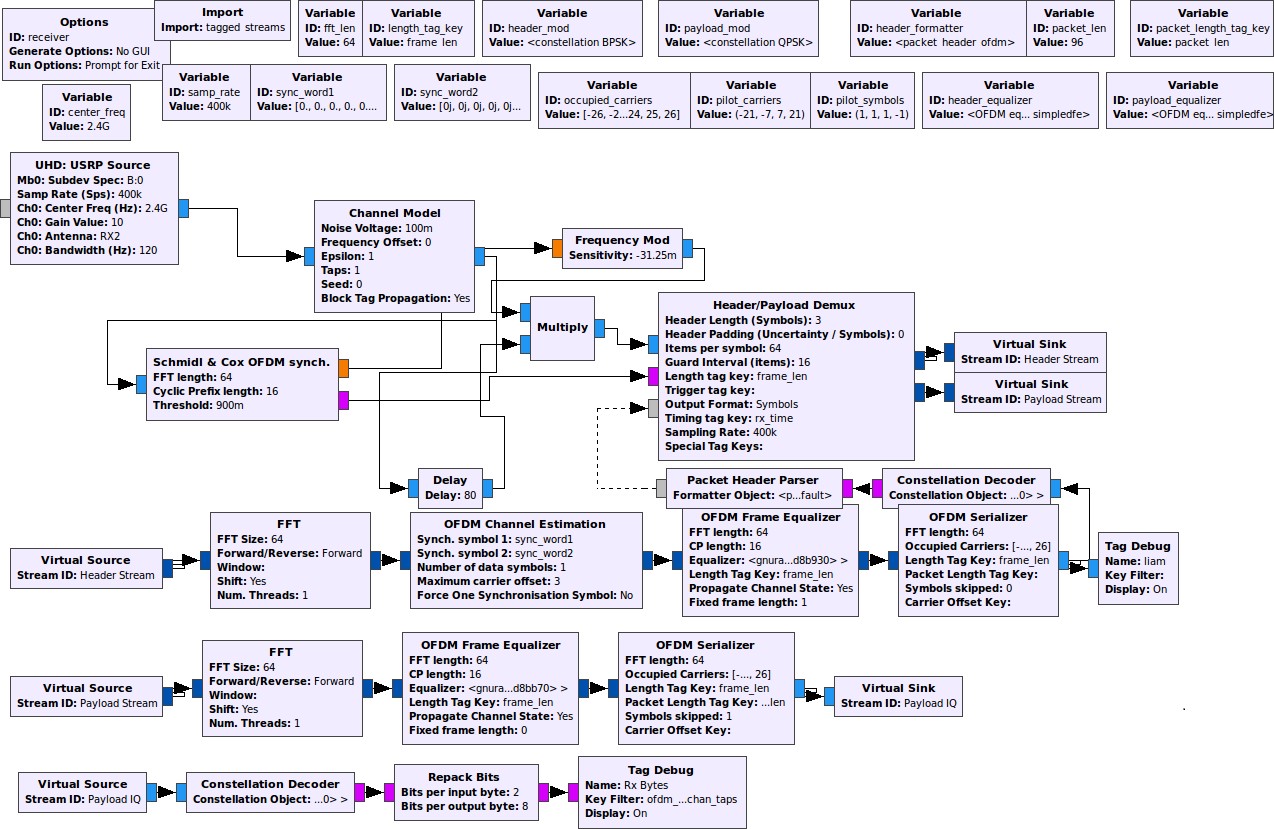
Firstly, the USRP transmitter and receiver devices were configured to communicate together using the GNU radio python package to set parameters such as central frequency, number of Orthogonal Frequency Division Multiplexing (OFDM) subcarriers, and power levels (see Table 2).

*Table 2 System Parameters*

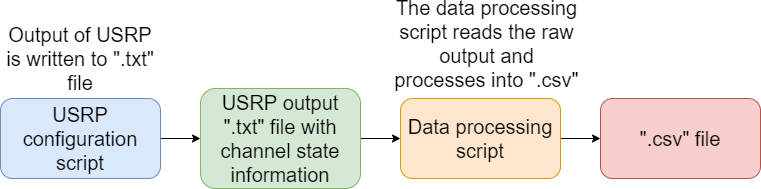
|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Central Frequency | 2.4 GHz |
| OFDM subcarriers | 51 |
| Transmitter Gain (dB) | 70 |
| Receiver Gain (dB) | 50 |

GNU Radio is a free and open-source software which is used in research for software-defined radios and signal processing. GNU Radio comes with examples of OFDM signal processing where the channel state information can be extracted. This example is modified to include the USRP as the transmitting and receiving devices. The GNU Radio software publishes the configuration in the format of a flow diagram which can be used to set up the blocks of the USRP and OFDM communication. Figure 3 shows the GNU Radio flow diagram that depicts the configurations of the USRP devices. The flow diagram can then be converted into a Python script, which can be executed to begin OFDM communication. The raw form of the channel state information output is written to text files which are then converted to a processable format (see Figure 4).





*Figure 3 GNU Radio Flow Diagram*



*Figure 4 Data flow in the data collection stage*

Secondly, the task was to collect the channel state information and create data sets from them in the form of “Comma Separated Value” (.csv) files. The CSV files would hold the data sets that will be used for training and testing of the Machine Learning (ML) algorithm. For this, another Python script is used to process the terminal output and filter out the channel state information complex numbers. Python

carries out mathematical functions to calculate the amplitude of the RF signal from the channel state information complex numbers. The amplitude values are then saved to CSV format for ML and to visualise the signal propagation through line graphs, see an example in Figure 5. The above process was repeated for all the data files in this data set.

A screenshot of a computer screen

Description automatically generated

*Figure 5 Channel state information capture of High Fall risk in ".csv" format and the*

*corresponding plot*

## Instrument- or software-specific information needed to interpret the data:

Data files are all in “.csv” format which could be opened using Microsoft Excel and further processed using Python.