Spectrum Vulnerability Meter – A Data Visualization Analytic Tool for Wireless Networks

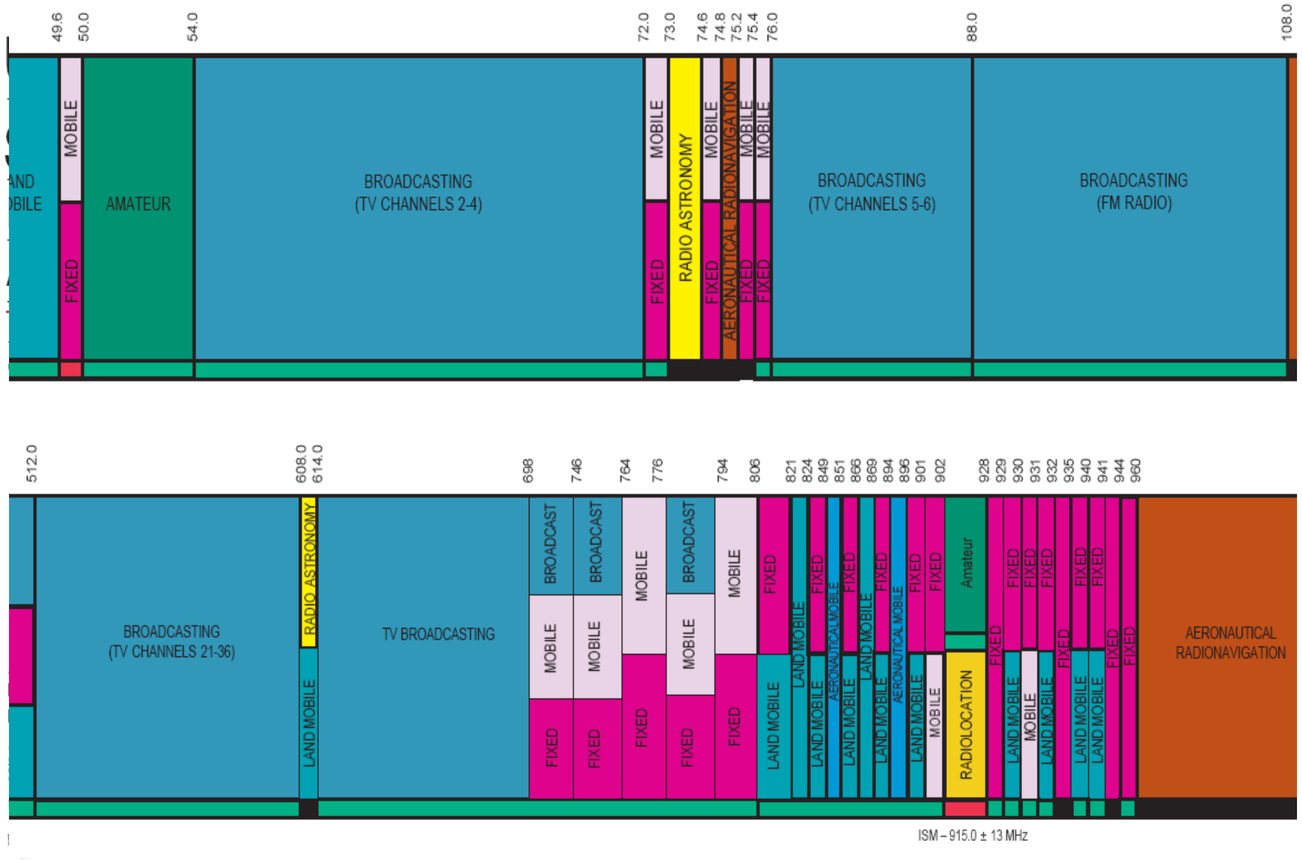
Presented By

Priyanka Samanta

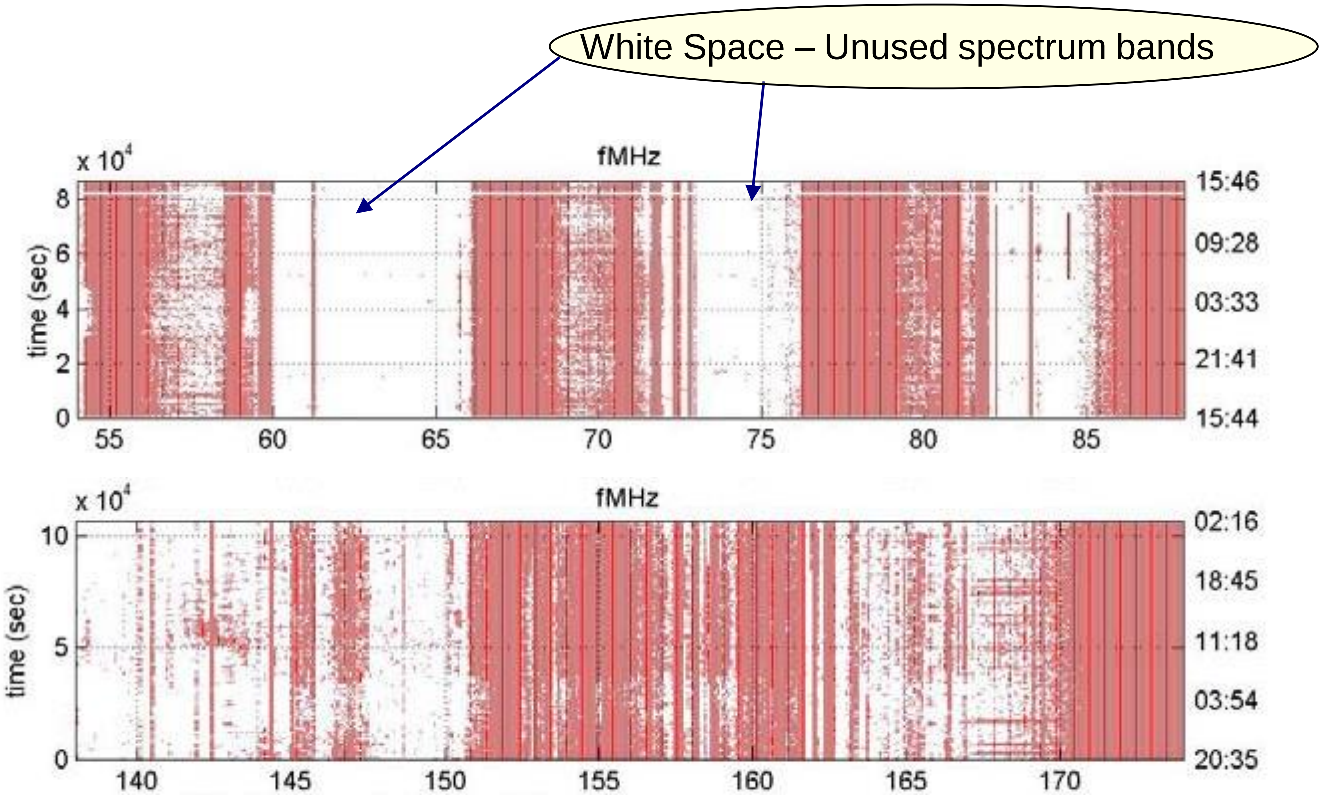
Ari Mermelstein

Department of Computer Science, The Graduate Center, CUNY

US spectrum access scenario



NYC occupancy summary



Motivation

- Next generation IoT application using 5G
- Dynamic Spectrum Allocation
- SSDF attack on DSA

Meta Data

DATASET NAMES

1. IN (Aachen, Germany, indoor in a modern office building, December 2006 – January 2007)
    - Subband 1: centre frequency: 770 MHz
    - Subband 2: centre frequency: 2250 MHz
    - Subband 3: centre frequency: 3750 MHz
    - Subband 4: centre frequency: 5250 MHz
2. NE (Maastricht, Netherlands, rooftop of a school in a residential area, May – June 2007)
    - Subband 1: centre frequency: 770 MHz
    - Subband 2: centre frequency: 2250 MHz
    - Subband 3: centre frequency: 3750 MHz
    - Subband 4: centre frequency: 5250 MHz
3. AB (Aachen, Germany, 3rd floor balcony in a residential area, June – July 2007)
    - Subband 1: centre frequency: 770 MHz
    - Subband 2: centre frequency: 2250 MHz
    - Subband 3: centre frequency: 3750 MHz
    - Subband 4: centre frequency: 5250 MHz

We concentrated on NE data set!!

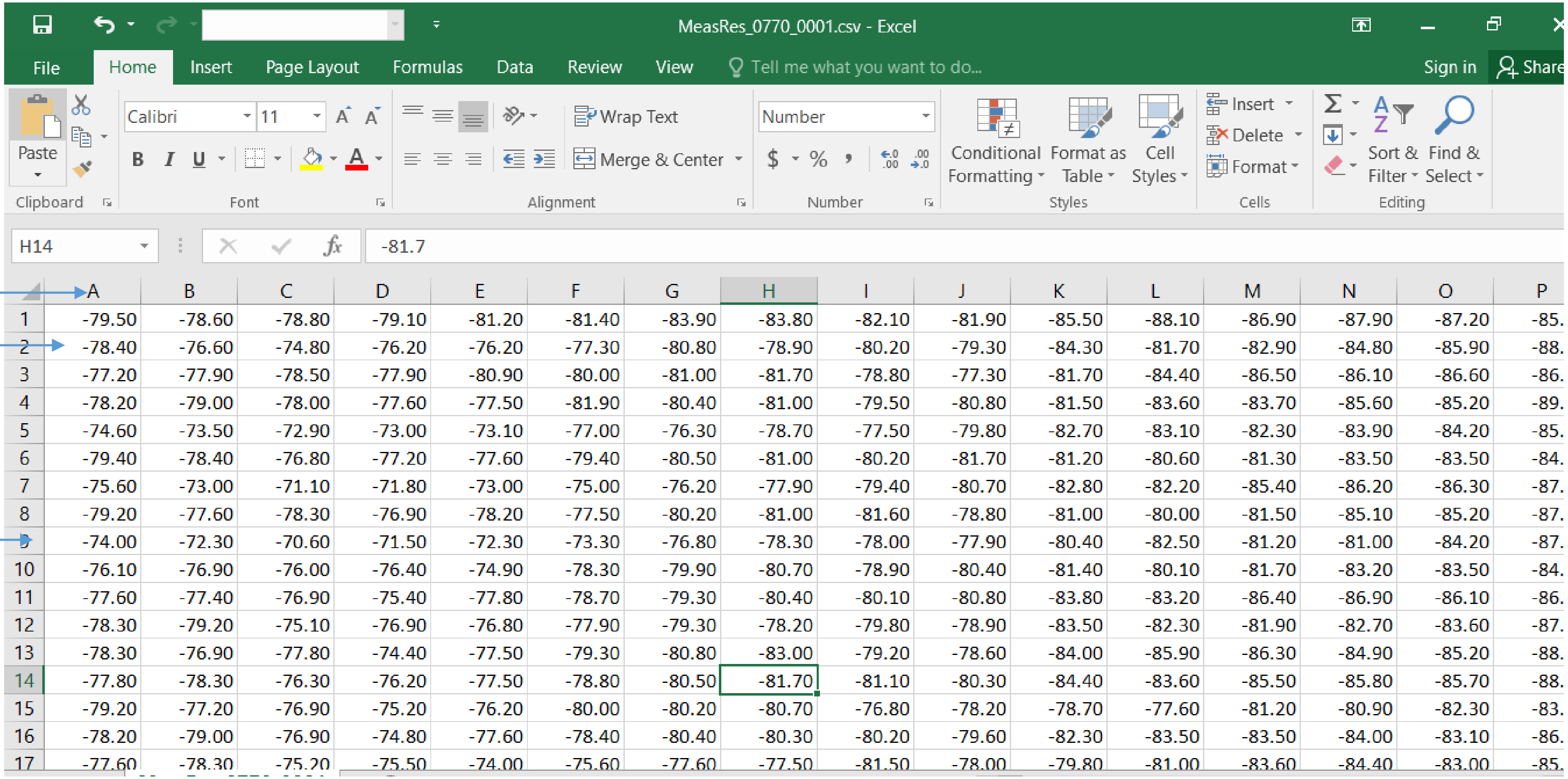
START TIME

1. IN
   - Subband 1: 2007-01-23 20:27:21 (Central European Winter Time = UTC+1)
   - Subband 2: 2006-12-27 16:26:01 (Central European Winter Time = UTC+1)
   - Subband 3: 2007-01-02 13:34:05 (Central European Winter Time = UTC+1)
   - Subband 4: 2007-01-08 12:33:13 (Central European Winter Time = UTC+1)
2. NE
   - Subband 1: 2007-05-15 11:00:59 (Central European Summer Time = UTC+2)
   - Subband 2: 2007-05-22 11:23:50 (Central European Summer Time = UTC+2)
   - Subband 3: 2007-05-29 11:22:51 (Central European Summer Time = UTC+2)
   - Subband 4: 2007-06-05 10:35:40 (Central European Summer Time = UTC+2)
3. AB
   - Subband 1: 2007-07-03 17:03:35 (Central European Summer Time = UTC+2)
   - Subband 2: 2007-07-25 09:55:44 (Central European Summer Time = UTC+2)
   - Subband 3: 2007-06-19 14:53:49 (Central European Summer Time = UTC+2)
   - Subband 4: 2007-06-26 13:56:46 (Central European Summer Time = UTC+2)

STOP TIME

1. IN
   - Subband 1: 2007-01-29 11:22:24 (Central European Winter Time = UTC+1)
   - Subband 2: 2007-01-02 13:21:56 (Central European Winter Time = UTC+1)
   - Subband 3: 2007-01-08 11:16:56 (Central European Winter Time = UTC+1)
   - Subband 4: 2007-01-15 11:25:36 (Central European Winter Time = UTC+1)
2. NE
   - Subband 1: 2007-05-22 11:14:40 (Central European Summer Time = UTC+2)
   - Subband 2: 2007-05-29 11:10:57 (Central European Summer Time = UTC+2)
   - Subband 3: 2007-06-05 10:35:05 (Central European Summer Time = UTC+2)
   - Subband 4: 2007-06-08 13:21:56 (Central European Summer Time = UTC+2)
3. AB
   - Subband 1: 2007-07-10 14:59:41 (Central European Summer Time = UTC+2)
   - Subband 2: 2007-08-08 13:21:47 (Central European Summer Time = UTC+2)
   - Subband 3: 2007-06-26 13:25:54 (Central European Summer Time = UTC+2)
   - Subband 4: 2007-07-03 16:50:47 (Central European Summer Time = UTC+2)

CSV files



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	-79.50	-78.60	-78.80	-79.10	-81.20	-81.40	-83.90	-83.80	-82.10	-81.90	-85.50	-88.10	-86.90	-87.90	-87.20	-85.
2	-78.40	-76.60	-74.80	-76.20	-76.20	-77.30	-80.80	-78.90	-80.20	-79.30	-84.30	-81.70	-82.90	-84.80	-85.90	-88.
3	-77.20	-77.90	-78.50	-77.90	-80.90	-80.00	-81.00	-81.70	-78.80	-77.30	-81.70	-84.40	-86.50	-86.10	-86.60	-86.
4	-78.20	-79.00	-78.00	-77.60	-77.50	-81.90	-80.40	-81.00	-79.50	-80.80	-81.50	-83.60	-83.70	-85.60	-85.20	-89.
5	-74.60	-73.50	-72.90	-73.00	-73.10	-77.00	-76.30	-78.70	-77.50	-79.80	-82.70	-83.10	-82.30	-83.90	-84.20	-85.
6	-79.40	-78.40	-76.80	-77.20	-77.60	-79.40	-80.50	-81.00	-80.20	-81.70	-81.20	-80.60	-81.30	-83.50	-83.50	-84.
7	-75.60	-73.00	-71.10	-71.80	-73.00	-75.00	-76.20	-77.90	-79.40	-80.70	-82.80	-82.20	-85.40	-86.20	-86.30	-87.
8	-79.20	-77.60	-78.30	-76.90	-78.20	-77.50	-80.20	-81.00	-81.60	-78.80	-81.00	-80.00	-81.50	-85.10	-85.20	-87.
9	-74.00	-72.30	-70.60	-71.50	-72.30	-73.30	-76.80	-78.30	-78.00	-77.90	-80.40	-82.50	-81.20	-81.00	-84.20	-87.
10	-76.10	-76.90	-76.00	-76.40	-74.90	-78.30	-79.90	-80.70	-78.90	-80.40	-81.40	-80.10	-81.70	-83.20	-83.50	-84.
11	-77.60	-77.40	-76.90	-75.40	-77.80	-78.70	-79.30	-80.40	-80.10	-80.80	-83.80	-83.20	-86.40	-86.90	-86.10	-86.
12	-78.30	-79.20	-75.10	-76.90	-76.80	-77.90	-79.30	-78.20	-79.80	-78.90	-83.50	-82.30	-81.90	-82.70	-83.60	-87.
13	-78.30	-76.90	-77.80	-74.40	-77.50	-79.30	-80.80	-83.00	-79.20	-78.60	-84.00	-85.90	-86.30	-84.90	-85.20	-88.
14	-77.80	-78.30	-76.30	-76.20	-77.50	-78.80	-80.50	-81.70	-81.10	-80.30	-84.40	-83.60	-85.50	-85.80	-85.70	-88.
15	-79.20	-77.20	-76.90	-75.20	-76.20	-80.00	-80.20	-80.70	-76.80	-78.20	-78.70	-77.60	-81.20	-80.90	-82.30	-83.
16	-78.20	-79.00	-76.90	-74.80	-77.60	-78.40	-80.40	-80.30	-80.20	-79.60	-82.30	-83.50	-83.50	-84.00	-83.10	-86.
17	-77.60	-78.30	-75.20	-75.50	-74.00	-75.60	-77.60	-77.50	-81.50	-78.00	-79.80	-81.00	-83.60	-84.40	-83.00	-85.

Channels
[BW: 200KHz]

PSD values
[dB]

Time Stamp
[1.8Sec interval]

Hardware setup

Figure 1: Measurement setup as deployed at the location in the Netherlands.

- Agilent E4440A spectrum analyzer with inbuilt pre-amplifier
- During the measurements a standard laptop configured the spectrum analyzer remotely using VISA-commands over TCP/IP over Ethernet.
- The running software is MATLAB.
- All software ran under Microsoft Windows XP operating system.
- Omnidirectional antennas:
    - A large discone antenna of type AOR DA-5000 covered the lowest frequency band between 20 MHz and 1.52 GHz.
    - A smaller discone antenna of type AOR DA-5000JA to receive signals in the next subband from 1.5 GHz up to 3 GHz.
    - Finally, a radom antenna of type Antennentechnik Bad Blankenburg AG KS 1-10 specified up to 10 GHz to cover the frequency range between 3 GHz and 6 GHz.

Explanation of Measurement Setup

TIME ZONE

- The time zone in which the measurements were taken is mostly Central European Summer Time or Central European Winter Time. The measurement data is based on these times and has not been adapted to UTC.

GEOGRAPHIC LOCATION

- We performed the first measurements in our laboratory in a modern office building in Aachen, Germany (IN: Latitude: 50° 47' 24.01" North, Longitude: 6° 3' 47.42" East).
- We selected a location in the Netherlands as second location. We placed our measurement box on the roof of the main building of the International School Maastricht (NE: Latitude: 50° 50' 32.34" North, Longitude: 5° 43' 14.93" East.), Maastricht, the Netherlands.
- We chose the third location as an example for locations where end-user devices capable of dynamic spectrum access might work in real scenarios. We measured the spectrum usage on a third-floor balcony of an older residential building in a rather central housing area of Aachen, Germany (AB: Latitude: 50° 46' 8.90" North, Longitude: 6° 4' 42.59" East).
- The view from the roof in the Netherlands was rather open and the location is in a residential area near downtown Maastricht.
- The balcony is open towards the area between Aachen downtown and Aachen main station with an angle of about 130°. The other directions are blocked by surrounding buildings.

NETWORK LOCATION

- Among others transmitters for cellular networks, broadcasting services, and short-range communication (WiFi, DECT, etc.) have been deployed near to both measurement locations.

CLOCK PARAMETERS

- The time stamps saved in the data files are taken from the internal clock of the laptop. Thus, their accuracy is not very good but clearly sufficient for our application since each sweep including the data transfer to the laptop took about 1.8 sec.

Code to Convert .mat to .csv

```
import scipy.io
import numpy as np
import sys

data = scipy.io.loadmat(sys.argv[1])
for i in data:
 if '__'not in i and 'readme' not in i:
  pos=sys.argv[1].find('.mat') #find the index of the . in the file name
  substr=sys.argv[1][:pos]     #substr = the name of the file without the .mat
  fileName=sys.argv[2]+substr+".csv" #fileName = the name of the file we want to create including the full path
  open(fileName,mode="wb").close()   #creates an empty file
  np.savetxt(fileName, data[i], delimiter=',') #saves all rows of .mat as .csv
```

Windows script

```
FOR %%G IN (*.mat) DO python converter.py %%G ..\..\..\..\Data\NE_5250\

PAUSE
```

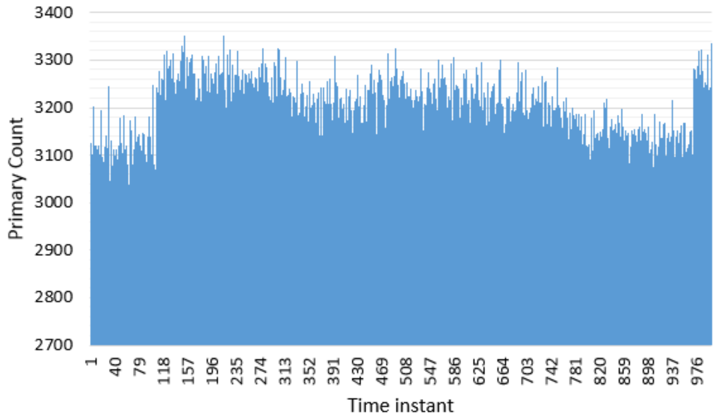
Preliminary codes

```
data = pd.read_csv("D:\\Data\\NE_5250\\MeasRes_5250_0146.csv")
number_list=[]
count =0
for index, row in data.iterrows():
    for cell in row:
        #print(cell, end=", ")
        if cell>(-90):

            count=count+1


    number_list.append(count)
    count=0
#print(number_list)
    #break
print(number_list)
```

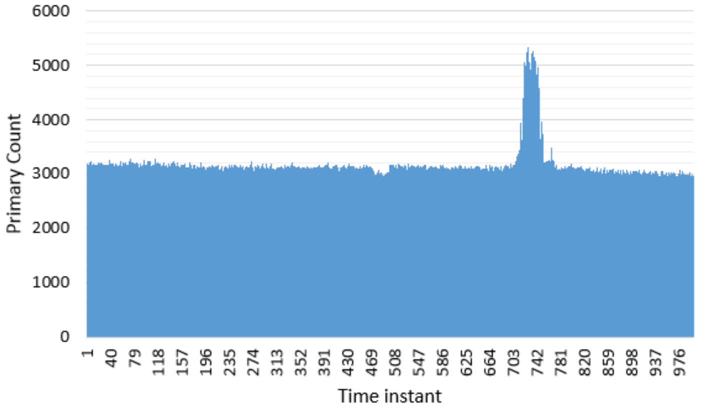
```
number_list_attack=[]
count =0
for index, row in data.iterrows():
    for cell in row:
        #print(cell, end=", ")
        x = np.random.rand()
        cell=cell+x
        if cell>(-90):

            count=count+1


    number_list_attack.append(count)
    count=0
print(number_list_attack)
```

5250 MHz

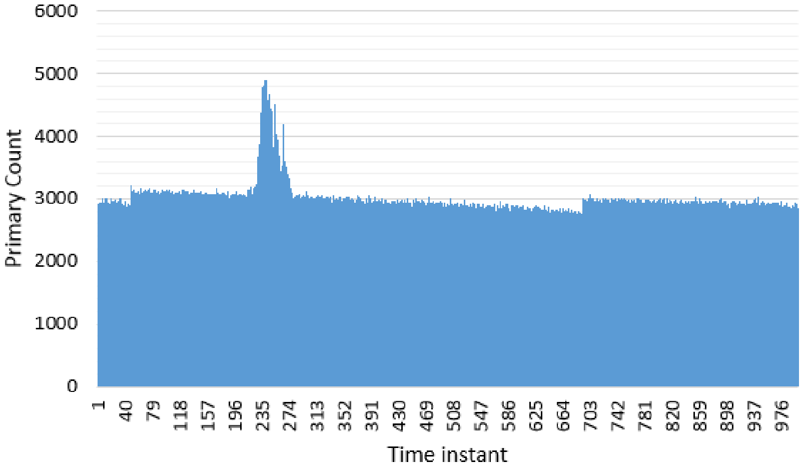
NE_5250\\MeasRes_5250_0001.csv



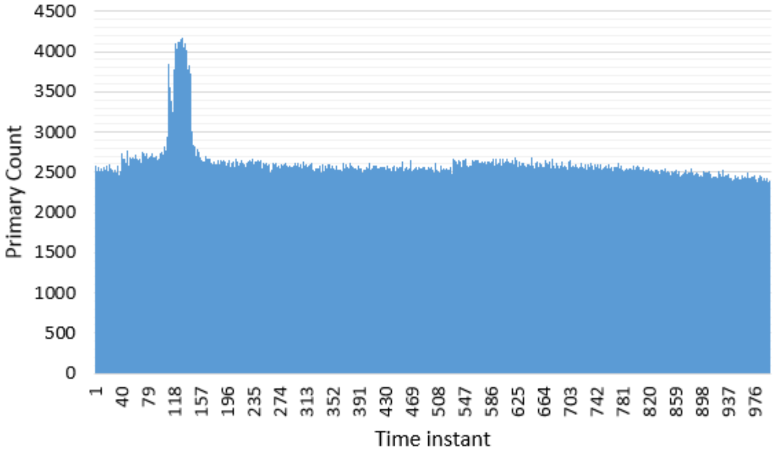
NE_5250\\MeasRes_5250_0052.csv



NE_3750\\MeasRes_3750_0100.csv



NE_3750\\MeasRes_3750_0146.csv



Objective Functions for visualization

Question 1: Is there any specific day when channels are most vulnerable?

Question 2: Is there any specific sub-band where channels are consistently vulnerable ?

Question 3: What is the probability of vulnerability for a particular channel under a particular sub-band given varying thresholds for vulnerability?

Question 4: What is the probability distribution of a particular channel of interest over the course of a week?

Coalescing all .csv's for each subband into one, big file

```python
import sys
import matplotlib.pyplot as plt
import pandas as pd
import csv

#if this big file already exists, open it for appending.  Otherwise, create it.
allData = open("NE770All.csv", "a+")
dataFile = open(sys.argv[1], "r") #open each individual small file for reading

#append each line of the small .csv to the large one
for line in dataFile:
    allData.write(line)

dataFile.close()
allData.close()

sys.exit(0)
```

```
FOR %%G IN (*.csv) DO python writeToOneFile.py %%G

PAUSE
```

DynamicViz Notebook

- Was a naïve attempt to read in and process each subband as one file.
- Problems:
  - The data structures were somewhat unwieldy and the syntax became very bulky.
  - Our computers don’t have enough memory to load such files (100+GB at a time!)
  - Using matplotlib didn’t give us enough flexibility and ease of use for our purposes.

plotlyWorkbookWithSizes Notebook

- We decided to break up each day’s worth of data into 4 parts so that we could handle the data at all.
- We were actually able to obtain suitable plots; however, they took a long time to run, and they didn’t give us that much power to visualize all of the data for an entire day at once.
- We found that plot.ly allowed us more flexibility to change the sizes and colors of the bubbles in the scatter plot, and the plots were dynamic by nature (they allow zooming, hovering to find a value, etc.)
- This gives the user a much easier interface by which to isolate individual channels to scrutinize further.

plotlyWorkbookWithSizesNumpy

- Created Numpy Arrays from the .csv files and compressed them into .npz files
- Could handle entire day
- No more partitioning required!!
- Used the (we thought) failed plot.ly code to generate plots much quicker.

NewHeatMap.py

- Creating Heat Map of a particular channel to show its probability distribution over a week

Dashboard

Question ???