RF Fingerprinting using the OneRadio MVP Platform

Alex Finestead and Jesse Yang
Advisors: John D.Sahr, Mohan Vaghul, Tony Goodson
University of Washington, Department of Electrical Engineering

Abstract

This project outlines and describes a method for identifying radio frequency signals based on a variety of waveform characteristics. A set of Python algorithms were developed to achieve this “fingerprinting”. Additionally, algorithms were written to cluster signals together, allowing for identification of distinct transmitter groups.

Background

OneRadio Corporation has produced a radio receiver with 2GHz bandwidth and a 90 dB dynamic range. Using data collected from this receiver, signals can be inspected and fingerprinted using the software algorithms developed in this project. Many RF devices will provide media access code (MAC address), however, this can be intentionally falsified to conceal the identity of the transmitter. This may be done to achieve many illicit actions, such as:

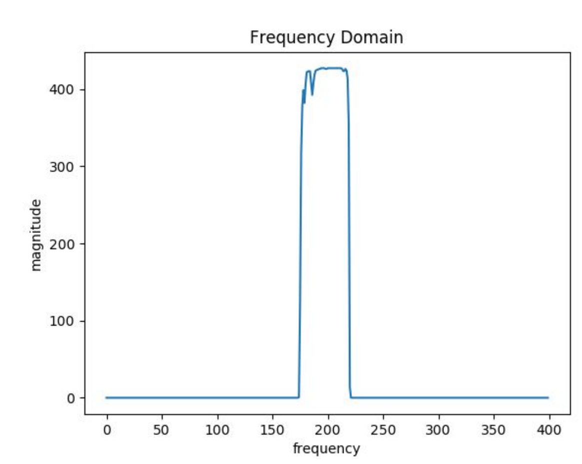
- Theft
- Denial of Service
- Transmitter Spoofing

This project attempts to create safeguards that make the aforementioned impossible.

Signal Characteristics

Transmitters can be very similar, however, certain signal characteristics vary between distinct transmitters. We classify these signals into groups by finding similarities on features such as:

- Power
- Frequency width
- Center frequency
- Energy
- RMS
- Rise time / Fall time




```
'RMS': 423.544219811501,
'centerfreq': 197.60312023345719,
'energy': 435734.76617729088,
'falltime': 4.9334259546956787,
'peak': 427.00010421679008,
'plot': 52,
'power': 423.47314261497121,
'risetime': 4.9979027359238728,
'width': 46.931328690619551}]
```

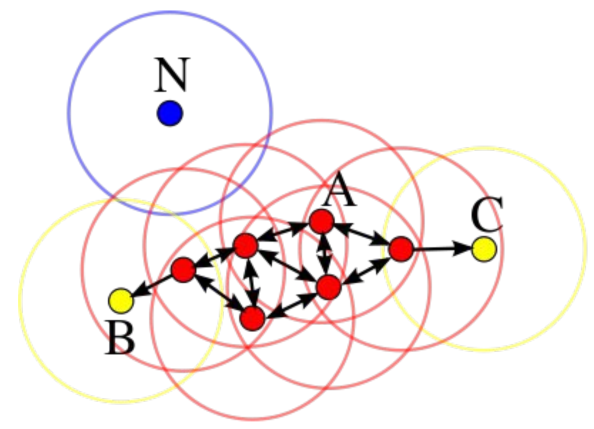
Data Collection

Data was collection was performed routinely throughout the project with the OneRadio platform. Data was typically taken in intervals of 10 seconds, with 4 tests performed for each collection type. Several tests were run to extract key transmitter features, such as toggling a radio transmitter on and off to inspect risetime and falltime characteristics.

Data was also taken at a variety of transmitter distances as a way of identifying waveform discrepencies that arise as distance increases.

Clustering Algorithms

A clustering algorithm, based on the DBSCAN algorithm (Density-Based Spatial Clustering of Applications with Noise) was written as a way of algorithmically and automatically extracting transmitter clusters.

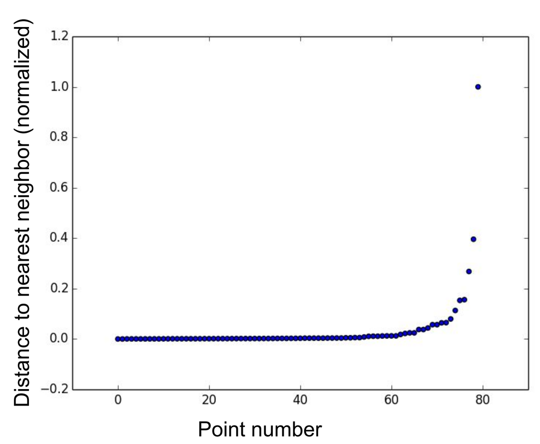


DBSCAN Visual

The diagram to the left presents a visual of the algorithms mechanics. Red points are considered “core” points, yellow points are “border” points, and blue points are “noise” points.

The DBSCAN algorithm clusters by the density of points in an area. All points closer than a given radius *r* to point *P* are merged into a cluster. If no points are closer then *r*, the point is considered noise.

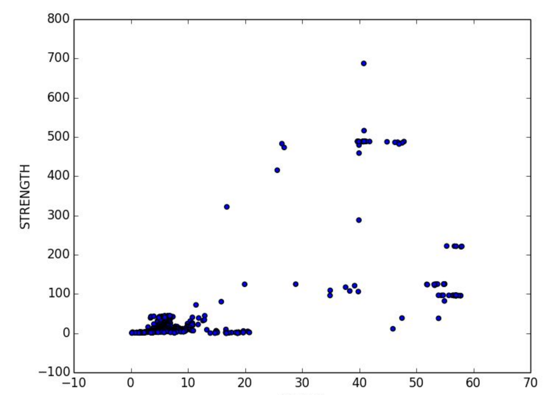
An optimal value for this radius is calculated by finding the minimum distance to another point for every point in the dataset (see right). The “elbow” of this plot is then calculated and it’s value used for radius *r*.



Selecting Optimal Radius

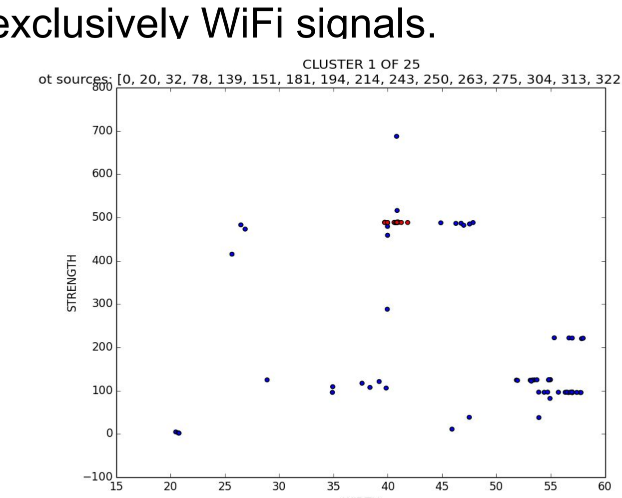
Our clustering algorithms were extended to be N-dimensional, meaning that transmitter groups can be clustered by any number of characteristics.

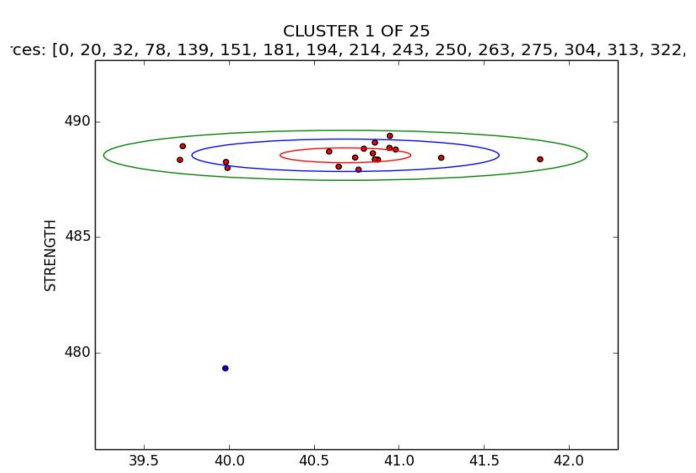
Results



The plot to the left demonstrates a dataset which has been plotted with signal width on the X-axis, and signal strength on the Y-axis. The large collection of data points in the bottom left corner is grouping of Bluetooth signals; the rest are WiFi.

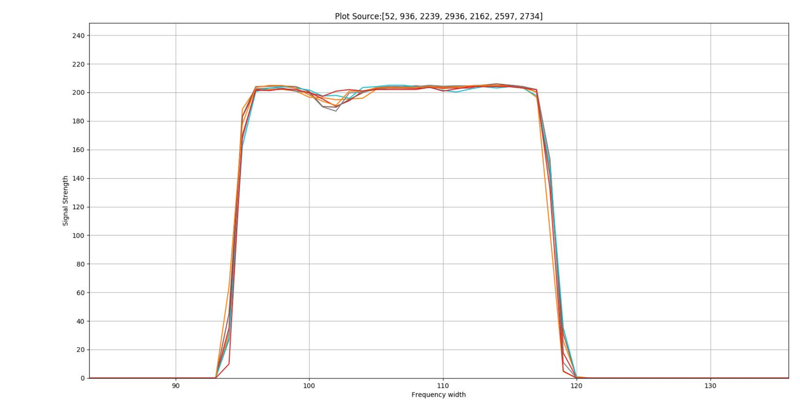
The plots below demonstrate the results of a clustering analysis on a set of exclusively WiFi signals.

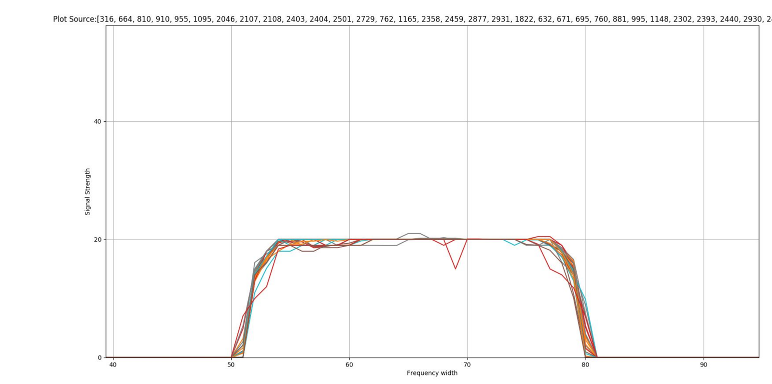




Data points highlighted in red are determined to be in a single transmitter group. The ellipses on the right plot indicate one, two, and three standard deviations away from the center of the cluster, essentially giving a measure of cluster certainty.

Results (cont.)





The plots to the left show the waveforms of all data points in two selected clusters overlaid.

Plots like these show the power of OneRadio together with the algorithms developed for this project; Transmitter groups can be accurately selected from a raw data set, relying on nothing but waveform characteristics.

This is proof that:

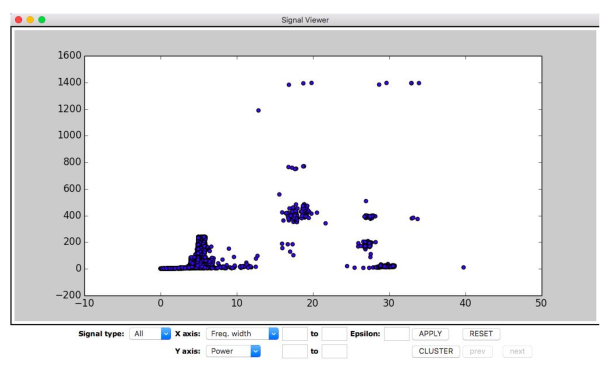
- Signals from transmitters can be grouped based on the metrics developed
- Our clustering algorithms can be used to reliably pull out groups of transmitters

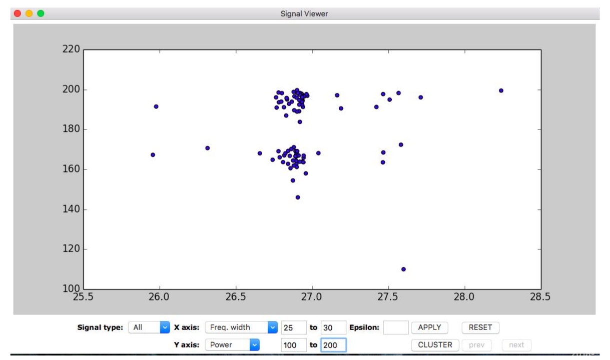
User Interface

A graphical user interface (GUI) was also developed in conjunction with the analysis and clustering algorithms as a way of visualizing the results.

This application gives the user the ability to intuitively plot and cluster data along selected axes, and select ranges, data types, and plots to display.

The application screenshot to the right shows the interface working in conjunction with the clustering algorithms.





Conclusions and Future Work

The program can address all identified features from the bullet list in the signal characteristics section at left, and has the ability to classify signal clusters. The results demonstrate the effectiveness of the technology developed.
Future work for this project includes:

- Identifying more features for reliability.
- Developing a more sensitive algorithm to classify Bluetooth signals
- Adding curve fit models as a way of extracting rise and fall times
- Converting the algorithms into GPU code for real-time implementation
- Adding features to the user interface to increase functionality.

On the whole, the project was a success. Transmitter groups can be identified algorithmically using nothing but waveform characteristics. While physical devices cannot yet be identified, the software developed here lays a solid framework for doing so in future projects.