Machine Learning Empowered Radio Frequency Signal Classification for UAS Detection

Michael Nilsen[1] / Sachin Shetty[1] / Kimberly Gold[2] / Charles Kamhoua[3]
1Virginia Modeling, Analysis and Simulation Center, Suffolk, VA
2Naval Surface Warfare Center, Crane, IN
3Army Research Lab, Adelphi, MD
UNITED STATES OF AMERICA

mnilsen@odu.edu / sshetty@odu.edu

ABSTRACT

Rapid developments in the unmanned aerial systems (UAS) have made its usage in a variety of offensive as well as defensive applications especially in military, high priority and sensitive government sites. The ability to accurately classify over-the-air radio signals will provide insights into spectrum utilization, device fingerprinting and protocol identification. These insights can aid in estimating the UAS transmitters capabilities without their knowledge. In this paper, we present a Radio Frequency Signal Classification (RF-Class) toolbox that can monitor, detect, and classify wireless signals emitted by UAS. The advantage of the RF-Class toolbox is extracting information about transmitters and providing receivers information about certain transmitted signals. The classification of RF signals will be done based on the modulation scheme recognition, exploitation of cyclostationary features and leveraging RF band allocation information. The modulation recognition capability can also be used for cyber offensive strategies. Once the modulation scheme is recognized, we can demodulate, decode and extract packets. Once the packets are extracted, we can accurately detect the protocol. The final step involves crafting a malicious packet and injecting the packet in the adversarial communication environment with intent to launch offensive operations. To demonstrate the feasibility and accuracy of our approach, we have evaluated the performance on a real environment with an UAS (Drone – DJI Phantom 4). Our initial experimental result showed that we were able to detect presence of drone signal successfully in presence of varying SNR regimes.

1.0 INTRODUCTION

The influx of UAS devices has increased the threat surface in both commercial and military domains. Current security solutions are facing a huge challenge in detection of the rogue UAS devices. These solutions leverage fingerprints that are dependent on characteristics unique to specific UAS and cannot detect new UAS that are being introduced in the market on a regular basis. Attackers can easily substitute UAS devices and successfully launch attacks. In order for the current solutions to be effective, there is a need for a library of fingerprints of all new UAS devices, which is an expensive task and cannot be accomplished in a reasonable amount of time.

There are several approaches to accurately fingerprint UAS devices and create a library of authorized and unauthorized devices. There are several approaches that leverage the physical properties of the devices to detect and geo-locate the presence of rogue UAS devices. The challenge with this approach is that they are prone to false positives due to potential errors in accurately collecting the physical characteristics. Instead, all UAS devices use standard communication protocols to transmit messages. However, the UAS devices modify the communication parameters to meet quality of service and security requirements. By fingerprinting UAS devices based on communication parameters, we can achieve higher degree of accuracy in detecting rogue devices and scale well as the approach is not dependent on any specific vendor or model. Additionally, as the appearance for many devices are able to be changed rather quickly, the communications protocols and standards are one of the more concrete aspects that allows for a high degree of accuracy in specific device classification.

In this paper, we present our ongoing work on development of a RF-Class Toolbox to aid in detecting and classifying UAS signals, which are captured in various SNR regimes to create a library of known UAS signals. For every detected signal, RF-Class toolbox will also identify the modulation parameters and the decoded packet structure. This information can be used to craft a waveform and launch a protocol attack on the adversarial UAS signal. The RF-Class toolbox's ability to detect signal energy, automatic modulation recognition and classification will be useful for counter UAS efforts.

The RF-Class toolbox depends on automatic radio frequency (RF) modulation classification which is an intermediate step between spectrum (i.e., signal) detection and demodulation. Implementation of advanced information services and systems such as signal detection and classification are challenging tasks, especially in a non-cooperative crowded environment where various factors like interference, low SNR, fading, phase and frequency offsets cause distortion to the received signal. Signal detection and classification are predominated processes in several civilian and military applications such as dynamic spectrum access (DSA), authentication, threat detection, etc. In a hostile non-cooperative environment, it is important to detect a primary user signal and demodulate it securely, while simultaneously detecting, jamming and/or modify the intruder signal.

Software Defined Radios (SDRs) are widely used to develop various civilian and military applications. The SDRs are open source and this makes them ideal for implementing a Radio Frequency Signal Classification (RF-Class) toolbox. SDRs usually works with variety of communication systems and often reconfigured through transmitted supplementary information. Modern signal transmission systems are intelligent, yielding an increase in the transmission efficiency by reducing the overhead. Such applications have emerged the need for flexible intelligent receiver, where the automatic recognition of the modulation of a detected signal is a major challenging task. Automatic modulation classification (AMC) is an important element that helps to detect and classify the known and/or unknown signal. Due to the non-cooperative nature and real-world scenarios, AMC faces various challenges due to unknown carrier frequency, phase offsets, time offsets, signal power, multipath fading, frequency selective fading, etc.

Numerous AMC techniques have been studied for more than a decade that concludes AMC design involves two phases: Signal Detection and Classification. The signal detection phase performs estimation of signal-to-noise ratio (SNR), signal power, etc. Whereas classification categorizes signals based on their modulation types and finds sub-modulation types. Due to the usage of various types of digital modulation techniques, it is difficult to select a particular signal detection method and classification technique that is applicable in all scenarios. So, it is needed to design and implement a flexible intelligent Radio Frequency Signal Classification (RF-Class) toolbox.

The classification techniques available in RF toolbox are categorized into two different categories of classifiers such as feature-based (FB) and likelihood-based (LB) classifiers. The latter is based on the likelihood function of the received signal and the decision is made comparing the likelihood ratio against a threshold. A solution offered by the LB algorithms is optimal in the Bayesian sense, i.e., it minimizes the probability of false classification. The optimal solution suffers from computational complexity, which in many cases of interest naturally gives rise to suboptimal classifiers. In the FB approach, on the other hand, several features are usually employed, and a decision is made based on their observed values. These features are normally chosen in an ad-hoc way. Although a FB-based method may not be optimal, it is usually simple to implement, with near-optimal performance, when designed properly.

In recent years, new technologies for wireless communications have emerged. The wireless industry has shown great interest in orthogonal frequency division multiplexing (OFDM) systems, due to the efficiency of OFDM schemes to transmit information in frequency selective fading channels, without complex equalizers. Multiple-input multiple-output (MIMO) systems have also received considerable attention, due to the significant capacity increase they offer. Such emerging technologies in wireless communications have raised new challenges for the designers of signal intelligence and SDR systems, such as, discriminating between

OFDM and single carrier modulations, identification of signals transmitted from multiple antenna systems, and so on.

2.0 APPROACH

We have designed and implemented a RF-Class toolbox on an SDR to classify presence of a UAS device within a specific frequency band. In the paper, we plan to propose the detailed design of the RF-Class toolbox, the implementation of the toolbox in an SDR and our results conducted on detecting DJI family of drones. Below is a brief description of the key elements of the RF-Class toolbox:

- RF Signal Detection – RF-Class's signal detection block will automatically extract signals by analysing broadband spectrums. We will employ Energy Detection (ED), Feature Detection (FD) and automatic Environment Thresholding (ET) to ensure detection and robustness to noise and fading effects. In some scenarios, we might also need to split the spectrum into sub channels and conduct energy detection within the sub channels.
- Cyclostationary Features Extraction – Modulated signals are typically cyclostationary processes. Cyclostationary processes have periodic autocorrelation functions and these functions can be extracted using Fourier analysis. This analysis can also give indication of modulation type. We will use cyclostationary-based signal processing algorithm such as spectral correlation function (SCF) and spectral coherence function (SOF) to fully observe the cyclic frequency features.
- Automatic Modulation Recognition – RF-Class's modulation recognition block will classify based on extracted cyclostationary features. We will have the ability implement recognition of PSK, FSK, QAM, OFDM schemes.
- Machine Learning Models – RF Class's extracted data is used to train machine learning models that are able to predict various classes of detected signals. This data is used as part of the feature-based (FB) classifiers and allows complex environment situations to be computed from mathematical models.
- Visualization – RF-Class's Visualization block will provide user friendly interfaces for users to analyse receiving signals. The block will visualize the receiving wideband spectrum, detected signal, modulation scheme and signal classification.

2.1 RF Signal Detection

RF Class's signal detection focuses on physical layer signal characteristics. Due to the nature of adversarial UAS configurations, this approach allows for raw signals to be extracted and classified. This technique extracts signals from wideband spectrums, allowing for entire bands to be monitored in real-time. The ED technique computes a signals power over a given period of time. This calculation converts a signal from the time to frequency domain and squares the magnitude of the signal, effectively computing the average power of the received signal. This technique aims at providing the system with a capability to filter out signals that don't meet specifications of UAS.

Of equal important to the signal detection approach is the ability to threshold the system in decreased SNR environments or in situations where signal levels fluctuate frequently. The ET technique uses baseline measurements to update environment thresholds that are used to compute various machine learning models. This technique identifies the lowest power calculation of the signal by searching and comparing the bins of the conversion to the frequency domain. In allowing the various environment baselines to be added to the models, models can be used across different environments, given the parameters of the UAS are homogenous in both situations.

SDRs have been proven one of the best pieces of hardware to run the RF Class system. Due to the nature of

SDRs that provide wide continuous RF coverage and operations of wide real-time bandwidths, the ability to split RF bands into multiple sub channels allows for entire bands to be monitored in real-time. This split requires multiple devices capable of operating at their respective device limits, allowing a network of devices to integrate workflows together into a unified system. This technique can be reconfigured to share resources across the network, allowing processing chunks to run instances of the RF class. These instances are then responsible for computing both ED and ET techniques for the channels being monitored.

2.2 Cyclostationary Feature Extraction

Cyclostationary feature extraction is a technique used to extract useful signal characteristics using FB classifiers. The signal is passed from the energy detection block in the flowgraph to a block used to detect unique parameters of the modulated signal. These features are extracted by using various spectral correlation functions and blind measurement functions based on predetermined and/or precomputed data inputs. This feature extraction focuses on parameters unique to Orthogonal Frequency Division Multiplexing (OFDM) modulated signals. This extraction is based on the predetermination of the UAS using a OFDM modulated signal. The four parameters from this modulated signal RF-Class was created to extract are shown in Table 1.

Table 1. Extracted Cyclostationary Features

Parameter	Variable	Units
Cyclic Prefix Length	CP	μs
Subcarriers	N	none
Subcarrier Spacing	Δf	kHz
Symbol Time	T_s	μs

The process of the cyclostationary feature extraction involves four steps including correlation functions and estimation of features based on assumptions that are then refined based on real data. The first step in extracting features starts with the discrete signal y[n]. In this case, the signal is a vector of ADC measurements performed by the software defined radio. Using this signal, we calculate the autocorrection that is used to estimate the FFT length or subcarriers. This technique is evaluating a blind approach to determining the first OFDM signal parameters, under assumptions of a random signal lag l. Equation 1 below shows the discrete autocorrelation function used to determine the number of subcarriers of the OFDM signal.

$$R_{yy}(l) = \sum_{n \in Z} y(n) * y'(n-l) \quad (1)$$

The number of subcarriers is then found by calculating the length of R. This calculation is the first step in determining the presence of a UAS signal, usually resulting from a higher number of subcarriers due to the large amount of information being transmitted.

The next step involves calculating the Cyclic Prefix length (CP) from the total length of subcarriers. However, this calculation involves a Cyclic Autocorrelation Function (CAF), a time-varying autocorrelation of a complex-valued cyclostationary signal x(t). For cyclostationary signals, the autocorrelation depends on a

central time t. This time t is centered between time instants $t_1$and t_2. Equations 2 and 3 show how times $t_1$and t_2 are calculated to be used as limits in the autocorrelation, Equation 4. In this autocorrelation, the result is a periodic function, which can be represented as a Fourier series, with $R_x^{\alpha}(\tau)$, the coefficient which is the CAF of the signal. This calculation takes a complex signal and allows for a fast Fourier transform to be implemented. Equation 5 represents the mentioned Fourier series expansion of Equation 4.

$$t_1 = t + \frac{\tau}{2} \tag{2}$$

$$t_2 = t - \frac{\tau}{2} \tag{3}$$

$$R_x(t,\tau) = E[x\left(t+\frac{\tau}{2}\right) * x^*\left(t-\frac{\tau}{2}\right)] \tag{4}$$

$$R_x(t,\tau) = \sum_{\alpha} R_x^{\alpha}(\tau) e^{i2\pi\alpha t} \tag{5}$$

However, the start of the CP is calculated by evaluating the peak value of the FFT of Equation 5, adjusted for the shift in frequency offset using only positive frequencies. This peak value represents the start of the CP length symbol of the total subcarriers. The CP is then found by subtracting the calculated peak value from the total length of subcarriers found in Equation 1.

The last two values of the cyclostationary features, subcarrier spacing and symbol time are found by using the first two parameters, as well as the sampling rate. First, the subcarrier spacing, the frequency of subcarrier spacing in the total OFDM signal, is found by dividing the sample rate by the total length of subcarriers, N, found in Equation 1. Symbol time is then calculated as the inverse of the subcarrier spacing. These four parameters become inputs to the AMC, using machine learning to recognize and determine if the calculated values match that of predefined UAS signals.

2.3 Automatic Modulation Recognition

Automatic Modulation Recognition (AMC), is a process that used machine learning classifiers to determine a modulation type of a signal. These classifiers are not defined to a certain machine learning approach and the RF-Class toolbox includes support for various approaches to be implemented. Due to the assumptions of the OFDM modulation scheme of the UAS signal from Section 2.2, AMC is then used to determine if those assumptions were correct. This technique is evaluating on a set of extracted features, which is why the machine learning algorithm, k-nearest neighbours (KNN) is used. The k-nearest neighbours' algorithm (k-NN) is a non-parametric method used for classification and regression [1]. This algorithm is evaluated in feature space, a transformed space evaluating distance metrics of closest data points of each feature in a predefined cluster size, k.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until function evaluation [1]. This offset of computation pairs perfectly with the SDR, allowing for a precomputed model to be loaded and then used to classify class membership of a set of features. This algorithm requires pre-recorded data sets from various UAS, compiled together to create a total training dataset. Currently, all data processing is done offline to dimension the dataset to be used effectively in the training phase. This process involves the clustering of data points and the computed nearest neighbour boundary plots, which are then used to create a model. This model is then a lookup for new instances of feature sets, that can be deployed in real-time to determine if an observed signal is that of a UAS in the library of signals.

2.4 Machine Learning Models

Using the AMC approach, models are created from captured signal data, compiled together to create a

training data set. Since the model is evaluated in feature space, the accuracy of the model is directly proportional to the accuracy of the data set to real UAS signals. The algorithm has the ability to predict on cases in which the observed signal doesn't precisely match any point in the training set, but requires the set to be broad enough. This broadness is defined to the environment the system is to be deployed in, but is able to vary environment to environment if they reflect similar captured scenarios.

Since, captured data in the training phase of the model is the most important metric, the RF-Class system focuses on data capturing optimization that allows a user to operate the system to create the data sets needed to train a model that can be used to classify the class membership of real-time feature capture. This process involves capturing data of various UAS scenarios, attempting to create a broad enough set that can accurately predict a rouge UAS signal in an environment.

The scenarios captured in this paper reflect: clear line of sight, shadowing, fading, unknown location, and pathloss due to environmental obstructions. These cases capture UAS under load and non-load conditions creating a dataset of a wide variety of conditions. Models loaded into the system for real-time calculations evaluate two or more classifications, with the class membership of "no signal present" as a case in every model. Since the evaluation time of a "model" or given dataset can be computed in seconds, the cycle of model creation from data capture can be repeated until a target level of accuracy is achieved.

2.5 Visualization

RF-Class's visualization block was created to have user friendly interfaces showing various received signals, wideband spectrums, verified modulation schemes, and class memberships of various UAS signals in the known library. These various plugins are integrated together to allow a user to choose what they want to see. In the flowgraph of the GNU Radio program, a user also has the ability to update parameters, change variables, and input computed models. This visualization is integrated in the GNU Radio framework and some of the interfaces are shown in figures 1 and 2. Figure 1 shows the user's ability to modify variables of one of the blocks, in this case, the cyclostationary feature extraction block. Conversely, figure 2 shows a frequency plot of captured wideband UAS signal time slice. This frequency plot shows the FFT of the discretized signal captured through the SDR.

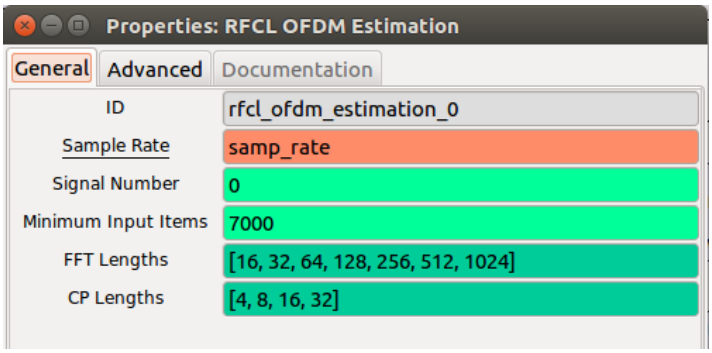


Figure 1. Cyclostationary Feature Extraction Block Parameters

The internal variables of this block set the variable initializations of the code performing the various data processing techniques. In this case, the variables above pertain directly to the parameter extraction techniques explained in section 2. The variable minimum input items allow the process of parameter extraction to be based on, in this case, 7000 or more measurements from the SDR, ensuring a valid output. In cases where less than this number of measurements are taken, the system continues to run until this number is reached. The FFT lengths variable sets a list of possible number of FFT sample size of subcarriers the system is able to choose. Since OFDM modulated signals multiplex a wideband signal into total carriers of 2^n, seven possible values are listed in the set. Finally, the CP lengths list verifies calculations made are of

those listed in the set. This account for slight inaccuracies in the calculation and make sure the calculated number is also a base 2 number.

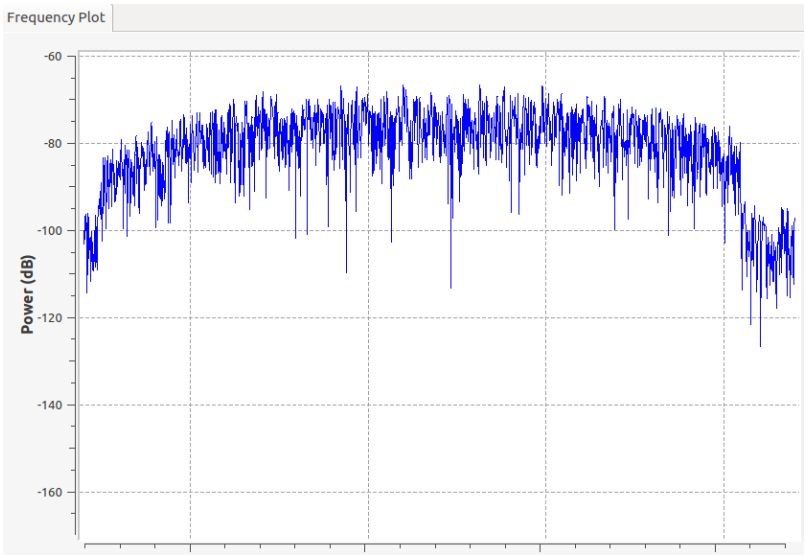


Figure 2. Frequency Plot of Captured UAS Signal

Lastly, an interface displaying the IQ data of the signal is shown below in figure 3. This plot shows measurements of a captured UAS signal before any signal analysis. This interface shows a scenario in which the UAS was operating under a load condition at a short distance range from the system.

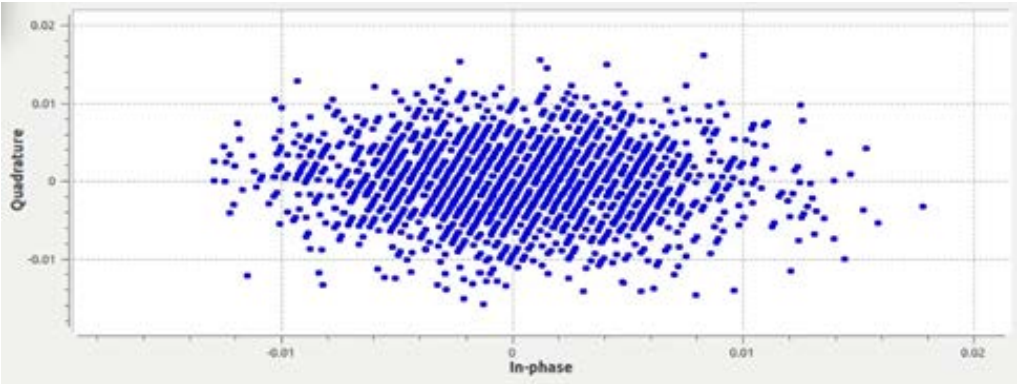


Figure 3. IQ Data Plot of Captured UAS Signal

3.0 APPROACH

3.1 Visualization

Testing and validation processes were completed for various UAS classes, environment conditions, and load conditions. In each case, the accuracy of the system was updated from a performance metric used in the training phase during the machine learning model creation phase. This performance metric used is called a confusion matrix. This matrix is responsible for evaluating true and false positives for predicted conditions versus the true condition. For this, Table 2 shows a basic confusion matrix showing four possible scenarios that can occur from a data points predicted class membership versus the true class membership of the point.

Table 2. Basic Confusion Matrix Validation Metric

		Actual Values	
		Positive	**Negative**
Predicted Values	**Positive**	True Positive	False Positive
	Negative	False Negative	True Negative

Table 3. RF-Class 2-Class Confusion Matrix

		Actual Values	
		Class 1	**Class 2**
Predicted Values	**Class 1**	True Positive	False Positive
	Class 2	False Negative	True Negative

A more useable performance metric specifically for the RF-Class system is shown in Table 3. Instead of a positive or negative validation, this case looks to classify between two different class memberships. The distribution of the results becomes a calculation of the instances of the four possible outputs of the classification. Starting in the top left of the bolded box, the true positive (TP) condition occurs when the system labels a data point as class 1 and it it's actual label, chosen during data collection, was also class 1. Conversely, the true negative (TN) case is the same as TP, but with class 2 instead. The two important outputs of this metric are when the outputs are false. These two outputs are what is used to tell which way the prediction system is leaning in incorrect classifying. The top right output, false positive, occurs when the system falsely predicts a class 2 point as class 1. The lower left is the opposite of this, incorrectly classifying a class 1 point as class 2. These validation parameters provide the user with useful data metrics showing if the system is leaning toward incorrectly classifying a certain class membership. As the basis of accuracy of the system, this is calculated by Equation 6 and a basic example using 10k total number of data points, correctly predicted 9,890 of those shown in Equation 7.

$$Accuracy = \frac{Total\ \#\ Correct\ Predictions}{Total\ \#\ of\ Data\ Points} * 100\% \quad (6)$$

$$Accuracy = \frac{9890}{10000} * 100\% = 98.9\%\ Accurate \quad (7)$$

3.2 RF-Class Experimental Test Setup

Using this performance metric, the system was then able to evaluate model performance after different datasets and conditions were loaded. The experimental test setup for lab testing is shown in Table 4. This experimental setup remained uniform across environments, with every metric explained in this paper following this experimental setup.

Table 4. Experimental Test Setup Components

Hardware	
Type	**Description**
Computer	Dell Laptop (i5, 2.3GHz, 16 Gb RAM)
SDR	Ettus Research B210 USRP
	Ettus Research B205 mini-i USRP
Antennas	2.4/5.8 GHz, 9 dBi, Directional Antenna
	2.4 GHz, 12 dBi, Omni-Directional Antenna
	5.8 GHz, 16 dBi, Directional Antenna
	2.4 GHz, 17 dBi, Directional Antenna
UAS	DJI Mavic 2
	DJI Phantom 4 V2
	DJI Mavic Air
Software	
GNU Radio Companion (GRC)	

This table outlines all of the possible components used for testing the RF-Class system. For the different types of hardware listed: both SDRs used in testing, all four of the possible antennas, and all three DJI Drones. Additionally, the software was GNU Radio Companion, a GUI version of the GNU Radio toolbox with custom RF-Class coded blocks performing various data processing. As mentioned before, GNU Radio is a signal processing framework creating a flowgraph of connected blocks to connect data to each other. Figure 4 shows the custom GNU Radio flowgraph of the RF-Class system, with all of the blocks used in compilation shown.

This flowgraph, when ran, outputs a GUI that allows a user to interact with various interfaces and visualize the real-time data from section 2.5. This flowgraph is unique to the RF-Class system and any block containing RFCL is a custom block. As mentioned in section 2.5, each unique interface of the system has the ability to be turned on/off to allow the user to display only useful information when performing various testing scenarios.

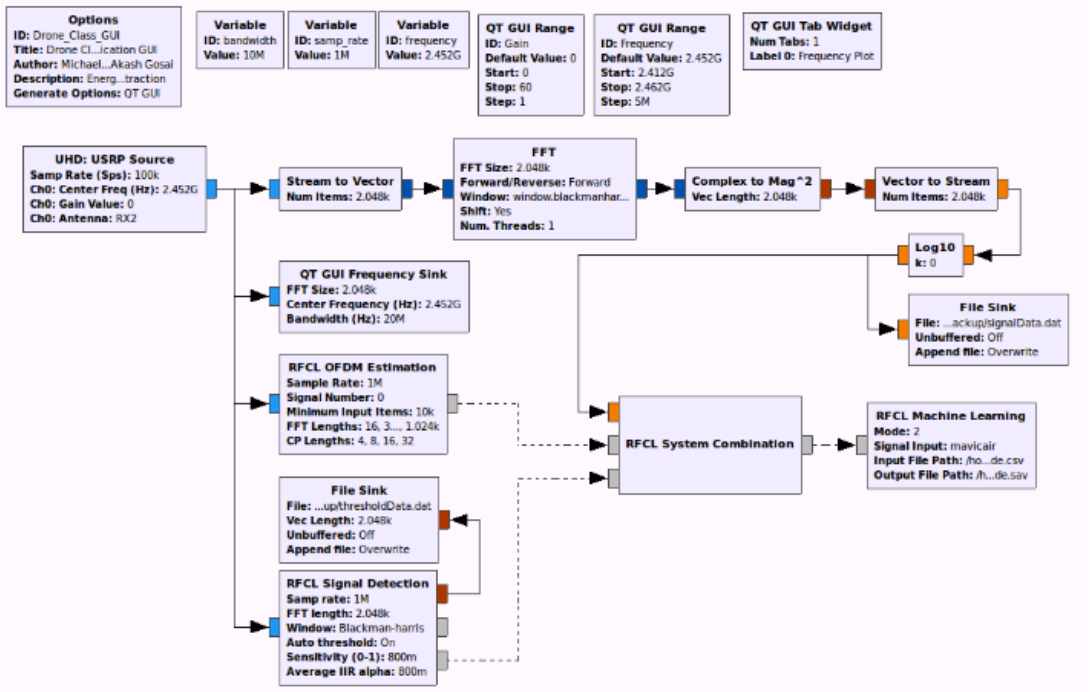


Figure 4. RF-Class GNU Radio Custom Flowgraph

The test setups include three basic scenario categories: clear line of sight, shadowing/fading, and no line of sight. Various test cases were merged into these three categories to simplify data verification and model accuracy. Starting with the first category, clear line of sight implies that the distances in the results are the drone-to-receiver distances and the user was able clearly see the UAS with minimal environmental obstructions. The second category, shadowing/fading, introduces various environmental obstructions with some line-of-sight knowledge. This category would imply the user is able to see the UAS, but there would be multiple obstructions in the path. The last category tests the system's ability to perform under a substantial signal loss, due to large obstructions. This category of tests generally involved positioning the UAS in front of the building and running the system from the back (~50m building) and determining if the UAS was detected.

3.3 Experimental Results

The RF-Class system was successfully able to detect the presence of UAS signals in various environments. These scenarios were captured with as many constants between tests as possible. For each of the testing sites, this paper focuses on the DJI Mavic 2 UAS, showing the system performance of the same setup across the three environments, with each providing new findings. For each of these environments, the same testing protocols and amount of data collected remained the same. This setup included the B210 USRP, 2.4/5.8 GHz, 9 dBi, Directional Antenna and the DJI Mavic 2 UAS. Additionally, a new model was created for each environment based on newly captured data.

The results of the system performance in detecting the presence of the DJI Mavic 2 drone outdoors at Test Site #1 for the three categories are Tables 5, 6, and 7 respectively. With the ability to test the RF-Class system in a smaller environment, the system was updated until the precision of the different tests produced the same intended results. This system reliability in the slightly rural environment of Test Site #1, proved not only functionality for the system but the baseline achievable performance.

Table 5. Clear Line of Sight (Test Site #1)

Number of Samples = 5000 (Random 20% of 25k)					
Distance (m)	Accuracy (%)	TP	FP	TN	FN
5	99.98	2519	0	2480	1
25	99.84	2514	5	2478	3
50	99.34	2501	18	2466	15
100	99.46	2502	17	2471	20
125	98.82	2483	36	2458	23
150	95.34	2311	124	2456	109

Table 6. Shadowing/Fading (Test Site #1)

Number of Samples = 5000 (Random 20% of 25k)					
Distance (m)	Accuracy (%)	TP	FP	TN	FN
Shadowing/Fading	88.57	2134	243	2294	329

Table 7. Clear Line of Sight (Test Site #1)

Number of Samples = 5000 (Random 20% of 25k)					
Distance (m)	Accuracy (%)	TP	FP	TN	FN
Unknown Location	81.22	1987	563	2074	376

After the extensive testing at Test Site #1, we wanted to prove the performance in a new testing environment, but keeping the same testing protocols. In a more rural environment, Test Site #2, we were able to test the system functionality on a larger dataset and extending the line-of-sight distance to as far as the environment would allow. The results of this test for the three categories are shown in Tables 8, 9, 10, respectively.

Table 8. Clear Line of Sight (Test Site #2)

Number of Samples = 20000 (Random 20% of 100k)					
Distance (m)	Accuracy (%)	TP	FP	TN	FN
5	99.98	10040	0	9956	4
100	99.34	10156	43	9712	89
200	98.78	10012	112	9744	132
300	96.49	9842	435	9456	267
400	94.23	8967	641	9879	513

Table 9. Shadowing/Fading (Test Site #2)

Number of Samples = 20000 (Random 20% of 100k)					
Distance (m)	Accuracy (%)	TP	FP	TN	FN
Shadowing/Fading	89.22	8156	1149	9688	1007

Table 10. Clear Line of Sight (Test Site #2)

Number of Samples = 20000 (Random 20% of 100k)					
Distance (m)	Accuracy (%)	TP	FP	TN	FN
Unknown Location	87.22	8912	1513	8532	1042

The last testing scenario again occurred in a new testing environment, Test Site 3. This site was the most rural of the three and allowed for us to test the maximum limits of the system, pertaining to line of sight, even when we were able to see the UAS. This test was based on the testing protocols of Test Site #2, including the increased amount of collected data, thus proving a broad range of possibilities of the system. This test didn't allow for the no light of site category to be tested, due to environment conditions. The results of this test for the first two categories are shown in Tables 11 and 12 respectively.

Table 11. Clear Line of Sight (Test Site #3)

Number of Samples = 20000 (Random 20% of 100k)					
Distance (m)	Accuracy (%)	TP	FP	TN	FN
100	99.98	10040	0	9956	4
200	99.34	10156	43	9712	89
500	98.77	10012	111	9745	132
600	93.14	9842	435	9456	267
700	91.88	8859	740	9767	632
800	87.20	9069	1254	8371	1306
900	81.09	8271	1967	7947	1815
1000	74.67	7915	2685	7019	2381

Table 12. Shadowing/Fading (Test Site #3)

Number of Samples = 20000 (Random 20% of 100k)					
Distance (m)	Accuracy (%)	TP	FP	TN	FN
Shadowing/Fading	64.58	6845	3755	6071	3329

3.4 Discussion

The above section of results outlines the various conditions in which the RF-Class system was put through in order to reliably and accurately detect the presence of a UAS signal in an environment. These extensive testing setups allowed critical refinements to the machine learning algorithm and models, resulting in a large dataset pertaining to each Test Site. Additionally, we were able to test one model in a different environment after some basic environment thresholding was completed, proving the success of a pretrained model to seamlessly flow between environments. The results concluded with a rough UAS detection radius of 1 Km and allowed a combined model across two environments to increase the model accuracy by ~3.7%, bringing the average detection accuracy of the model in every case to above 90%.

4.0 FUTURE WORK

This research focused mostly on the completed and verified work of UAS detection between that of a non-UAS signal. In doing this, we were introduced to the various machine learning algorithms and approaches to solve AMC and drone classification. However, as the current approach and conclusions were formulated, we started to notice the inaccuracies of the system's ability classify multiple UAS signals in the same

environment simultaneously. This observation arose from the outputs of each UAS signals that did not change from class to class. Since all of the three drones were from the same manufacturer, the chances of the same signal modulation type and parameters was high, with a defining characteristic not yet found.

Moving forward, our focus will remain on multi-UAS classification in the same environment. We see this problem accomplished with the following four sequential flow of milestones and predicted scenarios:

- Multi-UAS devices across classes present in the same band at the same time. This would imply that the UAS devices are not interfering with each other and operating on different channels within that band.
- Multi-UAS duplicate devices of the same class present in the same band at the same time. Again, this would imply that the UAS devices are not interfering with each other and operating on different channels within that band.
- Multi-UAS devices across classes operating on the same frequency channel within a specified band, leading to some signal overlap and/or interference
- Multi-UAS duplicate devices of the same class operating on the same frequency channel within a specified band, leading to some signal overlap and/or interference

This approach involves additional features that are used in the AMC and machine learning algorithms. We see this parameter still being part of the physical signal and are identifying feature candidates that would achieve the same level of accuracy as the system moves from single UAS detection to multiple. We are currently working on these approaches and don't reflect any updates to the processes outlined in this paper.

Moving into a multi-UAS classification, the addition of a feature is hypothesized to solve the classification problem. We plan to do this with the following three initial feature candidates or signal processing functions:

- Radio Frequency Band Allocation – this would provide the system with a windowing capability that would be able to simultaneously monitor an entire frequency without having to sweep the band. This monitoring would require at least two devices, each responsible for a section of channels in the frequency band. This initially would allow for multiple UAS to be detected if multiple devices are monitoring separate channels. Figure 5 below shows an initial graphic showing two roughly separated set of channels for each of the USRP devices, effectively covering the entire band simultaneously. The blue boxes show the captured channels in which each USRP would monitor.

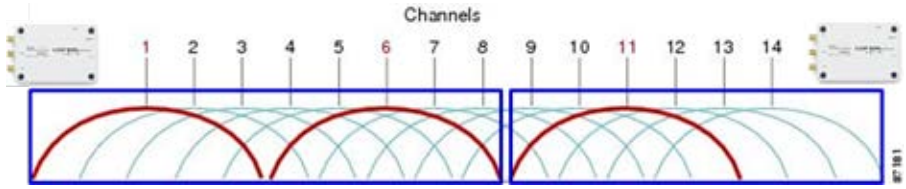


Figure 5. Proposed Band Allocation Approach

- IQ Data Processing – The approach would investigate both amplitude and phase of various UAS signals, adding these features to the AMC machine learning model. This approach involves data processing using spectral correlation functions providing values of cyclo-stationary nature of signal, which is a useful function explaining specific UAS devices spectral correlation as observed by the system. Lastly, this would analyse time slices of a specific drone under load and non-load conditions.
- Machine Learning Data Optimization – This last approach provides the system with the ability to use optimized data sets and well as a more functional UI in order for data collection to be easier,

providing more automation to the system. This would add the addition of environmental thresholding max value to capture full signal amplitude of drone, input of amplitude and phase from signal data stream, and include spectral correlation function output into model's feature inputs. Lastly, this would provide a user the ability to perform data collection for multiple UAS devices, perform machine learning training, and finish and load machine learning models.

5.0 CONCLUSION

The RF based UAS detection scheme provides an agnostic approach to detecting and identifying presence of UAS in several SNR regimes. In addition, this capability will also aid operators in eliminating non-UAS signals in physical space and focus their detection in physical spaces where UAS signals have been detected. This capability will improve the efficiency of detection as prior knowledge of RF fingerprints will aid in ensuring every detection will only focus on unknown RF signal detection. This would eventually help operators to speed up the detection and identification of UAS devices.

REFERENCES

[1] Wikipedia Contributors, En.wikipedia.org. *K-Nearest Neighbors Algorithm*. 2020

[2] Jithin Jagannath, Hanne M. Saarinen, and Andrew L. Drozd. Framework for Automatic Signal Classification Techniques (FACT) for Software Defined Radios. 10.1109/CISDA.2015.7208628.

[3] M. L. D. Wong and A. K. Nandi, "Automatic digital modulation recognition using spectral and statistical features with multi-layer perceptrons," in Proc. Int. Symp. Signal Processing and Its Applications, Kuala Lumpur, Malaysia, 2001, pp. 390-393.

[4] H. Deng, M. Doroslovacki, H. Mustafa, J. Xu, and S. Koo, "Instantaneous feature-based algorithm for HF digital modulation classification," in Proc. CISS Conf., 2003, John Hopkins University, Baltimore, MD, US.

[5] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," IEEE Trans. Commun., vol. 48, pp. 416-429, 2000.

[6] O. A. Dobre, A. Abdi, Y. Bar-Ness and W. Su, "Selection combining for modulation recognition in fading channels," in Proc. IEEE MILCOM, 2005, pp. 1-7

[7] J. Lopatka and M. Pedzisz, "Automatic modulation classification using statistical moments and fuzzy classifier," in Proc. IEEE ICSP, 2000, pp. 1500-1505.

[8] H. Yoshioka, Y. Shirato, I. Toyoda, and M. Umehira, "A fast modulation recognition technique using nearest neighbour rules with optimized threshold for modulation classification in Rayleigh fading channels," in Proc. IEEE Wireless Personal Multimedia Communications Conf., 2002, pp. 1049-1052.

[9] Ettus Research Knowledge Base by National Instruments, https://www.ettus.com/

Machine Learning Empowered Radio Frequency Signal Classification For UAS Detection

Michael Nilsen

Research Assistant, Virginia Modeling, Analysis and Simulation Center, Old Dominion University

DSP Engineer, Johns Hopkins University Applied Physics Lab, Laurel, MD US

Sachin Shetty

Associate Director - Virginia Modeling, Analysis and Simulation Center,

Associate Professor, Department of Computational Modeling and Simulation Engineering

Old Dominion University, Norfolk, VA USA

Kimberly Gold

Engineer

Naval Surface Warfare Center

Crane, Indiana USA

Charles Kamhoua

Senior Electronics Engineer

Network Sciences Division

Army Research Lab

Adelphi, MD USA

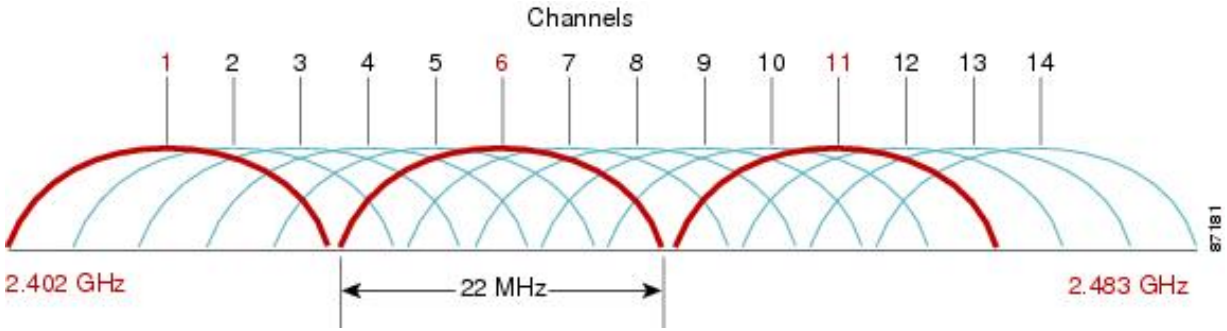
Overview

- **Project Goal**
  - Develop an automated RF classification capability in GNU Radio that can automatically monitor, detect and classify UAS signals across multiple bands
- **Benefits**
  - Detect and classify UAS signals to create a library of known UAS signals.
  - Craft a waveform and launch a protocol attack on the adversarial UAS signal.
  - Signal energy detection, automatic modulation recognition and classification will be useful for C-UAS efforts.

UAS Signal Detection

Bands

- **Band 1: 433 MHz ISM Band: 433.05 - 434.7**
- **Band 2: 2.4 GHz – WiFi g/b/n: ≈ 2.4 – 2.5 GHz**
- **Band 3: 5.8 GHz - WiFi a/h/j/n**

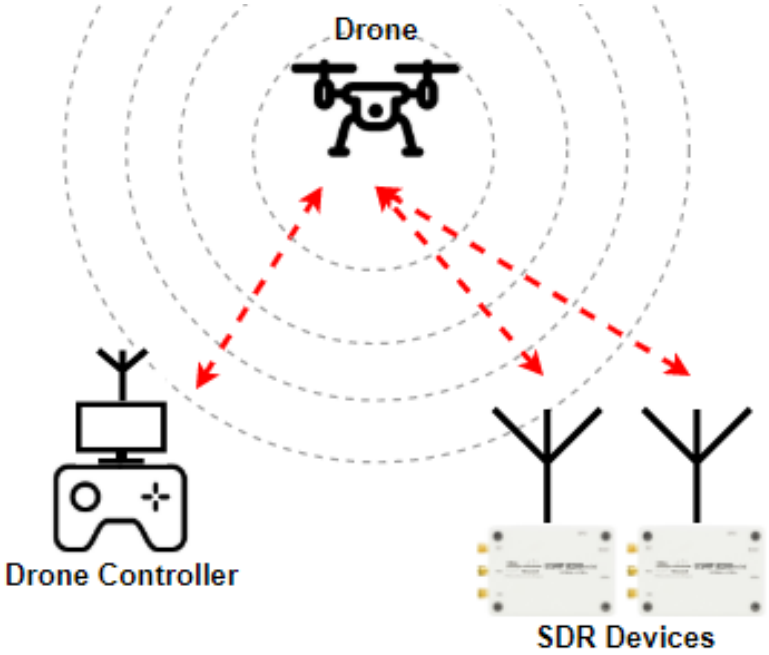


Signal

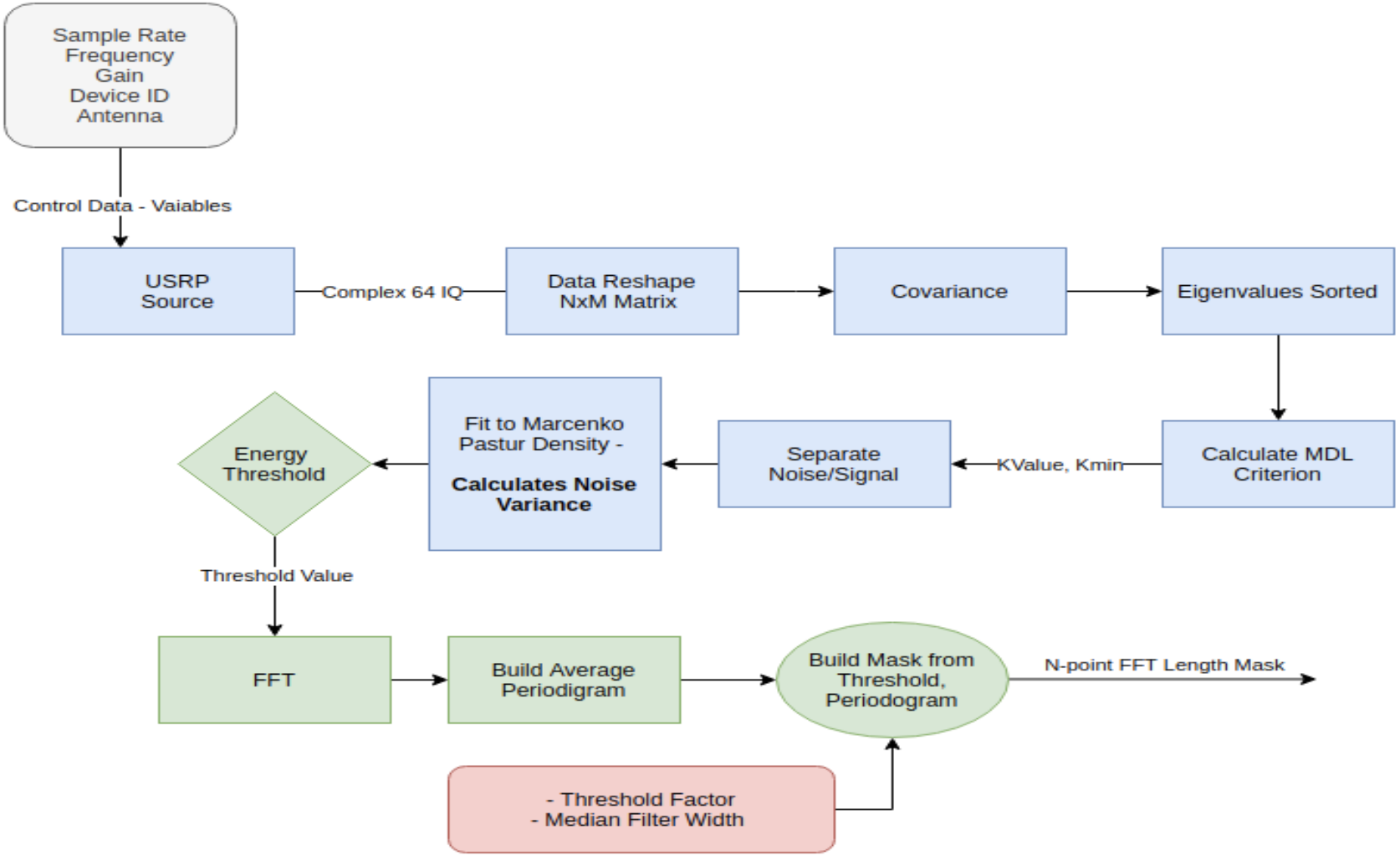
- **OFDM (orthogonal frequency division multiplexing): Drone specific transmit frequency channel will be selected**

Background

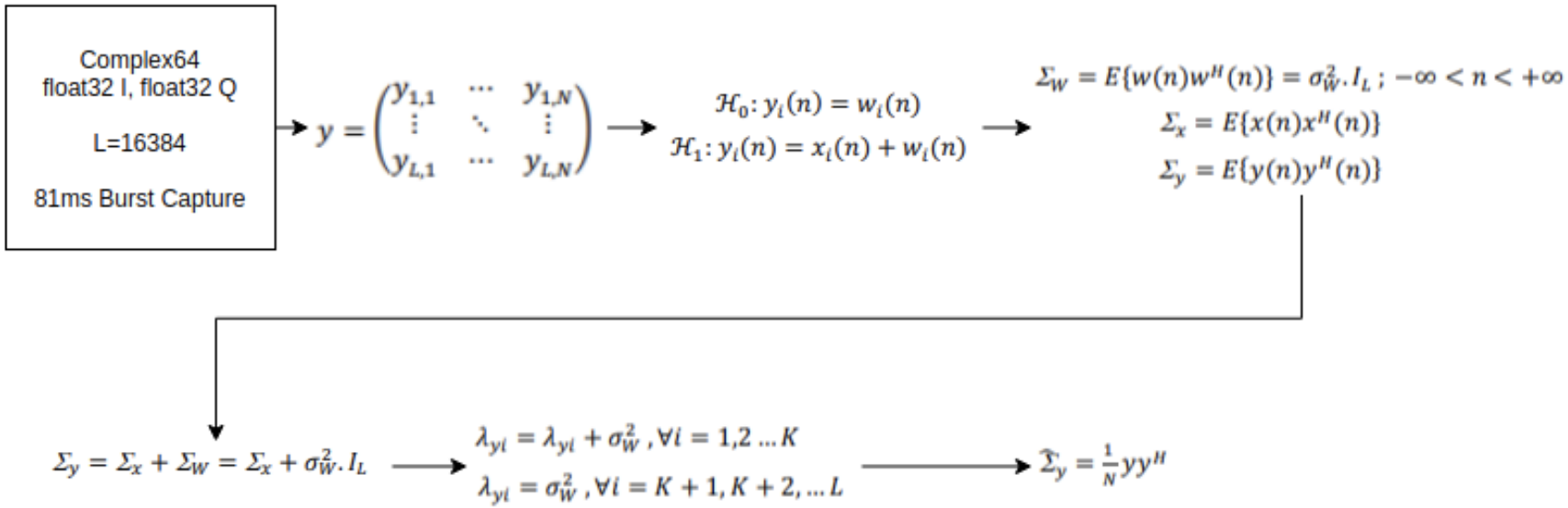
- **Radio Frequency Detection**
  - **Adaptive Signal Energy Detection**
  - **Modulation Scheme Recognition**
  - **Machine Learning Model**
  - **Multi-UAS Classification Approach**
    - **Frequency Band Allocation**
    - **ML Feature Data Processing**
    - **Signal Energy Transient Analysis**



Adaptive Energy Detection



Data Stream to Eigenvalues



- **Complex IQ Stream is converted to a sample covariance matrix in order to calculate eigenvalues**
  - **Signal noise (captured) is independent of signal**
  - **Eigenvalues are used to determine power threshold of noise/signal**

MDL Criterion

$$K = \arg\min_K \left(-(L-K)N \log \frac{\varphi(K)}{\theta(K)} + \frac{1}{2}K(2L-K)\log N\right); 0 \le K \le L-1$$

Where $\varphi(K)$ and $\theta(K)$ are given by:

$$\varphi(K) = \prod_{i=K+1}^{L} \lambda_i^{\frac{1}{L-K}}$$

$$\theta(K) = \frac{1}{L-K}\sum_{i=M+1}^{L} \lambda_i$$

$$\sigma_{W1}^2 = \frac{\lambda_L}{(1-\sqrt{p})^2}$$

$$\sigma_{W2}^2 = \frac{\lambda_{\hat{K}+1}}{(1+p)^2}$$

- **MDL (Minimum Description Length) is used to determine K, index of eigenvalues separating noise and signal**
  - **This function returns both index and noise variance**

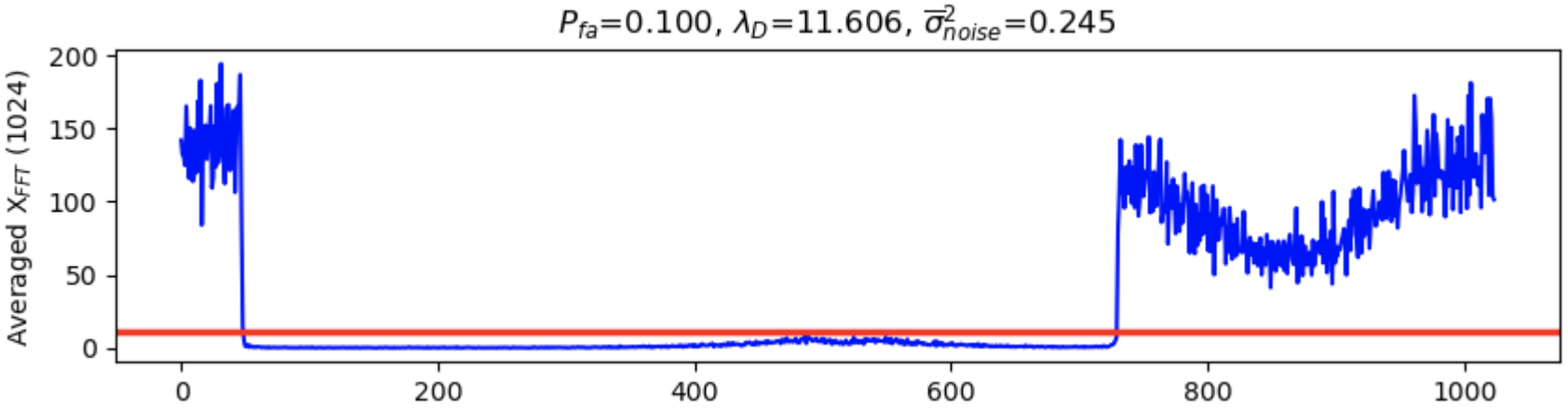
Marcenko Pasture Density

$$mp(p,\sigma_w) = dF^W(Z)$$

$$= \frac{\sqrt{(z-\sigma_w^2(1-\sqrt{p})^2)(\sigma_w^2(1+\sqrt{p})^2 - z)}}{2\pi\sigma_w^2 zp} dz \longrightarrow \lambda_D = \widehat{\sigma_w^2}\,(Q^{-1}(P_{fa})\sqrt{2N} + N))$$

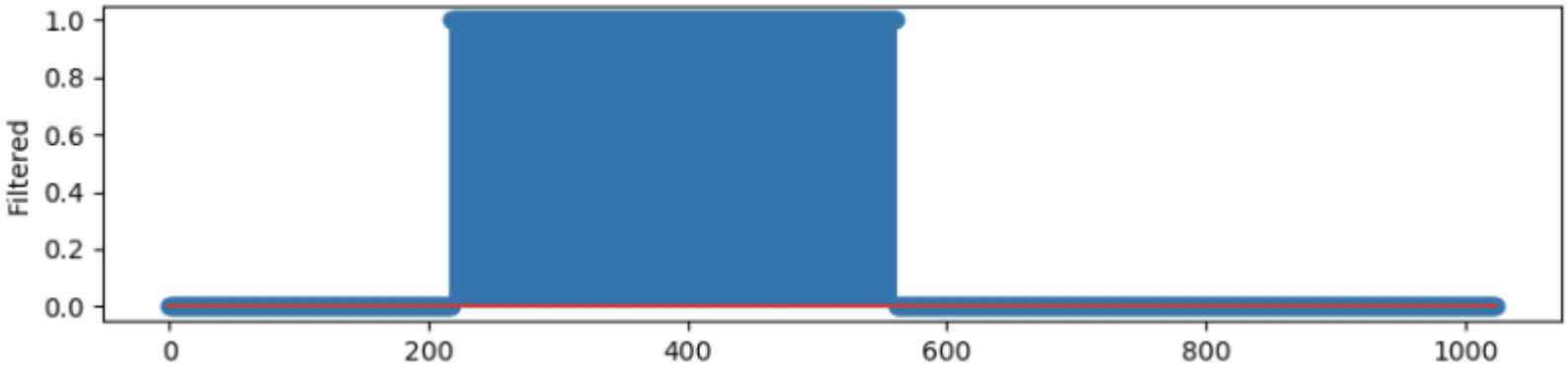
Where

$$\sigma_w^2(1-\sqrt{p})^2 \le z \le \sigma_w^2(1+\sqrt{p})^2$$

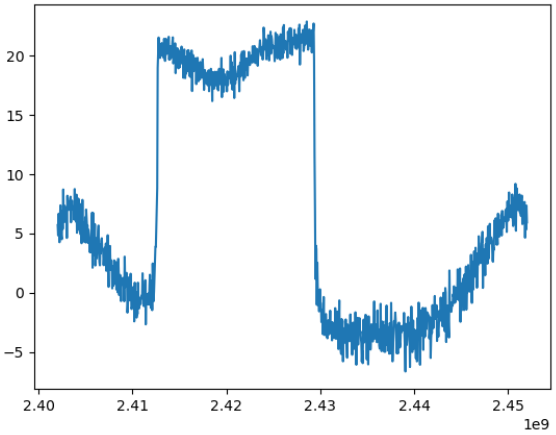


- **Returns noise/signal threshold used to build mask of periodogram**

Masking Function



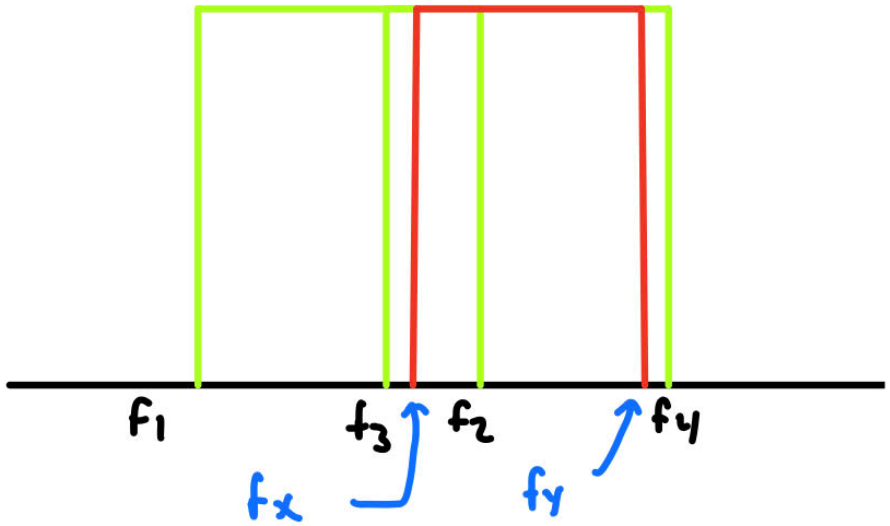
- **Mask is built with FFT bins where the value is greater than calculated threshold value**
  - **Returns a binary array of length L = NFFT**
  - **Mask span is converted to frequency values and interpolated based on a priori knowledge**



Channel Overlap

- Goal is to provide the distributed USRPs with information on which channel to tune to
- Function is needed to convert binary mask to start frequency, stop frequency, and channel span
- Masking calculation occurs with some error due to wide band signal captures from UAS devices
- Channel overlap function calculate the % overlap with an default channel on the observed band (2.4/5.8GHz)

Channel Overlap



Channel X = [f1 , f2]

Channel Y = [f3 , f4]

Unknown Channel U = [fx , fy]

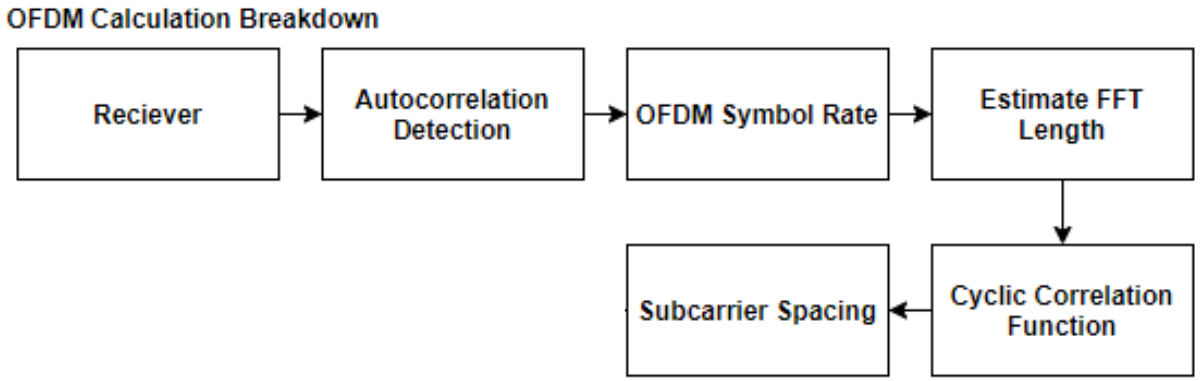
Feature Extraction

OFDM Parameters

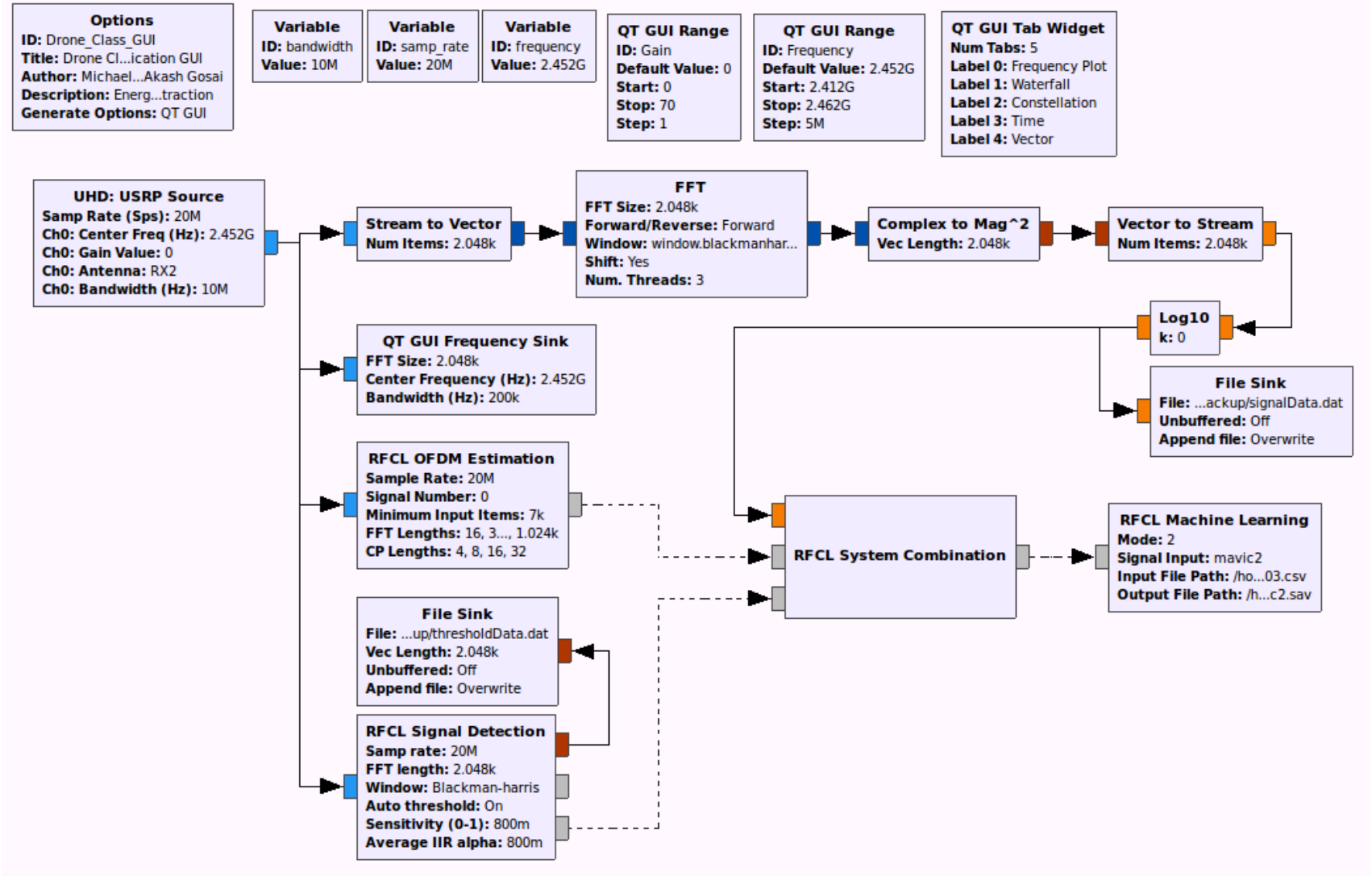
- **Cyclic Prefix Length :** Circular extension added to OFDM symbols in order to eliminate inter symbol interference
- **FFT Size:** Fast Fourier transformation of the digital signal to frequency domain using different bin sizes 64, 128, 256, 512, 1024…
- **Subcarrier Spacing:** spacing of the subcarriers of the OFDM symbol = 1/Ts, Ts (Symbol Time)
- **OFDM Symbol Duration:** Length of symbol time

Modulation Scheme Recognition

- **Blind OFDM can be detected by calculating the autocorrelation of the cyclic prefix of the received signal**
- **FFT lengths are estimated to determine the number of signal subcarriers**
- **The breakdown below outlines each step in the recognition process.**

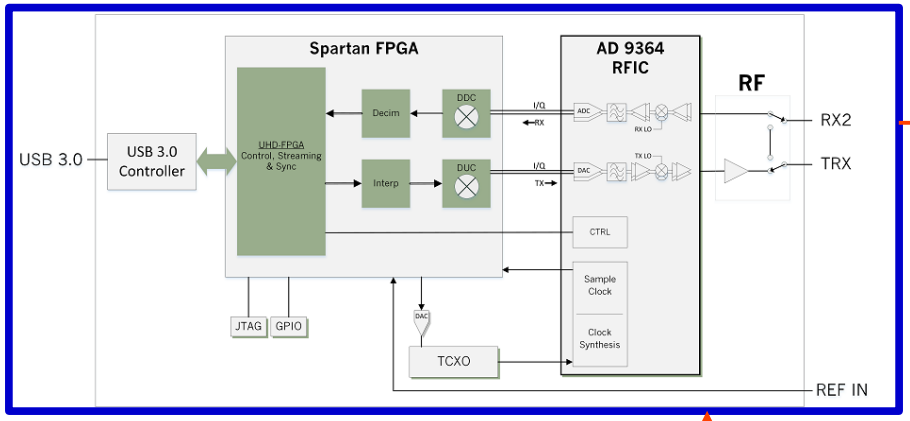


RF Classification Library



SDR Implementation

Ettus B205-1 Block Diagram



Complex64 IQ

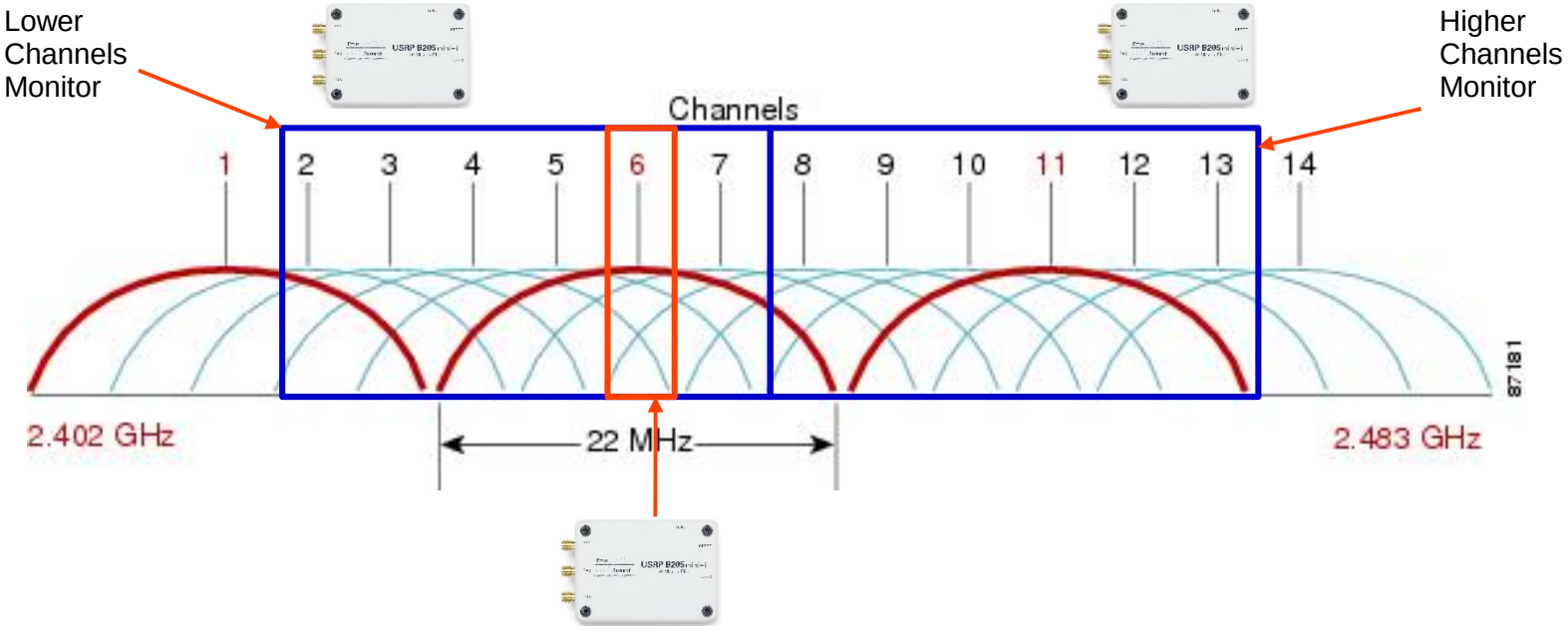
Host PC

GNU Radio

- Signal Processing
- Energy Detection
- OFDM Synchronization
- OFDM Demodulation
- Offline Data Collection
- Machine Learning Model

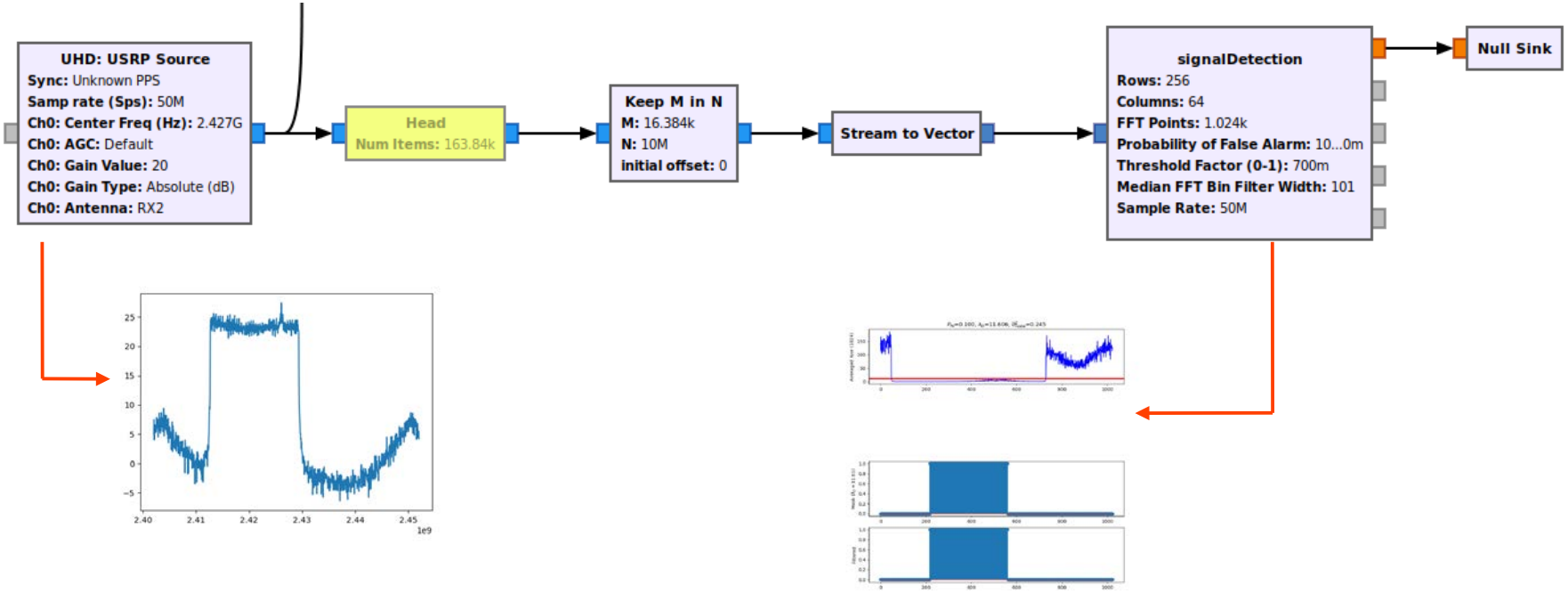
Control

SDR Implementation



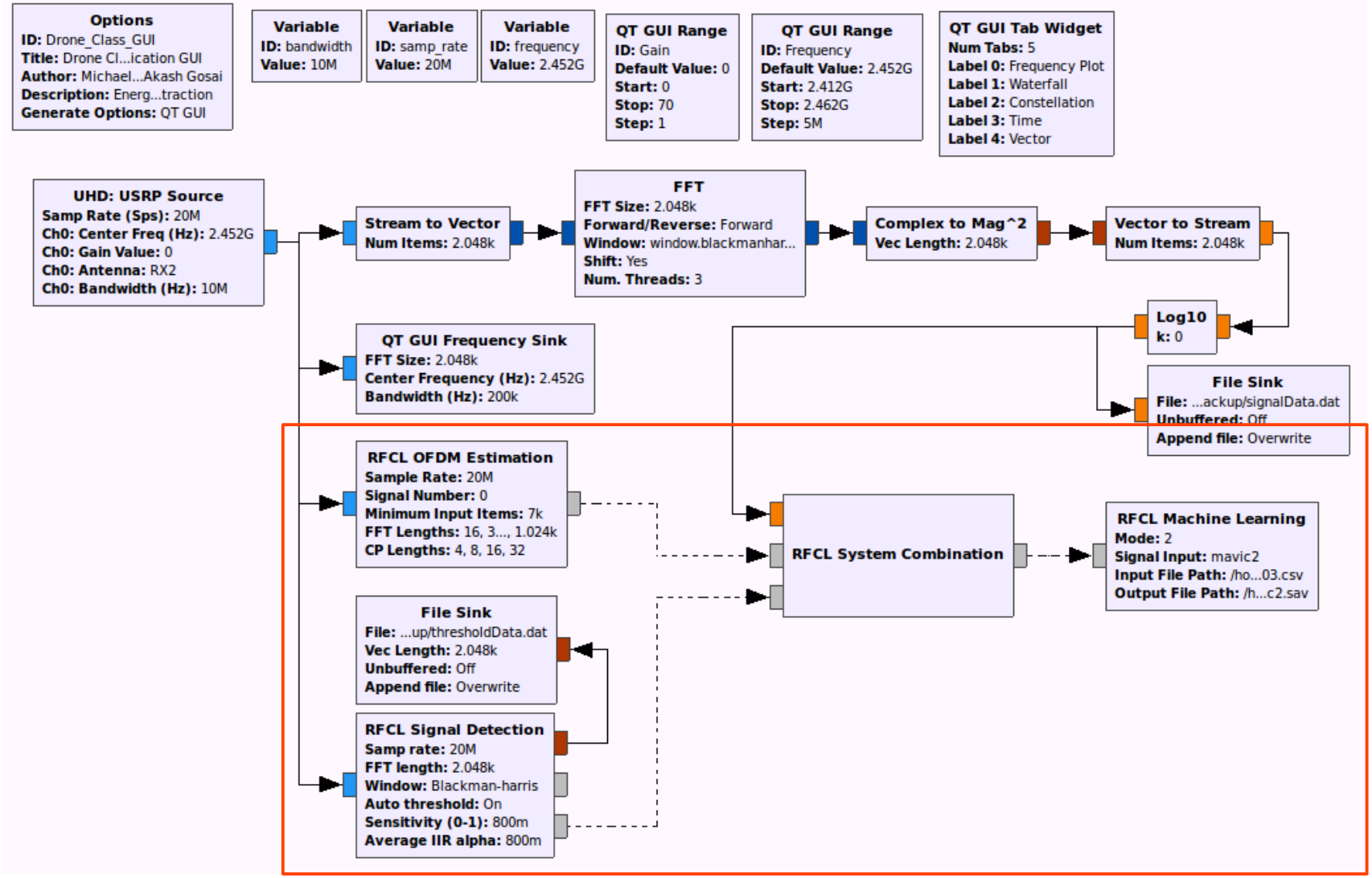
- Multiple B205 SDR split up the spectrum available by continuously monitoring and running an algorithm to find channels of potential interest
- A third B205 SDR (or B210) is then able to tune to a specific channel and determine the second phase drone identification algorithm. But each device can potentially split is 61.4 bandwidth into multiple streams.

Phase 1 GNU Radio



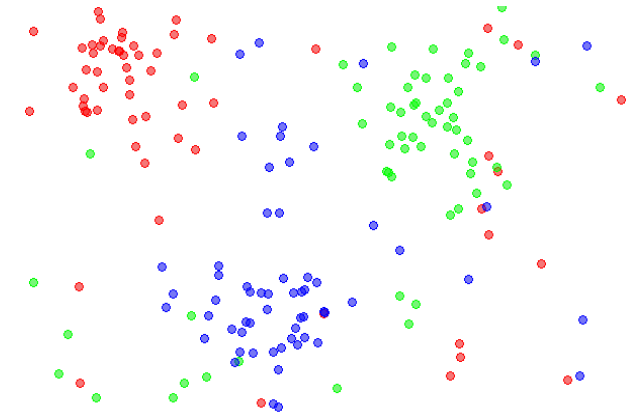
- Phase 1 algorithm is responsible for determining a channel that is occupied or potentially occupied with a drone. Phase 2 is then responsible for determining if the channel is occupied and then to classify the drone on the channel.
- Replaces current energy detection algorithm

Phase 2

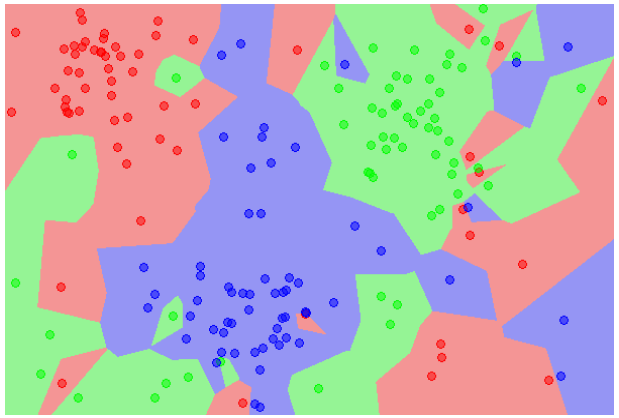

Options
ID: Drone_Class_GUI
Title: Drone Cl...ication GUI
Author: Michael...Akash Gosai
Description: Energ...traction
Generate Options: QT GUI
Variable
ID: bandwidth
Value: 10M
Variable
ID: samp_rate
Value: 20M
Variable
ID: frequency
Value: 2.452G
QT GUI Range
ID: Gain
Default Value: 0
Start: 0
Stop: 70
Step: 1
QT GUI Range
ID: Frequency
Default Value: 2.452G
Start: 2.412G
Stop: 2.462G
Step: 5M
QT GUI Tab Widget
Num Tabs: 5
Label 0: Frequency Plot
Label 1: Waterfall
Label 2: Constellation
Label 3: Time
Label 4: Vector
UHD: USRP Source
Samp Rate (Sps): 20M
Ch0: Center Freq (Hz): 2.452G
Ch0: Gain Value: 0
Ch0: Antenna: RX2
Ch0: Bandwidth (Hz): 10M
Stream to Vector
Num Items: 2.048k
FFT
FFT Size: 2.048k
Forward/Reverse: Forward
Window: window.blackmanhar...
Shift: Yes
Num. Threads: 3
Complex to Mag^2
Vec Length: 2.048k
Vector to Stream
Num Items: 2.048k
Log10
k: 0
QT GUI Frequency Sink
FFT Size: 2.048k
Center Frequency (Hz): 2.452G
Bandwidth (Hz): 200k
File Sink
File: ...ackup/signalData.dat
Unbuffered: Off
Append file: Overwrite
RFCL OFDM Estimation
Sample Rate: 20M
Signal Number: 0
Minimum Input Items: 7k
FFT Lengths: 16, 3..., 1.024k
CP Lengths: 4, 8, 16, 32
RFCL System Combination
RFCL Machine Learning
Mode: 2
Signal Input: mavic2
Input File Path: /ho...03.csv
Output File Path: /h...c2.sav
File Sink
File: ...up/thresholdData.dat
Vec Length: 2.048k
Unbuffered: Off
Append file: Overwrite
RFCL Signal Detection
Samp rate: 20M
FFT length: 2.048k
Window: Blackman-harris
Auto threshold: On
Sensitivity (0-1): 800m
Average IIR alpha: 800m

K-Nearest Neighbors Algorithm

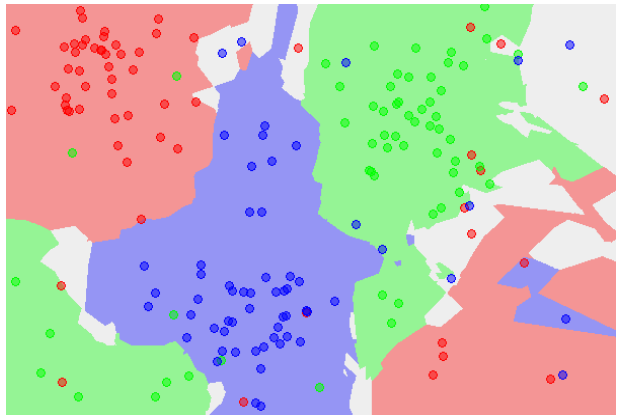
- **In KNN, the input consists of the k closest training examples in the feature space.**
- **Parameter selection and data collection are the two time consuming parts of the machine learning process**



Dataset [4]



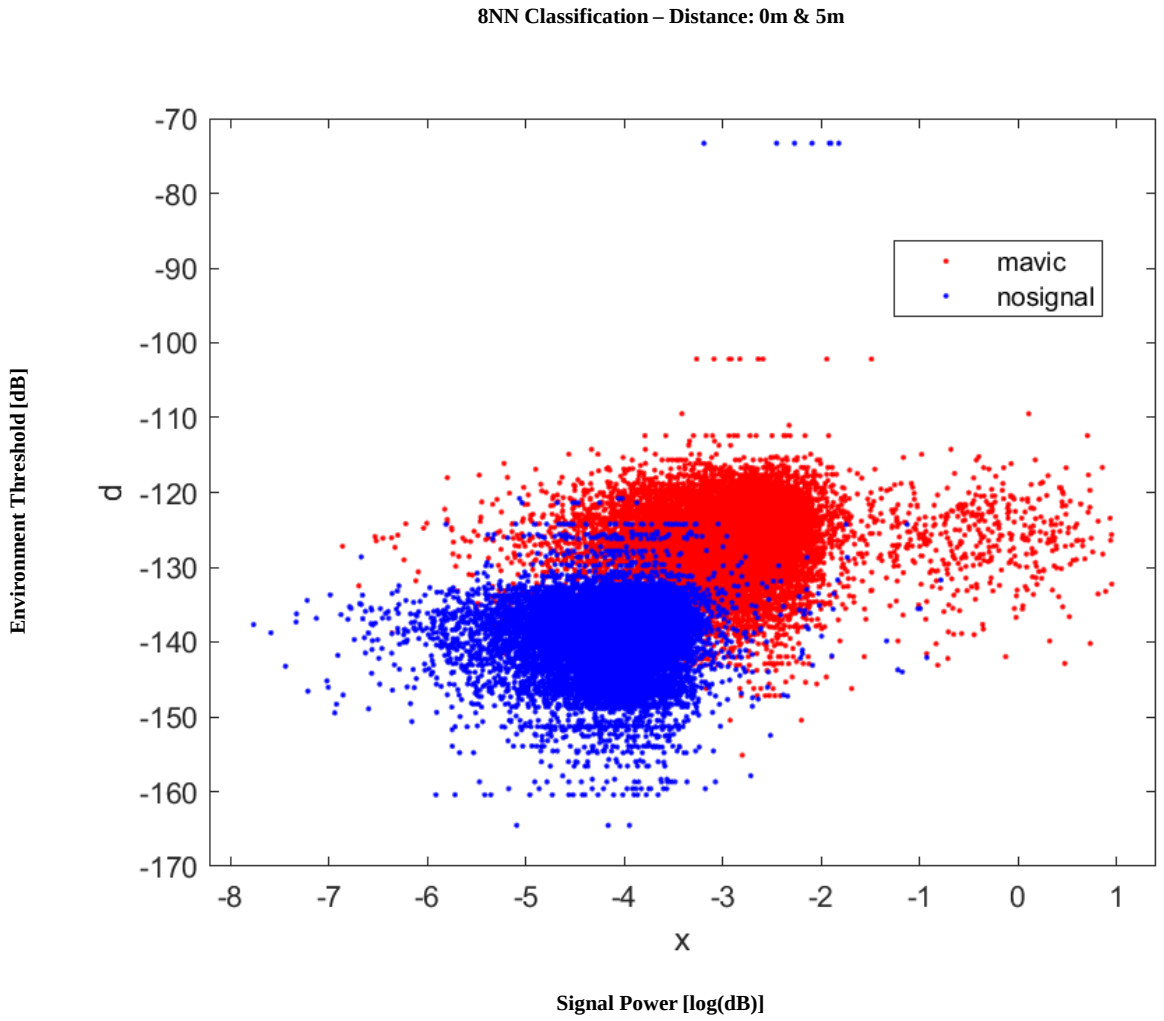
1NN Classification Map [4]



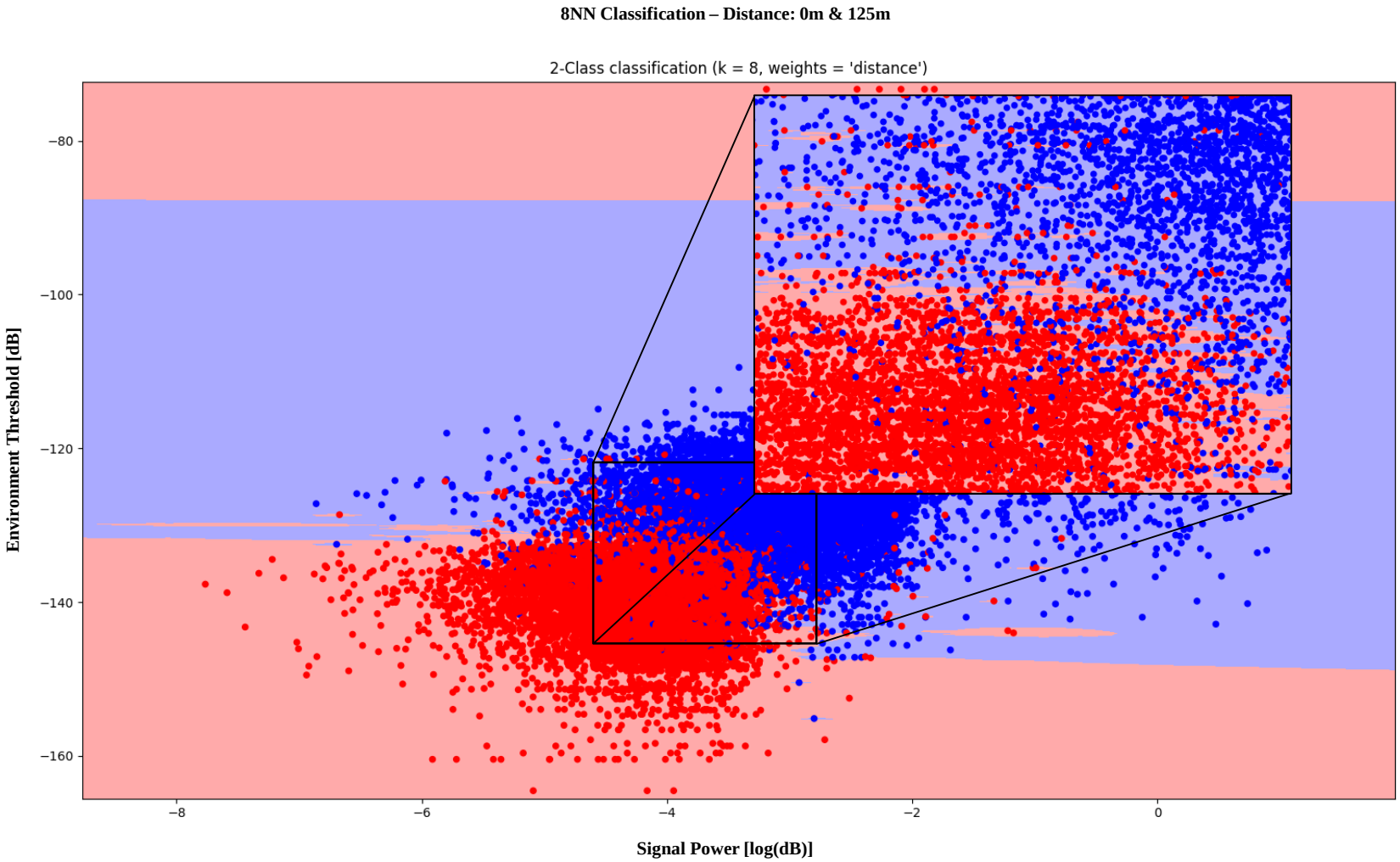
5NN Classification Map [4]

Source: Wikipedia contributors. (2021, February 21). K-nearest neighbors algorithm. In Wikipedia, The Free Encyclopedia.

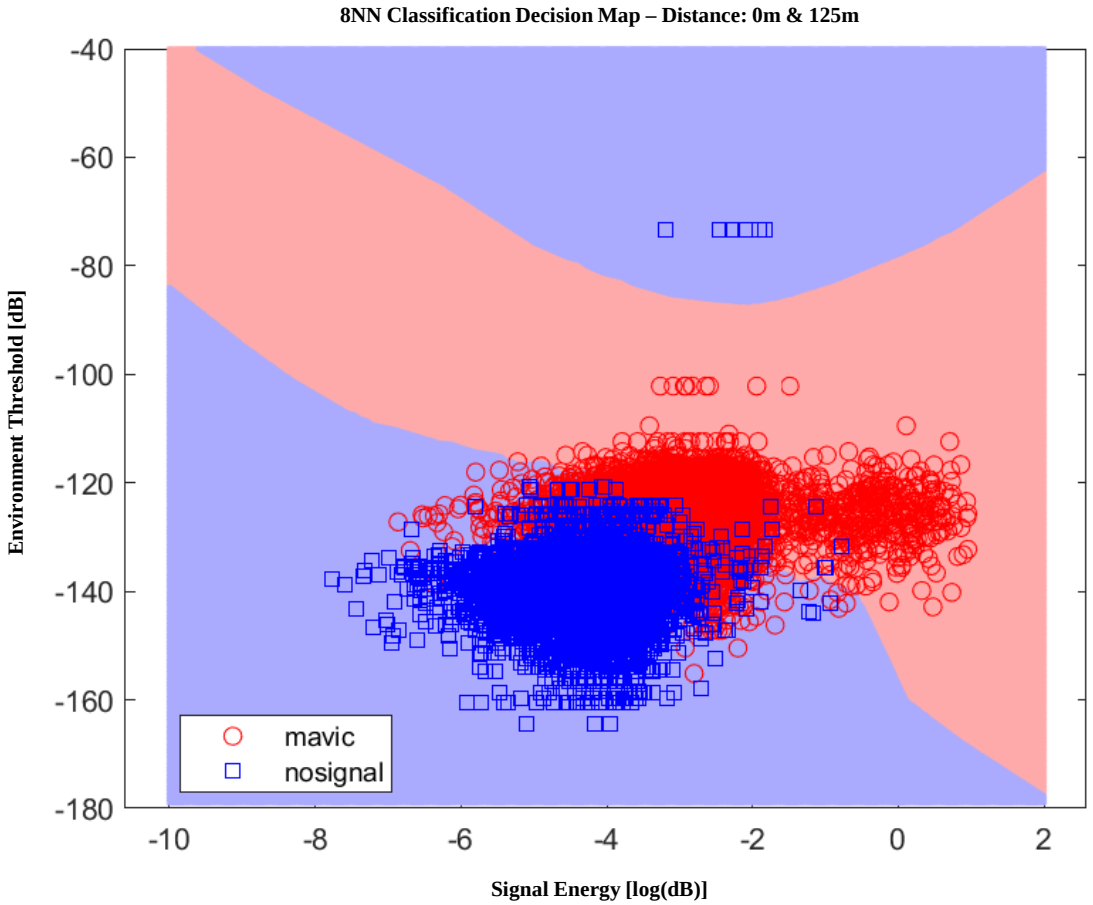
Machine Learning Model



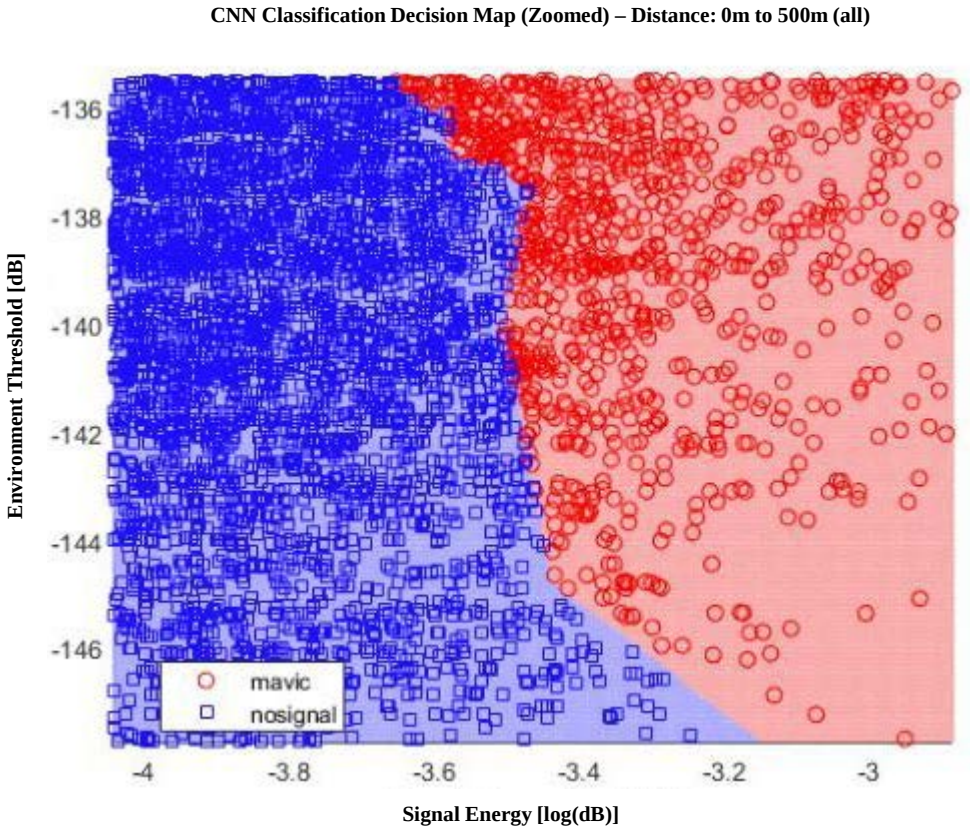
Machine Learning Model



Machine Learning Model



Machine Learning Model



Testing and Evaluation

- **Lab Testing**
- **Site 1 Testing**

Experimental Setup

S/N	Hardware	Software
1	Dell Laptop (i5, 2.3 GHz, 16 Gb RAM)	GNU Radio Companion
2	Ettus Research B210 USRP SDR	
3	2.4/5.8 GHz, 9 dBi Directional Antenna	
4	DJI Mavic 2	
5	DJI Phantom 4 V2	

Lab Testing

- **Data Acquisition** – Capture RF data from drones operating in indoor and outdoor settings for machine learning training
- **Model Building** - Train machine learning model on one drone class
- **Evaluation** - Assess performance based on detection rate.

Lab Test Setup

- **Clear Line of Sight**
  - Tested link conditions of various drone distances
- **Shadowing/Fading**
  - Physical environment obstructions with some line of sight knowledge
- **No Line of Sight**
  - Behind building and out of line of sight

Lab Results (Mavic 2)

Number of Samples = 5000 (Random 20% of 25k)					
Distance (m)	Detection Rate (%) (True Positive + True Negative)/5000	True Positive	False Positive	True Negative	False Negative
5	99.98	2519	0	2480	1
25	99.84	2514	5	2478	3
50	99.34	2501	18	2466	15
100	99.46	2502	17	2471	20
125	98.82	2483	36	2458	23
150	95.34	2311	124	2456	109

Number of Samples = 5000 (Random 20% of 25k)					
Distance (m)	Detection Rate (%)	True Positive	False Positive	True Negative	False Negative
Shadowing/Fading	88.57	2134	243	2294	329

Number of Samples = 5000 (Random 20% of 25k)					
Distance (m)	Detection Rate (%)	True Positive	False Positive	True Negative	False Negative
Unkown Location	81.22	1987	563	2074	376

Site 1 Purpose

- **Data Acquisition** - Capture real data in unique environment
- **Model Building** - Developing models for each drone family within environment
- **Model Comparison** – Assess isolated model performance vs. lab created model. Analyze impact merging dataset into one model.
- **Detection Range** – Test maximum detection range of system

Site 1 Test Setup

- **Clear Line of Sight**
  - Tested link conditions of various drone distances
- **Shadowing/Fading**
  - Physical environment obstructions with some line of sight knowledge

Site 1 Results (Phantom 4 V2)

Distance (m)	Detection Rate (%)	True Positive	False Positive	True Negative	False Negative
100	99.98	10040	0	9956	4
200	99.34	10156	43	9712	89
500	98.77	10012	111	9745	132
600	93.14	9842	435	9456	267
700	91.88	8859	740	9767	632
800	87.20	9069	1254	8371	1306
900	81.09	8271	1967	7947	1815
1000	74.67	7915	2685	7019	2381

Number of Samples = 20000 (Random 20% of 100k)					
Distance (m)	Detection Rate (%)	True Positive	False Positive	True Negative	False Negative
Shadowing/Fading	64.58	6845	3755	6071	3329

- Detection Rate = 1-(FalsePositve (Class1) + FalseNegative (Class2))*100
- This calculation is the accurate detection rate of identifying both class 1 and class 2. Both true cases listed above

Site 1 Results (Phantom 4 V2) Combined Model

Distance (m)	Detection Rate (%)	True Positive	False Positive	True Negative	False Negative
100	99.98	10398	2	9598	2
200	99.56	10155	46	9757	42
500	99.12	10507	93	9317	83
600	96.34	10212	388	9056	344
700	93.04	9676	682	8932	710
800	90.33	9394	948	8672	986
900	86.29	8802	1426	8456	1316
1000	83.44	8845	1755	7843	1557

Number of Samples = 20000 (Random 20% of 100k)

Distance (m)	Detection Rate (%)	True Positive	False Positive	True Negative	False Negative
Shadowing/Fading	75.88	2557	7133	2267	8043

- Detection Rate = 1-(FalsePositve (Class1) + FalseNegative (Class2))*100
- This calculation is the accurate detection rate of identifying both class 1 and class 2. Both true cases listed above

Analysis

- **Approach extends detection range to roughly 1Km**
- **Largest training data set size with multiple model performance metrics**
- **Combined model resulted in ~3.7% increase in performance vs. isolated model bringing average accuracy over 90%**

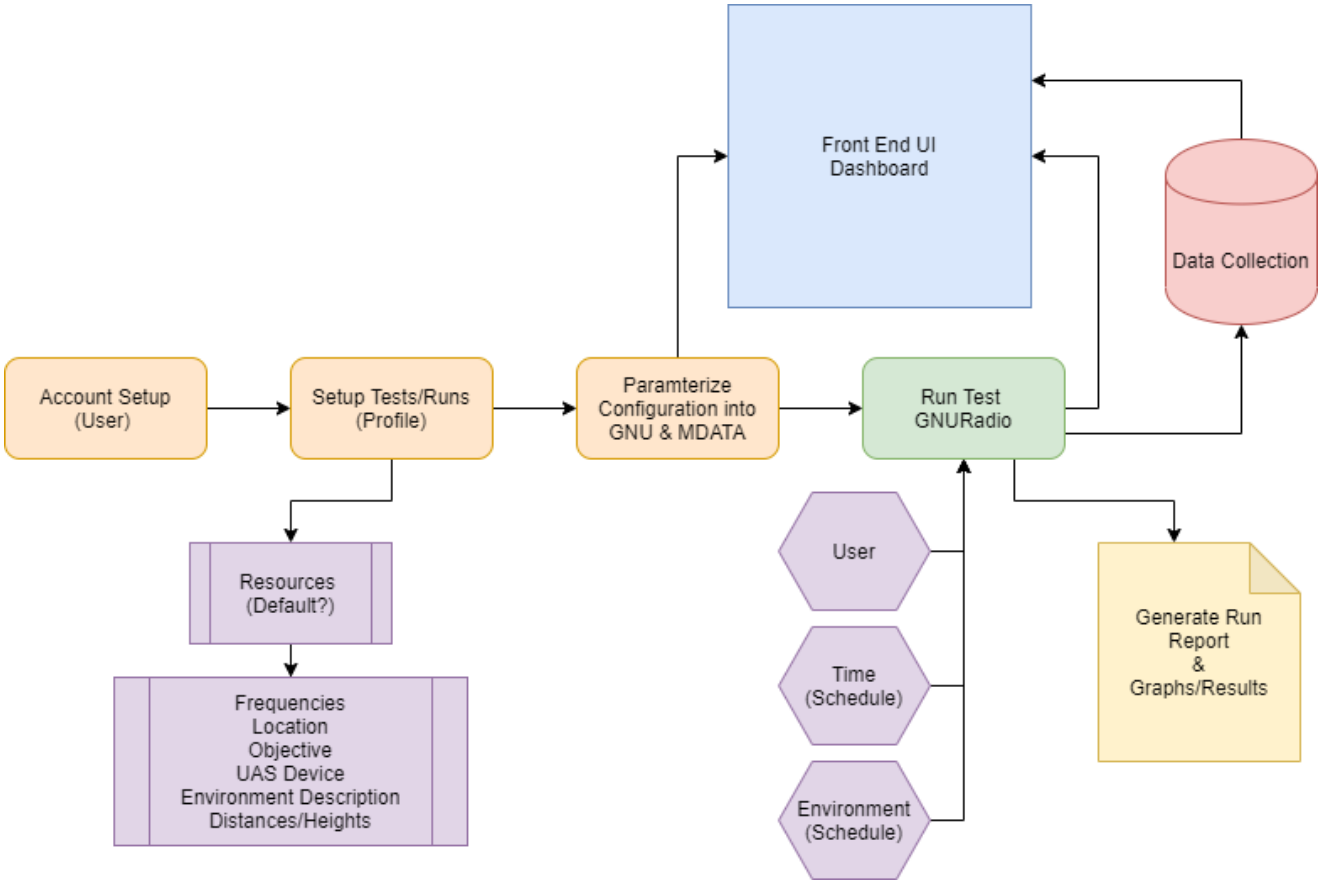
ML Parameter Data Processing

- Processing of seven current ML (Machine Learning) Model parameters specifically looking for drone class specific correlations
- Addition of environmental thresholding max value to capture full signal amplitude of drone
- Include spectral correlation function output in ML model feature input
- Expand classification model to include multi-drone classification steps

System Portal/Workflow

- User – Engineers or Testers responsible for conducting on site experiments
- Capabilities (under development)
  - Provide experiment profile
  - Customize experiment profile specific to system
  - Manage experiment execution
  - Manage Results and Data Files
  - Reuse Models
  - Schedule experiments (Batch)

System Portal/Workflow

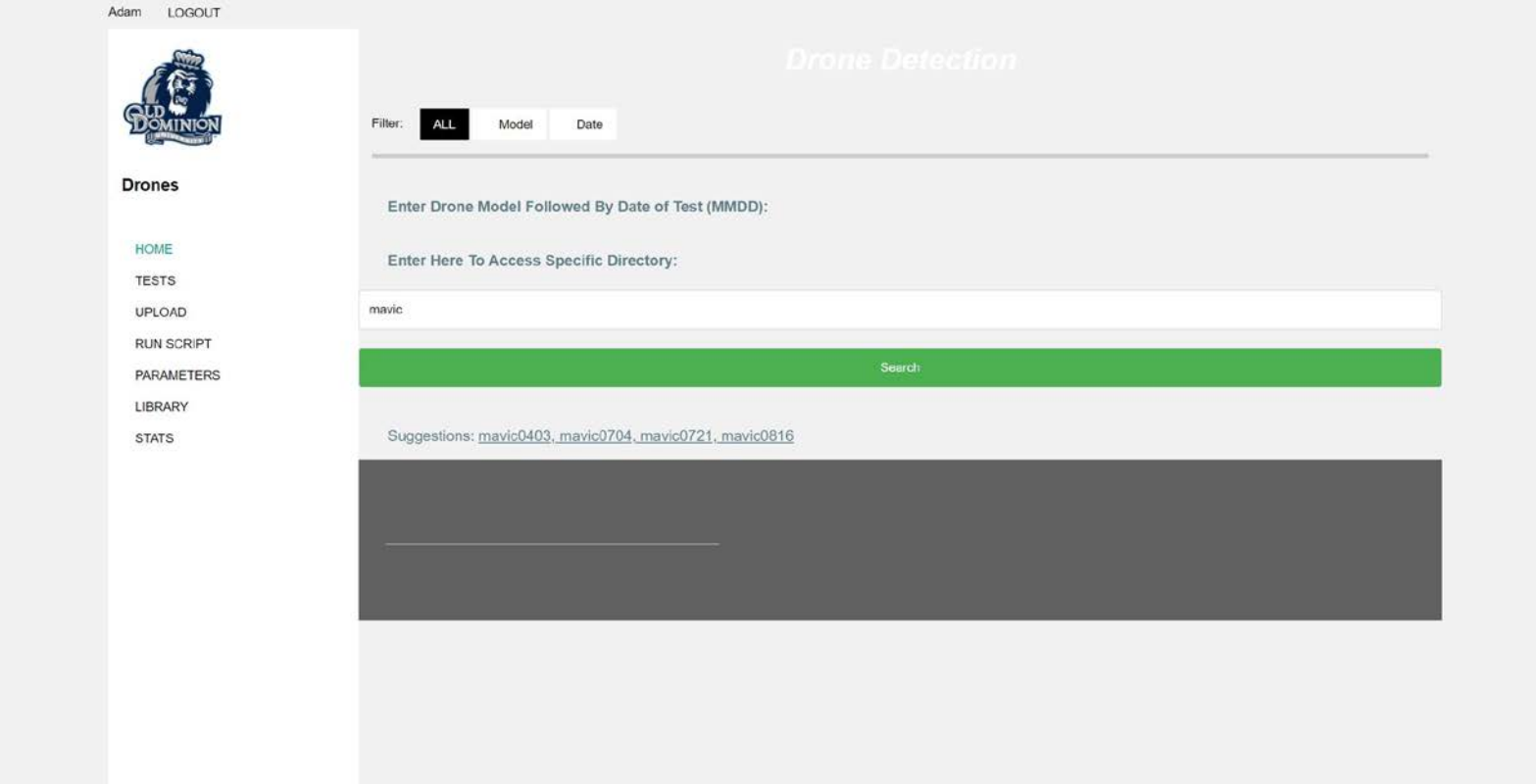


System (Back End) & Front End

- Include all data from trials with the ability to sort by different included system meta data
- Cluster of trials used in certain model creation (ex. trials 1-10 of 50 used for model creation)
- Ability to start model creation or visualization from front end (GUI)
- Automatic data generation, processing, and analysis
- Backend data storage and signal processing to be shown in the front end software

Front End Development

- **Example webpage front end in that is being currently developed**



Conclusion

- Provided a ML based RF classification platform for UAS detection
- Provides an agnostic approach to detecting and identifying presence of UAS in several SNR regimes.

- Aid operators in eliminating non-UAS signals in physical space and focus their detection in physical spaces where UAS signals have been detected.

- Improve the efficiency of detection as prior knowledge of RF fingerprints will aid in ensuring every detection will only focus on unknown RF signal detection.
- Help operators to speed up the detection and identification of UAS devices.

References

[1] M. Hamid, N. Bjorsell, and S. Ben Slimane, “Sample covariance matrix eigenvalues based blind SNR estimation,” IEEE International Instrumentation and Measurement Technology Conference Proceedings, 2014, pp. 718–722.

[2] J. Zhu, Z. Xu, F. Wang, B. Huang, and B. Zhang, “Double Threshold Energy Detection of Cooperative Spectrum Sensing in Cognitive Radio,” International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2008, pp. 1–5

[3] A. Quadri, M. R. Manesh, and N. Kaabouch, "Performance Comparison of Evolutionary Algorithms for Noise Cancellation in Cognitive Radio Systems," The IEEE Annual Computing and Communication Workshop and Conference,pp. 1-6, 2017.

[4] Wikipedia contributors. (2021, February 21). K-nearest neighbors algorithm. In Wikipedia, The Free Encyclopedia.