Signal Compression for Machine Learning Applications using RF Data

Isaac Rodriguez-Jimenez, *Student, Sonoma State University*

***Abstract*—The curse of dimensionality has plagued machine learning developers for decades. How much data is enough? Is more really better? The Hughes phenomenon [3] states that there is an optimal number of features that will maximize performance. Too much or not enough features will affect performance negatively. In the case of radio frequency recordings, each data point can be extremely large. Not having the proper computational resources can delay small scale organizations. I propose signal compression to solve this issue. I will compare two methods to try to capture the soul of a signal into a few features. Doing this will cut down machine learning training time into seconds and will not place additional strain on your integrated development environment. This paper discusses the following methods: (i) 13 key features/characteristics/aspects of an RF signal. Some of these key features are signal to noise ratio, mean, peak value, mean frequency, and band power. This approach received an accuracy score of 79%. (ii) 188 features using linear frequency cepstral coefficients. In this case, we take an RF signal and decompose it into 10 individual signals and consolidate them into 1. This method received an accuracy score of 97%. These approaches allow us to investigate how well a machine learning model can adapt to a condensed dataset. I setup a grid search to identify which machine learning models perform the best on the new data without hyper parameter tuning. Finally, the dataset used in this study comes from the Institute for Wireless Internet of Things, Electrical and Computer Engineering Department, Northeastern University, Boston, MA, USA [1]. https://genesys-lab.org/hovering-uavs**

***Index Terms*—UAV, RF Fingerprinting, Linear Frequency Cepstrum Coefficients, Mel Linear Cepstrum Coefficients, Fourier Transform, Feature Engineering**

I. INTRODUCTION

DATABASES around the world are constantly bombarded with storing large amounts of data that may or may not be useful. According to [2], 90% of the world's data has been generated in the last 2 years. Not only that, but every 2 years it has been found that the world's data doubles in volume. As of Aug 2023, the world's data is expected to be near 64 zettabytes, not including physical documents. Two solutions come to mind, we either develop new techniques for mass storage or we develop new techniques for compressing data. Many researchers are working on developing new ways to create larger storage devices, but what about storing the data more efficiently. There are many different types of data formats, it would be difficult for researchers to individually investigate the best way to compress a file for every type of data. Current compression applications have the ability to compress and decompress a file. The methods presented in this study are destructive and data cannot be recovered. The purpose of this study is to investigate if we can shed unnecessary data from a signal and retain key characteristics for machine learning applications. The resulting compression brings the original dataset from 10GB down to (i) 1.2 MB and (ii) 8.1 MB. Although the performance can vary from dataset to dataset, machine learning models may need different amounts of data based on the application and type of data recorded.

The author is part of the graduate program researching RFML practical applications, Electrical Engineering Department Sonoma State University, Rohnert Park, CA, USA.

Corresponding author: Isaac Rodriguez-Jimenez, email: rodriisa@sonoma.edu

In this case, the dataset used represents a collection of signals recorded from 7 unmanned aerial vehicles (UAVs). The signals recorded are the ones transmitted from the UAV to a receiver. This is important because we want to individually classify which UAV is transmitting a signal. This is called RF fingerprinting. The use of machine learning techniques to identify a device that transmits RF signals based on signal characteristics. These characteristics may stem from nonlinearities in the transmitting circuits. In order for the machine learning algorithm to properly classify 7 extremely similar devices, we need a proper balance of data points and features. This data set has approximately 13,000 signal recordings. Each recording is 97,000 samples long coming in at 0.8 MB.

Problem The dataset is too large for my computer to handle. This begs the question, is 97,000 samples really needed? The researchers [1] found great results working with the raw data applying preprocessing and data augmentation techniques before using a convolutional neural network. In my case, I do not have the processing power to clean up the data and create additional synthetic data. There are big data services like Hadoop, Azure, and AWS in order to distribute the processing and automate data pipelines. The issue with these services is that it does not encourage data compression and adds to the overall problem of having too much data stored in the world.

II. RELATED WORK

The following works are examples of researchers extracting features from transmitted signals. The techniques used by [4], utilized I/Q data. They propose that it is possible to distinguish between radio transmitters solely based on I/Q imbalance of quadrature modulation. They focus on the phase shift keying (PSK) of the signal and compare them to others. Some

transmitters use different PSK but for the ones that use the same scheme, the suggest it is possible to differentiate between different transmitters based on small variations on the constellation plot.

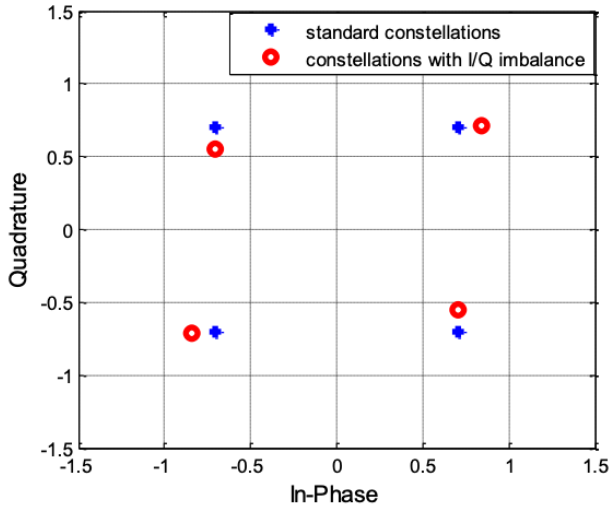

Fig. 1. Constellation plot when I/Q imbalance is present.

Although this technique only works for digital modulation, the researchers proposed a method that can also be used in analog modulation.

The following research [5] uses the time-based signal in order to extract features. The idea is that different devices have different energy envelopes of the instantaneous transient signal. This will allow them to uniquely identify the manufacture of the device. The main goal of this research is to improve the security of wireless networks.

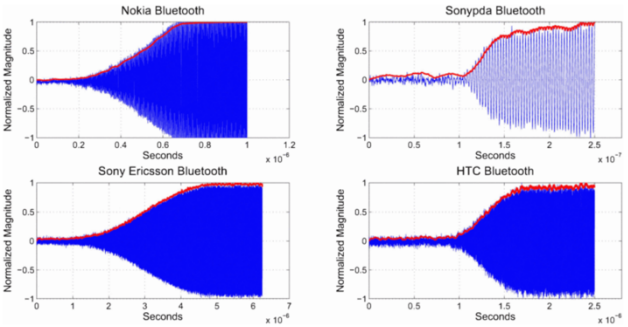

Fig. 2. Energy Envelopes for Different Devices.

Using the red signal, they extracted 6 key features.

- Area under the normalized curve
- Duration
- Maximum slope
- Kurtosis
- Skewness
- Variance

These features gave them great results. My work explores 2 methods. One is extracting features directly from the RF signal instead of the envelope. The second is taking the RF signal and decomposing it to smaller signals and taking my features from there.

III. Feature Engineering

The dataset we are using from [1] is formatted as a sigMF file. Each signal is represented as an array of complex numbers. The meta files contain the necessary data to revert the complex data into real RF signals. Finally, further processing is necessary to save the data onto MATLAB's workspace.

(i) My first method explores extracting features straight from the RF signal. I extracted 13 characteristic traits, some from the time domain and some from the frequency domain.

TABLE I
Features Extracted

Time Domain	Frequency Domain
Mean	Mean Frequency
Shape Factor	Median Frequency
Crest Factor	Occupied Bandwidth
Clearance Factor	Power Bandwidth
Signal to Noise Ratio	Band Power
Stand Deviation	
Peak Value	
Impulse Factor	

The purpose is to see if we can individually identify which UAV is transmitting based on the actual transmitted signal. MATLAB makes it easy to extract these features with a function called, "signalTimeFeatureExtractor" and "signalFrequencyFeatureExtractor". These features are quickly and easily extracted in order for us to have a baseline. After extracting these features, we have reduced our original dataset to 13 points per data point. The original dataset is 10GB while this new dataset is 1.2 MB. I don't believe it is possible to accurately depict a 100,000-point signal into 13 values, but we should try to see what kind of results we can get with the best performing machine learning model for this data. Using MATLAB's "Classification Learner", I was able to try multiple models at once with one dataset. The model chosen by MATLAB was a bagged decision tree which came out with an accuracy of 79% for 7 classes with a training time of 69 seconds.

(ii) This next method decomposes the signal before extracting features. The Linear Frequency Cepstrum (LFC) was a common technique used in language processing before deep learning really took off. It is the same process as the Mel Frequency Cepstrum (MFC). The MFC typically focuses on the audio spectrum with an emphasis on the lower audible frequencies. The LFC does not have a bias over the spectrum. Comparing figure 3 and figure 4 you can see the filter banks for each method. The MFC has smaller triangular bandpass

filters near the minimum frequency and larger triangles near the maximum frequency. This means more data is taken near the bottom and less at the top. The LFC on figure 4 has equal sized triangles throughout the spectrum. The dimensionality reduction comes from the number of filter bands in the bank.

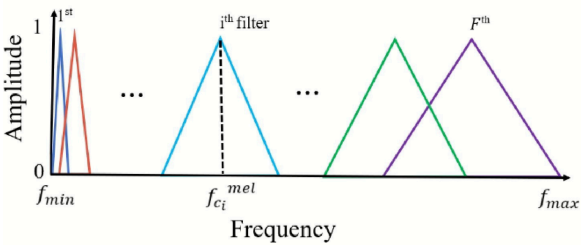

Fig. 3. Mel Frequency Filter Bank

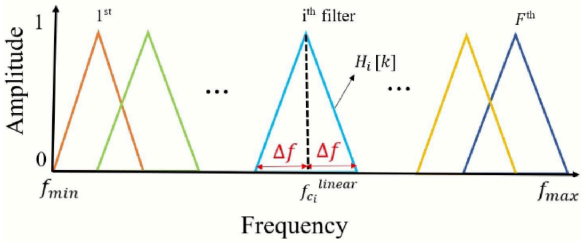

Fig. 4. Linear Frequency Filter Bank

The process involves capturing the log of the power spectrum using the signal's Fast Fourier Transform (FFT). Then in an effort to decorrelate the signals coefficients a discrete cosine transform (DCT) is applied. In essence the filter banks are applied to the signal's spectrogram in order to smooth out the fine details for better generalization across similar signals. Finally applying the discrete cosine transform introduces information from the spectral envelop leaving the resulting spectrogram visually different but computationally efficient.

I took the inspiration to use the LFC from [6]. These researchers we are able to use coefficients derived from the LFC and used them in their machine learning model to accurately classify UAVs from different manufactures.

I used a built in MATLAB function that extracts the LFC coefficients. The way this function works, is by taking the original signal and breaking it up into segments called windows. These windows usually overlap across the spectrum and thus are weighted. Common weighted windows taper on the ends such as a Hamming or Blackman. The fourier transform are applied to each window (spectrogram). Next the triangular bandpass filters are applied across all windows. Following this we take the log of the values and apply the Fourier transform again. This last Fourier transform is called a Discrete Cosine Transform (DCT). This function represents values as a summation of cosine functions. Using the linear frequency cepstrum coefficients, I take one 97,000-point signal and break it down to 10 individual signals with 188 points each. Each signal is considered a coefficient. Figure 5 shows 10 linear frequency coefficients derived for 1 signal.

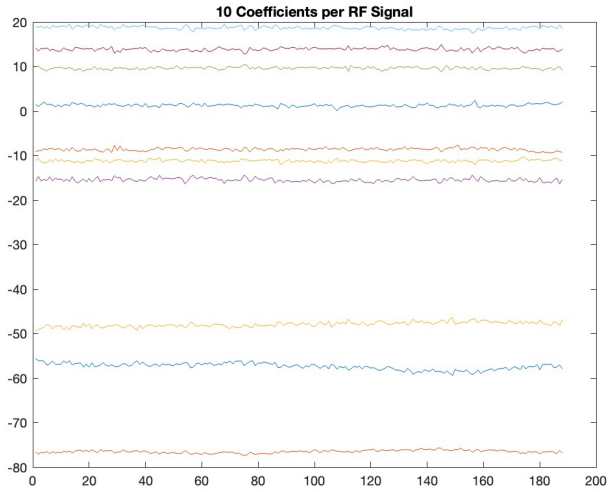

Fig. 5. LFC Coefficients for 1 RF Signal.

Researchers from [6] fed these signals into their machine learning model. In my case, I want to further reduce the amount of data saved per signal. I came up with the idea of consolidating all the coefficients into one signal. Each signal was added and subtracted in an alternating fashion using the equation below:

$$\sum_{n=1}^{10}(-1)^{1-n}coef(n)$$

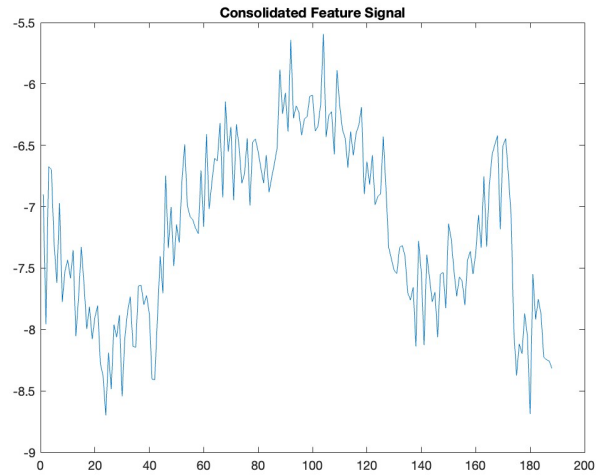

Fig. 6. 10 LFC Coefficients Consolidated into 1 Signal.

Different methods for combining the coefficients were experimented but this method gave us the most variance between UAV signals. The expectation is that each of these consolidated signals will be different enough for each individual UAV that a machine learning model can pick up on their specific transmitting characteristics. Figure 7 shows how each UAV can be visually segmented based on its consolidated coefficients. Using this method, reveals that each signal the UAVs transmit follow a similar pattern.

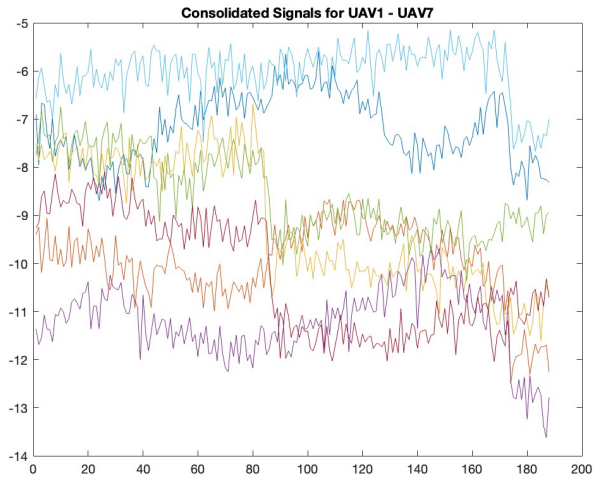

Fig. 7. Consolidated Signals for each UAV using 10 Coefficients.

IV. Results

I let MATLAB decide which classification model would work best for this data and a Quadratic Discriminant was chosen. The accuracy was 97.7% along with a training time 20.47 seconds. The original dataset was reduced from 10GB to 8.1MB with no impact on system performance. For this scenario MATLAB is a great starting point in determining which model investigate. Since the quadratic discriminant model essentially only has 1 hyper parameter that affects the regularization. Data augmentation might be necessary to increase the model's performance.

Comparing our results for methods 1 and 2, we can see the confusion matrices on figure 8 and figure 9. The confusion matrix shows how well a model predicts. For UAVs 1 – 7 the vertical axis is the truth class, and the horizontal is the predicted class. Looking at figure 8, truth class line 3 has a blue box and a beige box. This means that for UAV 3 the model correctly predicted 1248 times UAV 3 and 105 times wrong (UAV 4). Comparing both confusion matrices, the have essentially the same layout. Method 2 was able to remove some beige boxes from method 1 and lessen the ambiguity between UAVs. There are several tactics that can be used to further increase the prediction rate. Additional transformations can be made in order to decrease the data's entropy and increase the generalization between transmitting circuits. Another tactic is increasing the size of the dataset by augmenting the coefficients. In order to produce synthetic data, we need to check closely to see if we do not introduce highly correlated values or unwanted bias. Lastly, changing the number of features has a large impact on how well the model performs. Going from 13 to 188 features made an impact on how well a model was able to discern between the UAVs. Looking at the curse of dimensionality, for this dataset it is possible to say that I am near to the ideal number of features, but we also need to examine the effectiveness of the transformation (LFC). There are many different methods that can be used to extract features at various lengths. Some may be more effective than others at predicting. Selecting a minimum set of importance features can be done to see exactly how many features can best represent these signals in a machine learning classification application.

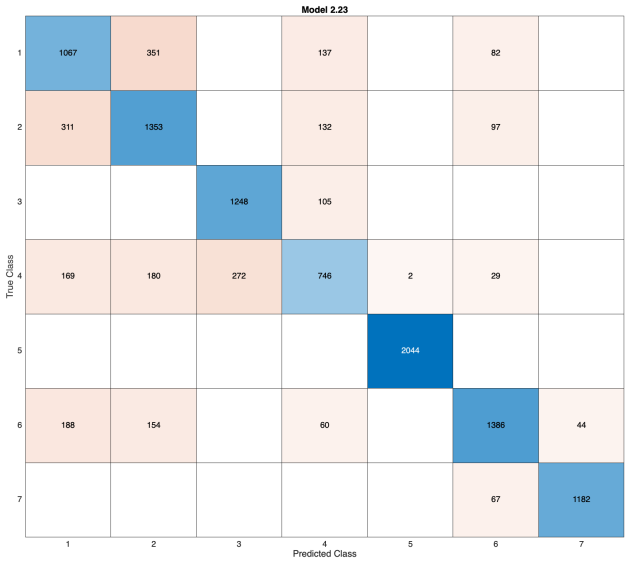

Fig. 8. Confusion Matrix for 13 Features Taken from RF Signal

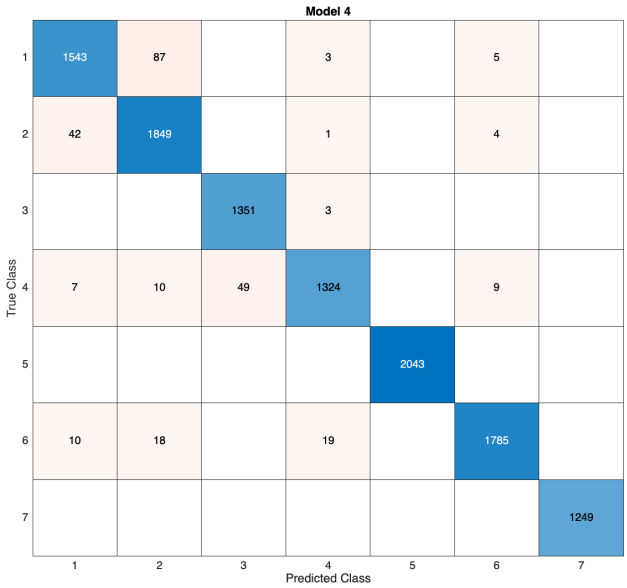

Fig. 9. Confusion Matrix from LFC Coefficients.

V. Conclusion

This study was performed in order to investigate the minimum number of features needed to properly train a machine learning model. This in turn reduces the amount of storage the dataset needs. This is useful for training models on computers that are not powerful enough to support the full size of the dataset. Simple data manipulation tasks may take longer than expected causing one to underestimate the time required to complete a task. The last thing to consider is file storage and transfer. Using smaller files allows developers to share data and store more effectively. Storing duplicate training data will have less of an impact than having duplicates of the

original dataset. Overall, it is uncertain what combinations of features, transformations and quantity of data are needed to accurately represent a RF signal for a classification model, but it can be said that the original signal is not necessary.

REFERENCES

[1] Nasim Soltani, Guillem Reus-Muns, Batool Salehi, Jennifer Dy, Stratis Ioannidis, and Kaushik Chowdhury, "*RF Fingerprinting Unmanned Aerial Vehicles with Non-standard Transmitter Waveforms*," *IEEE Transactions on Vehicular Technology*, Nov. 2020.

[2] K. Bart. "How much data is there in the world?" Rivery. https://rivery.io/blog/big-data-statistics-how-much-data-is-there-in-the-world/ (accessed Dec. 14, 2023).

[3] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, January 1968, doi: 10.1109/TIT.1968.1054102

[4] Fei Zhuo, Yuanling Huang, Jian Chen, "*Radio Frequency Fingerprint Extraction of Radio Emitter Based on I/Q Imbalance*," Procedia Computer Science, Volume 107, 2017, Pages 472-477, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2017.03.092.

[5] S. Ur Rehman, K. Sowerby and C. Coghill, "*RF fingerprint extraction from the energy envelope of an instantaneous transient signal*," 2012 Australian Communications Theory Workshop (AusCTW), Wellington, New Zealand, 2012, pp. 90-95, doi: 10.1109/AusCTW.2012.6164912.

[6] Rabiye Kılıç, Nida Kumbasar, Emin Argun Oral, Ibrahim Yucel Ozbek, "*Drone classification using RF signal based spectral features*," Engineering Science and Technology, an International Journal, Volume 28, 2022, 101028, ISSN 2215-0986, https://doi.org/10.1016/j.jestch.2021.06.008.