Radar Emitter Recognition using Hierarchical Feature Extraction within Magnitude and Frequency Domains

By

Robert A. Newport

BS California State Polytechnic University, Pomona
MSE Carnegie Mellon University

A thesis submitted to Macquarie University
for the degree of Masters of Research
Department of Computing
July 2019

MACQUARIE
University
SYDNEY · AUSTRALIA

Examiner's Copy

Typeset in LaTeX 2_ε.

Statement of Originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

(Signed) ______________________ Date: July 29, 2019

Robert A. Newport

Acknowledgements

Firstly, I would like to thank my family for their encouragement through my studies, especially my mother who provided support for me despite her own challenges recovering from breast cancer.

I would like to thank Defence Science and Technology Group, Adelaide, specifically Mr Mark Cooke for spectrogram data used in experiments and feedback during key phases of research, and Mr Ross Kyprianou for his advice and feedback through much of the work. Their role as mentor and advisor helped keep the thesis on track.

I would like to thank John C. Wise with Radars UK [1] for access to a sample of 100 radar emitter Pulse Descriptor Words. This data was critical during the research of traditional radar emitter recognition techniques due to the scarcity of public information online.

Finally, I would like to thank my supervisor, Professor Len Hamey, for his encouragement during my application, and guidance throughout the preparation of this thesis. His unwavering encouragement and superb knowledge of signal processing was a beacon through the fog of radar emitter recognition material.

[1] Website: http://www.radars.org.uk

Abstract

Radar Emitter Recognition (RER) is used in Electronic Warfare (EW) to avoid being detected during stealth operations, or to detect specific radar installations for targeting in an offensive campaign. Its role in both offensive and defensive measures make it a critical component within the scope of a larger Electronic Countermeasure (ECM) strategy. Radar spectrograms are captured from emissions, measuring frequency and power over time. The emission samples vary in the amount of time spent in each frequency, making a single approach to feature extraction ineffective. This thesis attempts to address RER challenges using modified radar signals from spectrograms sampled from five different classes, each containing 250 unique examples. A hierarchical approach is used where spectrograms with fewer time intervals have their features extracted from the magnitude spectrum, while signals with more time intervals have features extracted from the frequency domain. Both of these feature extraction methods are tested using k-Nearest Neighbor (kNN) and Support Vector Machine (SVM) algorithms. Experiments show that the hierarchical approach to feature extraction is a viable new way of thinking about spectrogram based RER.

Contents

List of Figures

List of Tables

"All warfare is based on deception. Hence, when we are able to attack, we must seem unable; when using our forces, we must appear inactive; when we are near, we must make the enemy believe we are far away; when far away, we must make him believe we are near."

Sun Tzu, The Art of War

1 Introduction

The basic operation of radar is to send an electromagnetic pulse from an emitter and measure the time it takes for it to return to a receiver. The time delay can be used in calculations to determine the distance and trajectory of a distant object. Radar systems can detect weather patterns, vehicle speeds, aircraft locations, and many other things depending on their function and purpose[3]. In a military setting, radar emitter detection maintains a critical role in battlefield intelligence[4].

1.1 Radar Emitter Recognition Beginnings

In the field of Electronic Warfare (EW), Radar Emitter Recognition (RER) can prevent a radar guided surface-to-air missile (SAM) from reaching its target[5]. Not only is recognition critical for identifying radar emitters for battlefield intelligence, but also for the jamming of these signals for stealth[4]. These *cat-and-mouse* style games were primarily developed during, and after, America's Vietnam War following an attack by Soviet-made Surface-to-Air SA-2 missiles that shot down a US$5.5 million RA-5C reconnaissance aircraft[6]. This event motivated staff[1] to head a Blue Ribbon Air Staff Task Force in charge of developing what was to evolve into Electronic Countermeasures (ECM) and Electronic Counter-Countermeasures (ECCM), for both the avoidance of radar detection through trickery, and the avoidance of the trickery itself. The first attempt at real-time battle-ready RER systems was seen in project Wild Weasel[6] where military jets tracked the polar trajectory of incoming surface-to-air missiles and drew rocket fire while other jets could conduct their bombing runs. The few moments of advanced warning these early radar detectors provided were often enough to allow pilots to deploy reflective metallic chaff in order to create deceptive radar returns, or to fly evasive maneuvers[3].

1.2 Traditional Features in Early Radar Detection

As radar technology advanced through the 1970s, a more diverse range of emitter signals launched a new standard for categorizing radar features. Traditionally, automatic RER uses tabulated Pulse Descriptor Words (PDW) to train a classifier to identify signals using a handful of the emitter's signal features, such as its pulse width and amplitude[4]. PDWs are the unique characteristics of different emitter signals that are extracted manually by a radar intelligence operator or, more contemporarily, done automatically through an algorithm[4]. This classification process would require a catalog of PDWs and the ability to use attributes to exclude candidates until a small set of possible emitters are left[4]. In-flight systems for automatic emitter detection during field operations have also been developed, followed concurrently with electrical engineering and software development. Accuracy was susceptible to signal to noise ratios more challenging than 10 dB, as shown by the scarcity of research tests reported beyond that value[7].

In contemporary research, identification involves training a machine learning algorithm to classify features extracted from the sampled waveform. This is a challenge due to emission complexity; therefore, generalization of the data is required to provide features for training that do not succumb to *data sparsity*, which is the loss of statistical relevance due to signal data

[1] Director of Operational Requirements and Development Plans, U.S. Air Force Headquarters Brigadier General K.C. Dempster

being spread thin over a large dimensional space[8]; defined by the number of time samples and frequency bins. Utilizing the entire sample instead of just PDW characteristics of an emitter allows for a customized and novel approach to feature extraction, with the possibility of greater noise resilience.

Moving away from PDWs as features, intrapulse characteristics of waveforms are being scrutinized using fractals and entropy based calculations to extract energy characteristics of specific waveforms for increased accuracy in prediction[9, 2]. New research is leaving PDWs behind in favor of pursuing the opportunity for novel approaches to emitter recognition using various techniques for generalizing an emitter's signal[10]. However, since so much of the initial research was done using established PDW features for developing classifiers in lieu of exploring new features, there exists a lot of opportunities for novel feature extraction techniques while still using traditional classifiers. This will leverage established knowledge while providing a fresh approach to prediction and training that could yield greater accuracy and noise resistance through the development of new feature extraction methods.

1.3 Extracting Features from Spectrogram Samples

At its most simple, a spectrogram is a measure of how frequencies vary in strength over time[11]. Radar emissions vary greatly in frequency and power over time, with different time intervals revealing characteristic changes in frequency and time. The aim of this thesis is to develop feature extraction from radar spectrograms to provide a classifier agnostic process for signal prediction. The constructed radar samples are generated from a database of 250 in-phase and quadrature-phase (IQ) data fed through an algorithm, modifying the signal to add attributes such as noise. The IQ data show the changes in the phase and amplitude of a radar signal, in order to capture unique signatures in a radar waveform. The phase and amplitude changes parsed from sample attributes make up the modulation profile, which is assembled into a spectrogram. This thesis proposes waveform features as a replacement of PDW characteristics for the purpose of prediction using an arbitrary classifier.

The scope of this study focuses primarily on spectrogram data from five different signal classes: Continuous Wave (CW), Frequency Modulation (FM) Linear Up, FM Asymmetric Up, FM Symmetric

Up, and FM Triangle Up. The signals contain varying numbers of time intervals, different degrees of detail over the frequency excursion, and samples with noise within each cohort. The innovative concept introduced in this thesis is a hierarchical approach to features extraction for samples with long and short time intervals. Shorter intervals revealing greater changes over the frequency spectrum will have features extracted from a graph of the signal power over frequency, whereas samples with longer time intervals will have features extracted from a graph of frequency over time. The purpose of this approach is to increase the separability of the features, increasing overall accuracy using any modern classifier. Testing will implement Euclidean distance K-Nearest Neighbor (kNN) and linear Support Vector Machine (SVM) classifiers using features created by the methods described in this thesis. With an understanding of how this hierarchical approach increases waveform separability, future work can continue improving the methods used to extract features, making this study an extensible platform for continuing research.

1.4 Overview

The aim of exploring classifier agnostic feature extraction methods will be accomplished through the following high level steps: exploratory data analysis of the simulated radar samples, examination of waveform characteristics at both long and short time intervals, and interpolation methods for optimizing dimensionality of frequency and magnitude subsamples. The thesis starts with a Chapter 1 Introduction, followed by a Chapter 2 Literature Review, covering contemporary research using non-traditional feature extraction methods, providing a context for where this research fits within the field. This should prepare the reader for Chapter 3 Methods, introducing the spectrogram data, with an exploration of axes, parameters, and plots. The methods used to extract features from the samples are also described here. Chapter 4 Results explores how the extracted features can be used with kNN and SVM classifiers with an analysis of classification accuracy. A confusion matrix provides insight into limitations and strengths for each feature model, which provides data for discussion within the chapter. Finally, Chapter 5 Conclusion outlines how well the study has achieved its aim, limitations, and the opportunity for future work.

"Be extremely subtle even to the point of formlessness. Be extremely mysterious even to the point of soundlessness. Thereby you can be the director of the opponent's fate."

Sun Tzu, The Art of War

2

Literature Review

The increasing penetration of computers and software systems in the late 1960s and early 1970s provided a capability to quickly compute probabilities for matches from a database of characteristics, provided those features were organized uniformly for database storage[6]. In 1972 the ALR-45, based on a CPU and software architecture model implementation, was able to execute probability statements[6] based on Pulse Repetition Interval, Pulse Coding, Frequency, Pulse Width, and other emitter-specific attributes. This system worked exceptionally well, and lead to a 1976 patent[12] explicitly outlining the importance of "digital words" to simplify the processing of emitter recognition by organizing various Radar attributes into comparable waveform characteristics. As the pursuit of radar emitter recognition advanced, the digital words described in this patent become known as Pulse Descriptor Words in the Electronic Warfare research field[4].

2.1 Traditional Features in Early Emitter Recognition

The increased use of descriptors can be seen as a product of the technology used to exploit them. Increased storage requirements for descriptor sample data of radar emitters made accurate automatic recognition an increasing possibility if clusters of those features could be compared in a statistical manner on a computer. The jump into CPU and software based Radar devices such as the ALR-45 were a bridge that compelled Applied Technology to cross from a circuit design company to a computer sciences company[6]. The use of a descriptor format illustrates how Radar Emitter Recognition technology could fit into an emerging software framework by leveraging an efficient data structure to store and retrieve features.

2.1.1 Pulse Descriptor Words in Detail

In the same way that buildings, mountains, tanks, and aircraft carriers reflect light, they also reflect radio waves. Like light, a portion of radio waves emitted from a Radar transmitter are bounced back[3]. At its most rudimentary, a Radar receiver can calculate the length of time it takes for a round trip between transmission and reception to determine the distance a target is from the device. Different factors affect attributes of the Radar signal, including its range, resolution, and strength. These factors could include transmission power, transmission duration, antenna size, number of search scans of the area, wavelength, and different types of pulse compression[3].

Traditional methods used in *a priori* database filtering for identifying Radar signals include the following five classic parameters[13]: Pulse Width (PW), Direction of Arrival (DOA), Time of Arrival (TOA), Pulse Amplitude (PA), and Radio Frequency (RF). These parameters operate in Electronic Warfare systems at 500k to 1M pulses per second[14] making parsing computationally expensive. A traditional format used to describe an emitter is the Pulse Descriptor Word (PDW), which may contain some or all of these parameters[4]. Traditionally, these five classic parameters are packed into a PDW, an Electronic Warfare receiver sends them to a pulse-sort processor for Pulse Repetition Interval (PRI) identification and modulation isolation. The PRI is a calculation of $\frac{1}{PRF}$ where the PRF is the number of frequency pulses per time unit, normally measured as one second[4]. They are then compared to an existing emitter database[15]. Random modulations, produced by hardware amplifiers and power fluctuations, such as jitter, stagger, and switching aberrations, may interfere with the correlation of emitter data, causing final identification to generate an emitter list with a variable recognition score. To improve these scores for emitter recognition, researchers in the field use machine learning to increase the efficiency and accuracy of the Electronic Warfare recognition process.

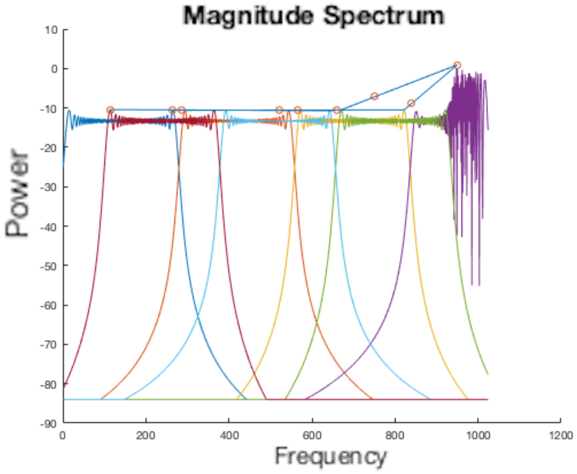
2.2 Beyond Traditional Features

As described in Section 2.1.1, Pulse Descriptor Words were a traditional format used in *a priori* filtering and classification of radar emitters during the inception of recognition research. However, with the dense battlefield of agile radar emitters able to artificially stagger and jitter pulse repetition intervals on a pulse by pulse basis, the need for more research is becoming increasingly urgent. This section introduces the reader to contemporary techniques in feature extraction for machine learning. Feature extraction can be generalized into the following three categories[7]: 1) the five classic PDW parameters matched with a priori waveform data, 2) waveforms as inputs for deep learning, 3) waveform intrapulse signals for feature analysis. The five classic PDW parameters in 1) were explained in Section 2.1.1 as a background to the RER field. A break from the reliance on the five traditional parameters by many of the contemporary research papers included in this study implies that research into individual radar features is becoming increasingly popular, especially since many traditional methods can barely meet the requirements of warfare with signal-to-noise performance not greater than than 10 dB[7]. Deep learning in 2) will not be covered in this thesis since the focus is narrowed down to how feature extraction performs using kNN and SVM, with classification performance being a gauge of feature extraction success rather than an end goal. However, this study will scrutinize waveform intrapulse signals, the final item 3) of the three categories, during the data exploration phase of this study.

2.2.1 Intrapulse Information

Modern radar emitters have the ability to artificially jitter and stagger consecutive Pulse Repetition Intervals (PRI). This ability portends that relying on the classic PDW parameters for features in emitter recognition will become increasingly inadequate. Researchers looking for other features

FIGURE 2.1: FM Triangle Up intrapulse frequency detail over the magnitude spectrum.

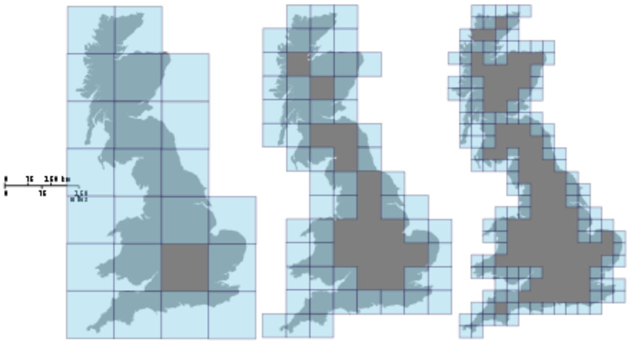


of emitter waveforms are currently exploring the details within the structure of each pulse, otherwise known as intrapulse information. Researchers, including Ming-Qiu Ren *et al's* kernel PCA analysis[16] are finding that the time-frequency structure of contemporary radar waveforms is where many distinctive features are hidden. This information could contain features hidden in the frequency domain, which may be revealed in the magnitude spectrum. Figure 2.1 shows intrapulse details over the frequency excursion separated into differently colored time intervals.

Intrapulse features can include waveform attributes such as the rise time and angle, time of slope and fall, angle of fall, and line of regression[17]. Reducing the feature set to eliminate uninformative and redundant features is highly specific to the waveform since many different types of signals will contain different significant intrapulse data. One way to encapsulate intrapulse features is to use quadratic time-frequency distributions (TFD) as is demonstrated by Ren *et al*[16]. In this study, a feature set is derived from time-frequency images to maximize extraclass variations and minimize intraclass ones. Then, using kernel principle component analysis (KPCA), features were obtained in low dimensional space in order to reduce computational load, and kept highly discriminative in order to maximize accuracy.

A different method to maximize accuracy while tolerating changes to signal to noise ratio extracts the Wigner Trispectrum (WT)[18] of a captured waveform, extracted and simplified into 2D features for the extraction of coefficients, for differentiation between emitter signals. Since the WT is not noise sensitive, signal to noise ratios between 10 dB and 5 dB have shown accuracy

FIGURE 2.2: Box Dimension used to solve the measurement of the British Coastline[1].



between 99.875% and 92.375% respectively, while signal to noise ratio above 15 dB resulted in 100% accuracy among the 8 simulated radar emitters[18].

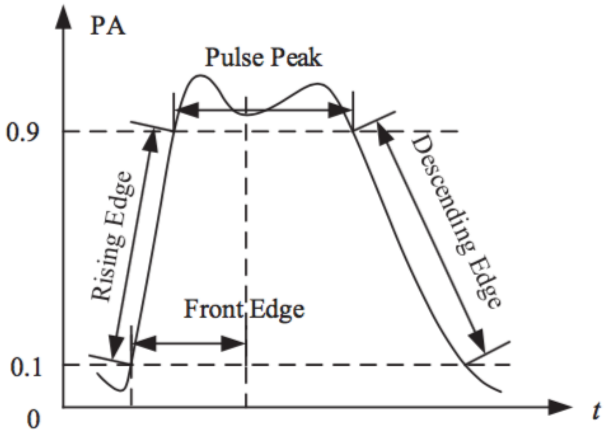
Intrapulse modulation analysis plays an interesting role in feature extraction for sparse representation based classification because the sparse approximation of the signal becomes its class label. This means that the number of features replaces the choice of features as being the critical factor in maintaining accuracy. Xie's research in *Robust intra-pulse modulation recognition via sparse representation*[19] demonstrated accuracy between 90% and 99% at a SNR between 6 dB and 8 dB using a dictionary learning method for representing each type of modulated pulse.

2.2.2 Fractal Dimensions

The fractal dimension of a signal describes the statistical index of the complexity of a waveform as the scale in which it is measured changes[20]. It is particularly well suited to quantitatively describe the distribution information of individual envelope waveforms[9]. By using this method, spurious modulations could be used as features using the Minkowski-Bouligand dimension, also known as the Box counting Dimension (BD), to classify individual signal envelopes.

BD has been used as a theoretical measure of the coast of Britain, as illustrated in Figure 2.2, solving the confounding problem of infinite distance as the measurement is scaled closer to the coastline[21]. Similarly, the scales of irregularity and complexity in the signal envelope are characterized using the BD as a time series. A unit square method is used to compute the BD in the study by Xu *et al*[9]; advanced computational power in the future will facilitate the implementation of more complex algorithms including the Kompasu method and the Brown model. Recognition

FIGURE 2.3: Novel features can be found in the front edge of a pulse envelope[2]



accuracy of 68.8% to 70.9% in the study showed higher accuracy rates compared to moments-based feature extraction[9]. Rates were slightly lower than the much more computationally expensive Square Integral Bispectra (SIB)[22] with 70.0% to 73.8%. These results demonstrate that fractal dimensions can be used in RER research.

As an alternative to BD, Correlation Dimension (CD) is much simpler and less computationally expensive and can be obtained from sampling the signals directly[23]. A Genetic Programming (GP) algorithm calculates the correlation dimension by first preprocessing radar emitter signals using the Fourier transform, normalizing signal energy, and then solving the normalization and center frequency of bandwidth. The process then involves resampling the signal in order to make all the signals the same length. This prevents signal length from altering correlation calculations. Finally, the signal is reconstructed in phase space[23].

In addition to the BD, the Information Dimension (ID) of the front edge of a pulse envelope can be a novel area for feature extraction. The front edge is used primarily because it is insensitive to multipath effects compared to other parts of the pulse envelope. Though, distortions in the envelope can also occur with noise and fading in the channel[2]. The definition of the *rising edge* is the intrapulse segment of the waveform where the pulse begins to rise from 10% to 90% of the ascending pulse amplitude, whereas the *front edge* includes both the *rising edge* and a portion of the *pulse peak segment* as shown in Figure 2.3. The three step process to isolating features using this method begins with preprocessing. This includes resampling of the short front edge waveform feature using an interpolation technique. These could include spline, linear, nearest neighbor, sinc, polynomial, or Gaussian interpolation. The purpose of the interpolation is to enhance the convergence of the fractal calculations since the front edge signal lengths are usually short. After adequate interpolation is applied to normalize the signals, a BD of the front edge's pulse envelope is calculated. The BD represents the geometric scale of the fractal waveform. Finally, the BD of the front edge of the pulse envelope is introduced to illustrate the fractal set's spatial distribution information. Using High Order Moments to sort signals, the front edge fractal feature extraction method was able to achieve 97.74% to 98.84% accuracy on simulated signals and 95.9% to 96.5% accuracy using hardware generated emitter signals[2].

2.2.3 Extents as features

Scale Invariant Feature Transform (SIFT) can be used to find interesting points in an image[24]; however, if the features are to be used in machine learning, it is important that they are in high contrast regions of an image, such as edges, so they can be detectable under changes in lighting and scale. SIFT can be used to extract features from preprocessed waveforms for the purpose of emitter recognition[25] by first obtaining a time-frequency analysis of the emitter signal. The preprocessed 3D time-frequency result is then normalized into a 2D grayscale image. Finally, SIFT position and scale features are extracted from a Gaussian difference pyramid of the grayscale image. Scale features can be used to suppress noise features. With the features successfully extracted, a Support Vector Machine, or any other machine learning process, can be used to automatically identify radiation sources based on SIFT position features. Recognition rates up to 0 dB signal to noise ratio are above 90% using SVM with SIFT based feature extraction[25], making this method particularly effective in noisy environments.

As an alternative to SIFT, the distance between the curves of multiple waveforms can be calculated using Frechet distance and used to measure the distinct features among different emitter waveforms. The distinctions are hidden in the time-domain structures of the instantaneous frequency and pulse envelope, both of which are features adopted in the research conducted by Chen *et al*[26]. After the Frechet distances for both instantaneous frequency and pulse envelope are calculated, they are compared to Hausdorff[27] and Dynamic Time Warping (DTW)[28] distance calculations for similarity. Finally, a minimal distance criterion makes the decision about the relationship between each class and the test feature vector. Tests using Frechet, Hausdorff, and DTW showed superior results for Frechet distance using 1000 fingerprint vectors with computational complexity being on par with the other distance algorithms.

Unlike the geometry dependent SIFT and Frechet methods, waveform distribution characteristics can be used to describe features in a waveform using entropy. Entropy is defined slightly differently in all the fields that use the term which originated from the German physicist Rudolf Clausius who proposed the idea in 1865. Used in emitter feature extraction, *Singular spectrum index entropy*, *Shannon entropy*, and *Singular spectrum Shannon entropy* features can be used to describe radar signal waveform distribution characteristics[29]. In statistical thermodynamics, entropy is defined as the degree of disorder in a system, with less disorder meaning less entropy[30]. Therefore, the entropy function can be "a function of the n-dimensional probability vector" when there are N probabilities that sum to 1, with each probability being between 0 and 1[29]. Changing the Fast Fourier Transform (FFT) algorithm allows straight forward switching between entropy types. Using a neural network classifier on four kinds of signals, 100% accuracy was achieved at 0 dB SNR with 84% at -5 dB and 66.5% at -10 dB.

2.3 Conclusion Summary of Literature

In current research, experiments are focused on a small set of modulation modes in low noise settings. Focusing on other modes like CW, NLFM, FDK, or PSK[9] and achieving higher accuracy with signal to noise ratio below 10 dB would be a natural next step, since most Radar Emitter Recognition methods cannot achieve adequate warfare level accuracy below 10 dB SNR[7]. In cases where noise is introduced into samples, many tests only include Gaussian white noise[31], which may imply that the methods contained within the experiment are not suitable for other types of noise. Current research appears to focus on basic features within the waveform, neglecting combined or unique features[32] within specific waveform types. Compared to the classical five parameters for a piori waveform table prediction, analysis has shown that features are not a priori predictable, and that the more extracted features are available, the better the performance will be in radar recognition[33, 34, 35]. New research should focus on greater accuracy in high noise environments (below 5 dB SNR) with varying Pulse Widths of five or greater radar types.

"Attack him where he is unprepared, appear where you are not expected."

Sun Tzu, The Art of War

3 Methods

As seen in the Literature Review, most new research either focuses on new classifiers using the old Pulse Descriptor Word (PDW) standard, or examines radar waveform features with specific classifiers in mind. Therefore, research opportunities exist where captured radar signals can be examined for features which can be used for recognition using any machine learning algorithm. This chapter postulates that accurate prediction can be achieved through a hierarchical approach to feature extraction, and that this could lay the foundation for improved interpolation and generalization in future research. The methods described in this chapter exploit different dimensions of a sampled spectrogram depending on the number of time intervals it contains. The resulting features are shown to provide accurate prediction when used with traditional machine learning classifiers.

The spectrogram matrix samples in each class contain multiple rows each representing a time interval, with N_{fft} number of columns each representing a frequency. Each cell contains a magnitude value representing the power at its frequency position. This paper claims that by changing the way features are extracted based on the number of rows, a greater degree of accuracy can be achieved when using traditional machine learning classifiers. In samples where there are few rows, a conversion into the frequency domain will yield few data points, therefore a different approach is taken where the max power per column is used as a feature. Alternatively, when a large number of time intervals are contained in a sample, the frequency domain provides a better degree of feature separability than the magnitude spectrum. The mid point between few and many time interval rows was determined empirically through experimentation to be 12.

3.1 Method Overview

In order to evaluate the feature extraction methods described in this chapter, two classifiers representing different interpretability and complexity are examined. Support Vector Machine (SVM) produces decision surfaces in high dimensional space, while K-Nearest Neighbor (kNN) develops representative values as the mean of features. When trying to choose between models, the best is both the simplest and most representative[36]. However, the aim of this thesis is to find how the feature extraction methods perform best under both these classifiers. Optimizing classification through manipulation of the model is outside the scope of this thesis. If SVM cannot substantially outperform kNN, and if kNN prediction accuracy remains high, the preferred classifier for the extracted features would be kNN, indicating a high reliance on the mean of representative values. However, this may also indicate that the data set may be overly homogeneous, making the extraction method susceptible to unknown edge cases in future data samples[37]. Best results using SVM will indicate that feature extraction is dependent on high dimensional space. A confusion matrix, and Monte Carlo Cross validation, both described in later sections, will be used to mitigate the risks associated overfitting, where training is too tightly connected with a specific set of data, making it less accurate when making predictions outside its training set cohort[38].

3.2 Simulated Radar Spectrogram Data

A spectrogram is a representation of a signal and has been used to graphically communicate data ranging from auditory samples to stock market fluctuations. In radar processing, time and frequency are the most important variables[39] used to describe a signal. As shown in Figure 3.1, spectrograms are often expressed as a color coded graphical plot with frequency and time as the X and Y axes, and a color spectrum representing power, a radar spectrogram typically illustrates the signal power and frequency distribution over time. The spectrograms included in the data set are the result of an analysis of electromagnetic radar frequencies representing “a time-frequency-intensity display of the short-time spectrum"[40]. Using a fast Fourier transform, radar pulse measurements in the time and frequency domains create a cohesive visual representation of the signal. Noise and other signal disturbances like jitter and stagger are easily visible in spectrograms making them a useful diagnostic tool to facilitate an improved understanding of these unwanted transient perturbations[41].

Simulated radar spectrogram creation details are outside the scope of this research. However, at a high level, the construction consists of an algorithm that uses sinusoids that are amplitude-modulated as in-phase and quadrature (IQ) components in a complex matrix, with radar specific pulse descriptor variables like pulse repetition interval and duration, along with arbitrary variables such as Signal to Noise ratio (SNR) and the $N-$ length[1] of the signal used in the Fast Fourier Transform (N_{FFT})[4]. N_{FFT} determines the number of signal samples and the number of frequency bins in each row of the spectrogram, which means that it also determines the sampling interval of the spectrogram; therefore the N_{FFT} ensures that all compared samples, regardless of their time interval, are the same size[11].

Parameters inside each simulated radar sample include raw signal in-phase and quadrature (IQ) components composed of complex numbers, modulation characteristics, noise parameters, and signal length[4]. Table 3.10 lists the simulated waveform classes used in the test data set. Additionally, in order to test the impact of noise, random Gaussian noise with a signal to noise ratio of 0 dB is introduced to study the impact it has on prediction accuracy.

Generated spectrogram data N_{FFT} in this thesis are all $1024 * 2$ in length, with the negative half of the spectrum culled from the dual-sided FFT during sample processing. Spectrograms are time-stacked and set with parameters, including N_{FFT} as a parameter, to an input sample rate, with each sample taking Ts seconds. Each row is therefore $T_{row} = Ts * N_{fft}$ seconds.

[1] N_{FFT} is not to be confused with the Inverse of the FFT, bearing the same name in some software packages.

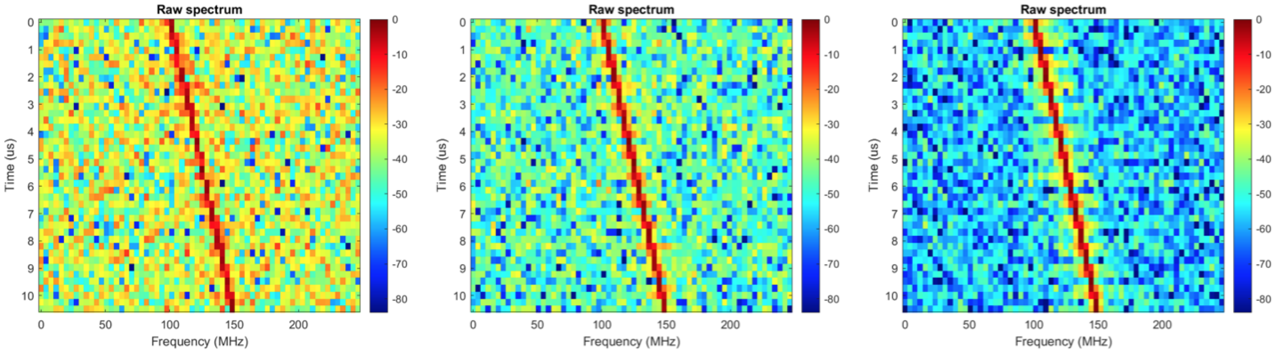


FIGURE 3.1: FM Linear Up -6dB, 0dB, and 6dB SNR.

3.2.1 Spectrogram Signal to Noise Ratio

The signal to noise ratio (SNR), at its most basic, is a dimensionless ratio of wanted to unwanted signal as shown by Equation 3.1.

$$SNR = P_s / P_N \tag{3.1}$$

White Gaussian noise is used to create a SNR at 0 dB by randomly creating a signal with varying frequencies and equal intensities. The equation of power and frequency with the random signal as shown in Equation 3.2 ensures that a constant power spectral density is maintained. Figure 3.1 below shows *FM Linear Up* sample number 101 at SNR -6dB, 0dB, and 6dB with a floor of -84 dB. A spectrogram with 0 dB has equal amounts of noise and signal power over the whole spectrogram matrix.

$$SNR = \frac{\int s^{2(t)dt}}{N_0/2} \tag{3.2}$$

A risk when using artificially categorized data is the creation of artifacts, which are unintended artificial patterns. The digital signals provided were sampled at a fixed rate and converted to spectrograms with $N_{FFT} = 2048$ sample bins per time interval. Figure 3.2 (left) shows a sequential plot of the number of time intervals per sample. It illustrates how the time intervals start small and get higher, peaking at the highest time interval every 50th sample. If training only sampled high time interval spectrograms, either by design or by accident, these artifacts would be prone to poor classifier performance, causing poor accuracy even with highly separable features. These

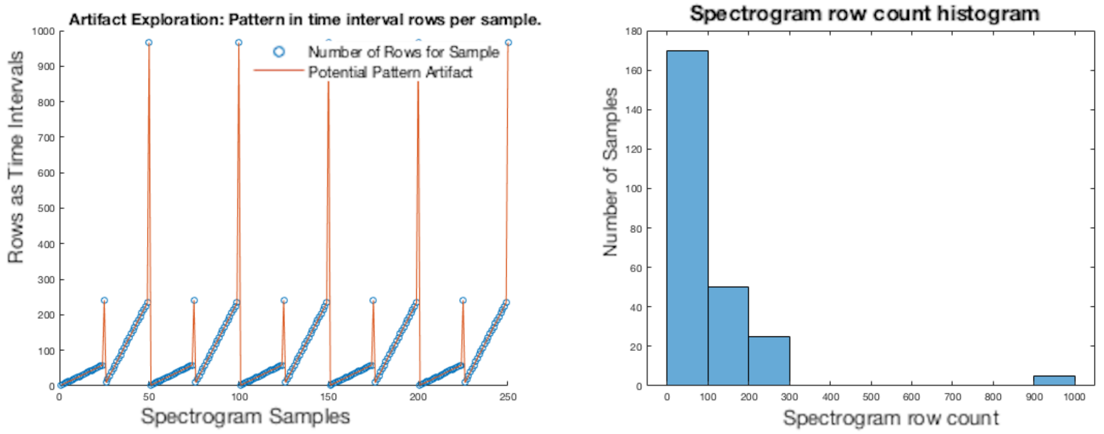


FIGURE 3.2: Repeating number of data rows per sample (left), histogram of row count per sample (right).

patterns could be formed from gradually increasing magnitudes in sequences of IQ data, groupings of data with time periods of similar duration, or any other artificially organized sequence of data by attribute. This artificial grouping could create problems during extraction and training by amplifying homogeneous traits in the data, distorting accuracy when using test data within its grouped cohort. This can be seen as both a false negative prediction, described in the next section, when a cluster of samples, which happen to share artificially similar attributes that are atypical to their class, are trained in a model used to detect a sample outside of that cluster. Alternatively, artifacts could reduce the prediction accuracy if a homogeneous cohort within the sample class is trained and tested within itself. Random selection and cross validation will be used to mitigate artifact issues during testing in order to break these potential patterns. This will be accomplished by shuffling the order of the 250 samples and then splitting data samples into an exclusive testing and training cohort set from the shuffled data for use in a Monte Carlo Cross Validation testing procedure explained earlier in the Section 3.5.2.

3.2.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is best defined in polarization with Initial Data Analysis (IDA). EDA looks at general characteristics of the data, including its structure and spread. IDA takes a more focused and narrow approach, looking at missing values and sparsity. An example of IDA used to examine data sparsity would be an inventory of missing values in a Pulse Descriptor Word database. A large number of missing values would decrease the statistical relevance of the value as a parameter, thus making the values sparse. An option here is to remove the value as a feature to increase statistical relevance with the remaining less sparse values. Unlike IDA, EDA methods focus more on overarching data structures[42]. This examination focused on both a plot of histogram view versus time periods, and standard deviation versus time period. The goal was to find where the frequency excursions cluster, with the goal of exploiting the structures in each cluster as features.

3.2.3 Waveform Class Descriptions and Shape Geometry

Figure 3.3 illustrates five classes with different Signal to Noise ratios, representing the test samples representing different types of basic waveforms used in radar. The first class is Continuous Wave (CW), which is characterized by a constant frequency and amplitude. CW is one of the earliest types of electromagnetic waveforms used in early wireless Morse code transmission, where signals were switched on and off through a sinusoidal transmitter. The rest of the classes encode information in the carrier waveform through Frequency Modulation (FM). The shape of the waveform modulation is how the class is named. For example, FM Symmetric Up contains mirrored frequency changes from its peak frequency whereas FM Linear Up displays linear changes in a singular direction. Shape geometry for each class can be seen in Figure 3.3, representing spectrograms with no noise.

3.2.4 Spectrograms Coherence

Due to the increasing dimensionality of the spectrogram as N_{FFT} increases, it suffers from the "Curse of Dimensionality" where the increasing volume of space causes the data to become sparse, reducing its statistical significance[43]. Too few features will result in less accuracy. Learning methods are confounded by too much variance[44], especially if the majority is comprised of pulse modulation as is often seen in radar samples. Figure 3.4 illustrates this "Curse of Dimensionality" with the figure top right showing a high number of superfluous signals accumulating in the center of the magnitude spectrum.

In radar samples included in this study, spectrogram rows represent time bins that range from 2 to 966 rows per time-frequency matrix. Figure 3.2 shows a histogram view of row counts per

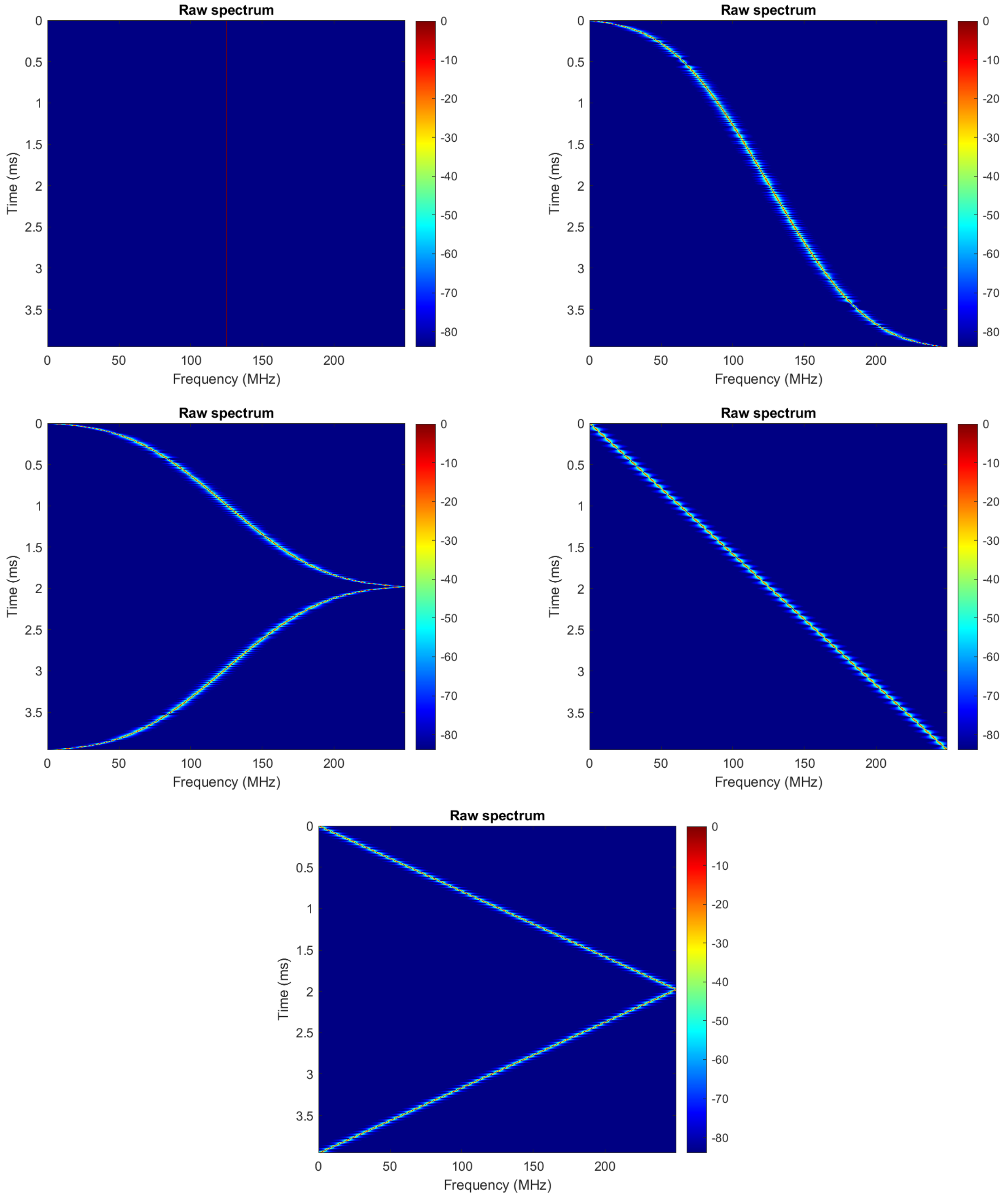


FIGURE 3.3: Spectrogram Samples for each class, CW (top left) displaying a faint red vertical straight line, FM Asymmetric Up (top right), FM Symmetric Up (middle left), FM Linear Up (middle right), and FM Triangle Up (bottom). Signals in these figures have no noise and are $N_{FFT} = 2048$ as a parameter of the frequency conversion process described in more detail in Section 3.2.

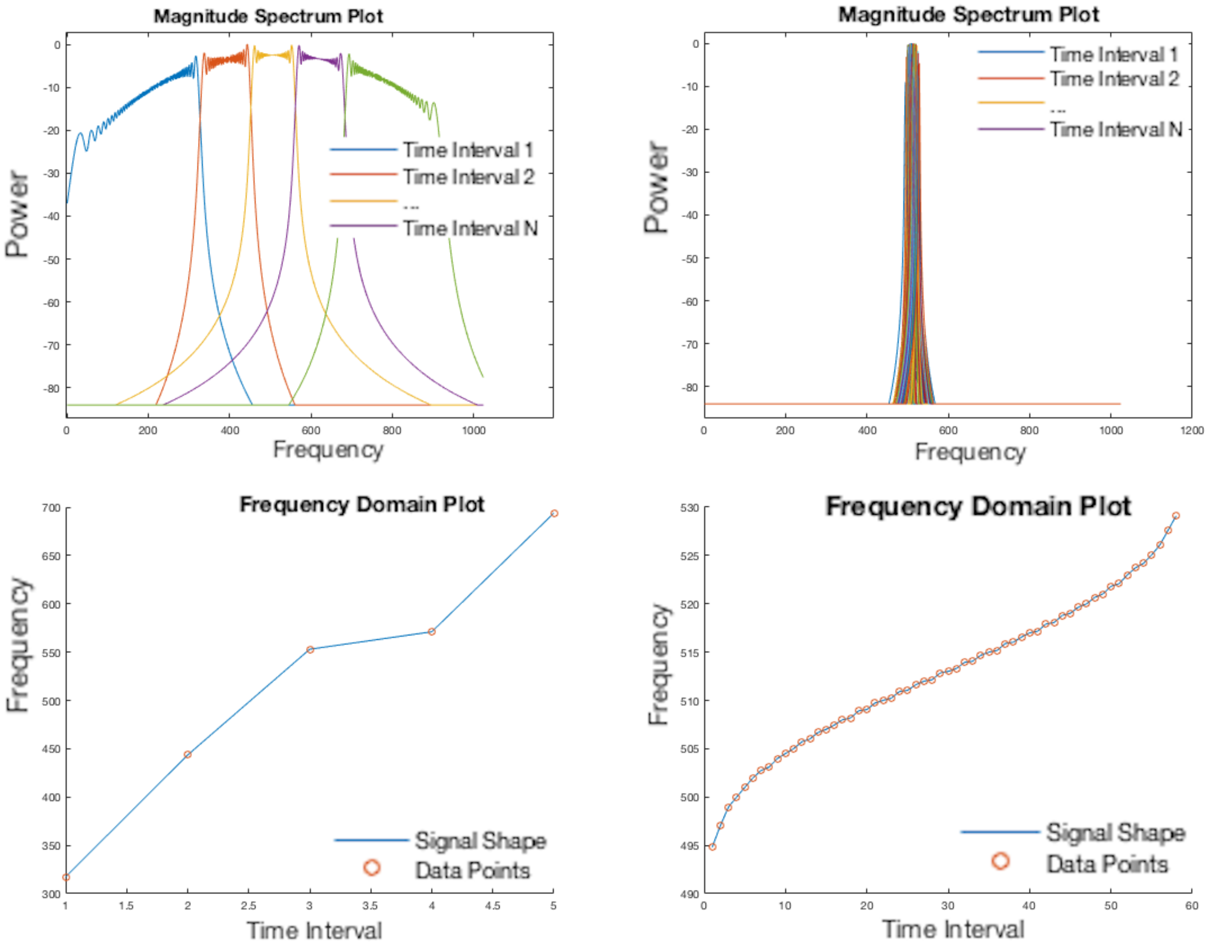


FIGURE 3.4: Samples 202 (left) and 81 (right), magnitude spectrum (top) and frequency domain (bottom).

sample, showing a large number of samples contain fewer than 100 time intervals. As rows increase, details begin to emerge, revealing feature shapes within the changing frequency over time, as seen in Figure 3.5. With very few time bins, samples with few spectrogram matrix rows become harder to differentiate when viewing them as Frequency versus Time. We can see this exhibited in Figure 3.5 where the spectrogram for both FM Symmetric Up (left) and FM Triangle up (right) in the top row look very similar but in the bottom row are easily visually distinguished. This is due to the top row containing samples with only 4 time intervals whereas the bottom row samples contain 36.

The frequency domain is a frequency versus time plot, whereas the magnitude spectrum is a plot of frequency against its power[3]. In cases where there are few time bins, the frequency domain may not reveal enough details about the signal shape to make an accurate prediction of

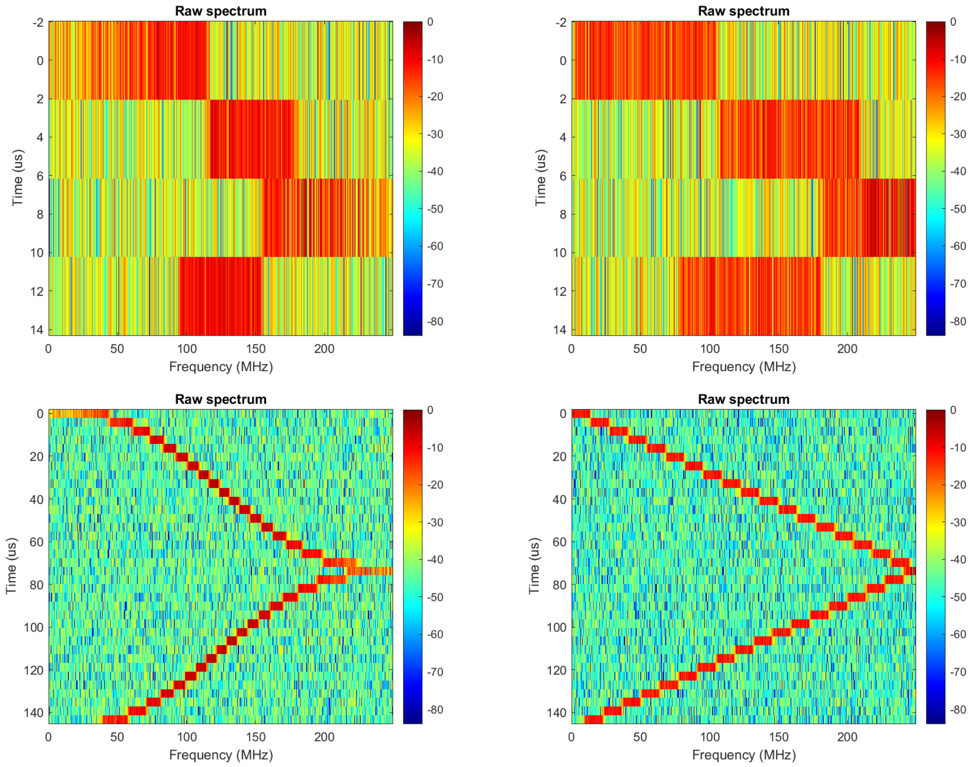


FIGURE 3.5: Samples with 0dB SNR, FM Symmetric Up (left), FM Triangle Up (right), low time interval (top), high (bottom).

signal type. For example, in Figure 3.4, the magnitude spectrum conversion of Sample 202 on the top left contains more salient data points compared to the relatively sparse frequency domain version on the bottom left. Alternatively, in Figure 3.4, the frequency spectrum conversion of Sample 81 on the bottom right shows a greater number of separable data points than its magnitude spectrum seen top right. The significant difference between these two samples is that Sample 202 contains 5 time intervals and Sample 81 contains 58 time intervals. Exploration of the data set has shown that with fewer rows the magnitude spectrum plot exhibits greater salient data separability, whereas the frequency domain works best for spectrograms with more rows. Empirically, the point at which the frequency domain out performs the magnitude spectrum was found to be 12 time intervals. However, with increased testing in future work, and with improvements to interpolation methods during extraction, this empirically designated demarcation may change.

3.2.5 Confounding Data

Spurious associations between variables can lead to data confounds as seen in the artificial pattern of spectrogram periods in Figure 3.2. The current sample set contains spectrogram samples constructed from IQ data that contains varying amounts of detail and differences in time intervals, as seen in the histogram in Figure 3.2. This is exploited by the hierarchical feature extraction methods in this thesis where features are extracted in the frequency domain where the number of rows is long and from the power spectrum when number of rows is short. However, if a sample contains little detail in the frequency domain and a short number of rows, the methods outlined in this study may fail to extract viable features. No samples in the simulated radar data contain low detail and a short number of rows. However, the possibility for this type of signal may exist.

3.3 Frequency Over Time Feature Extraction

Each spectrogram contains values for power and frequency per time interval row. To increase the statistical significance of features, extraction will focus on only two axes. This study hypothesizes that in spectrograms that contain a high number of rows, features extracted from the frequency spectrum will yield greater statistical significance than in the magnitude spectrum due to increased separability of matrix values as shown in Figure 3.4 and described in Section 3.2.4.

Figure 3.6 shows a magnitude spectrum plot of all time intervals stacked (left) where signal peaks can be seen to exhibit a pattern. Close inspection using a scaled version of the data shows where signal peak maxima fall directly on their respective frequencies. The plot shown in Figure 3.7 illustrates (left) where a signal's peak power maximum falls directly on a frequency marking its time interval frequency explicitly. In this plot, the data point maximum of the matrix first row lies directly on frequency 511, with data points on either side of the max value showing equal power distribution. This means that the data point represents the true magnitude peak maximum of the frequency. This will require first understanding how signal position influences peak representation in the frequency spectrum and also how best to interpolate these positions for accurate signal shape representation.

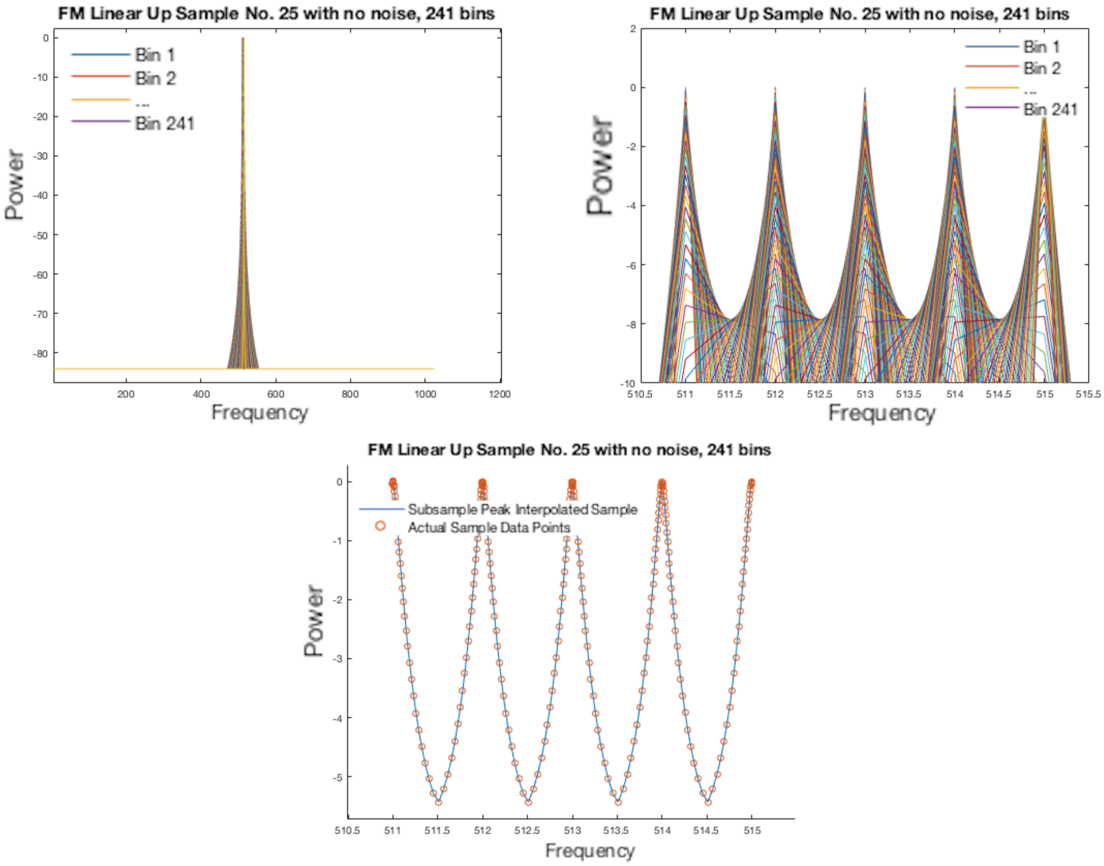


FIGURE 3.6: The frequency domain, where each row is a time interval consisting of $N_{FFT} = 2048$ frequency bias. Stacked (left), closeup of peaks (middle), maxima (right).

3.3.1 Defining Class Geometry using Subsample Peak Interpolation

In the frequency domain, each class has a unique signal geometry which can be exploited through feature extraction for training and prediction. In order to properly define this geometry, interpolation of values falling in between frequency bins will need to be calculated to precisely determine the correct shape during transformation. In cases where maximum magnitude falls directly on a frequency bin, transformations into the frequency domain are a simple conversion to a matrix where each frequency along the Y axis represents a time interval t sequentially on the X axis. However, the middle plot in Figure 3.6 illustrates how some of the time intervals, as indicated by differently colored signal lines, do not fall directly on a frequency bin. Thus, it can be concluded that the true peak magnitude maxima for the intra-frequency period is not being shown in the plot. Instead, the maxima exists at an interpolated point somewhere between the frequencies. In order to describe the true shape of the sample in the frequency domain, these frequency sub-bins must

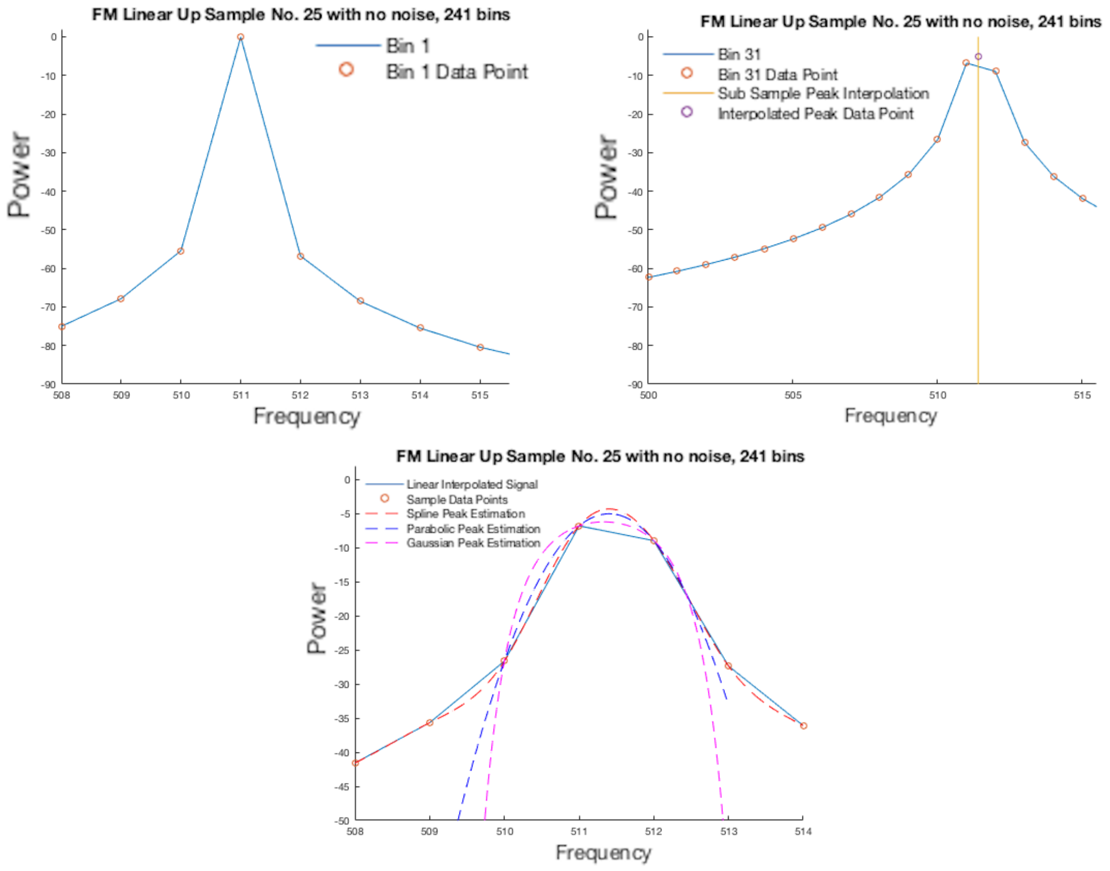


FIGURE 3.7: Frequency (left), subsampled frequency (middle) stacked interpolation methods (right).

be properly interpolated using a method that will accurately model the shape of the peak maxima.

Simple linear interpolation is a calculated position on a straight line between two points. This may work when describing a time interval signal peak that falls directly on a frequency bin as shown in the left plot in Figure 3.7. However, if the time interval signal peak is intra-frequency, as it is in the middle and right plots in Figure 3.7, linear interpolation will not reveal correct magnitudes at peak maxima. In addition to linear, three other interpolation methods were explored in this thesis: Spline, Parabolic, and Gaussian.

Cubic Spline Interpolation

Cubic spline interpolation is a piecewise polynomial method for determining a calculated position between two points on a curve, instead of a straight line as shown in linear interpolation. A third order polynomial calculates the position based on surrounding points[45]. Spline interpolation provides the greatest estimation for magnitude, given the intra-frequency data points (x_{i-1}, y_{i-1}) and (x_i, y_i) on either end of the uninterpolated max peak, as seen in the closeup plot (right) in Figure 3.7. The $n+1$ data points required for spline calculation $\{(x_i, y_i) : i = 0, 1, \ldots, n\}$ with polynomials $y = q_i(x), i = 1, 2, \ldots, n$ produced a curvature for the max peak $y = f(x)$ using Equation 3.3.

$$\kappa = \frac{y^n}{(1+y^2)^{3/2}} \tag{3.3}$$

Parabolic Interpolation

Parabolic peak interpolation can be used to find peak maxima by fitting three points $a < b < c$ corresponding to function values $f(a) \leq f(b) > f(c)$ to create a linear solution through direct substitution, as seen in Equation 3.4. The substitution is equivalent to the matrix product of the x values multiplied by the matrix of A, B, and C values.

$$\begin{aligned} A \cdot x_1^2 + B \cdot x_1 + C = y_1 \\ A \cdot x_2^2 + B \cdot x_2 + C = y_2 \\ A \cdot x_3^2 + B \cdot x_3 + C = y_3 \end{aligned} \equiv \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \tag{3.4}$$

Inverting the matrix and multiplying it by the y vector provides a direct solution for Equation 3.5. Finally, the peak maximum can be found from the quadratic as seen in lines 5 and 6 of Equation 3.5.

$$\begin{aligned} (x_1 - x_2)(x_1 - x_3)(x_2 - x_3) &= d \\ \frac{x_3 \cdot (y_2 - y_1) + x_2 \cdot (y_1 - y_3) + x_1 \cdot (y_3 - y_2)}{d} &= A \\ \frac{x_3^2 \cdot (y_1 - y_2) + x_2^2 \cdot (y_3 - y_1) + x_1^2 \cdot (y_2 - y_3)}{d} &= B \\ \frac{x_2 \cdot x_3 \cdot (x_2 - x_3) \cdot y_1 + x_3 \cdot x_1 \cdot (x_3 - x_1) \cdot y_2 + x_1 \cdot x_2 \cdot (x_1 - x_2) \cdot y_3}{d} &= C \\ \frac{-B}{2 \cdot A} &= x_{max} \\ \frac{C - B^2}{4 \cdot A} &= y_{max} \end{aligned} \tag{3.5}$$

The results of curvature κ can be seen as a blue dotted line in Figure 3.8 middle and closeup right plots. It lies in between spline, resulting in the largest magnitude, and Gaussian as lowest.

Gaussian Interpolation

Gaussian peak interpolation follows a similar process to find A, B, and C values using a mutual denominator d. However, a log function of the y matrix is used, as seen in Equation 3.6.

$$\begin{aligned}
log(y) &= lny \\
x_3 \cdot (lny_2 - lny_1) + x_2 \cdot (lny_1 - lny_3) + x_1 \cdot (lny_3 - lny_2) &= A \\
x_3^2 \cdot (lny_1 - lny_2) + x_2^2 \cdot (lny_3 - lny_1) + x_1^2 \cdot (lny_2 - lny_3) &= B \\
x_2 \cdot x_3 \cdot (x_2 - x_3) \cdot lny_1 + x_3 \cdot x_1 \cdot (x_3 - x_1) \cdot lny_2 + x_1 \cdot x_2 \cdot (x_1 - x_2) \cdot lny_3 &= C \\
(x_1 - x_2) \cdot (x_1 - x_3) \cdot (x_2 - x_3) &= d \\
\frac{-B}{2 \cdot A} &= x_{max} \\
exp(\frac{\frac{C}{d} - B^2}{4 \cdot A \cdot d}) &= y_{max}
\end{aligned} \tag{3.6}$$

After natural log lny is derived from values in vector $y = [y_1, y_2, y_3]$, the inverse of the matrix, similar to Equation 3.4, is calculated for A, B, and C values. These values are then plugged into line 6 and 7 in Equation 3.6 to find the maximum x_{max}, y_{max} intra-frequency peak magnitude. Figure 3.8 illustrates Gaussian curvature κ as a pink dotted line.

Curvature κ Interpolation Conclusions

Figure 3.7 (middle, right) indicates where the subsampled peak is estimated to be located using the various methods illustrated in the plot's legend. It juxtaposes Spline, Parabolic, and Gaussian curvature κ estimations with Spline as the highest, Gaussian as the lowest, and Parabolic as the middle. All calculations require at least three points, which consist of a non-interpolated max value and two adjacent non-interpolated data points. Empirical testing using known points found that Parabolic interpolation is the most accurate estimator of points typically found in a power versus frequency matrix.

After the subsample peaks are calculated, a new matrix is created containing the peaks and amplitudes of each bin in sequence. As we can see in Figure 3.8 (left), this creates a shape similar to the max value of the tip of the signal as seen in Figure 3.6 (middle). However, the class geometry of

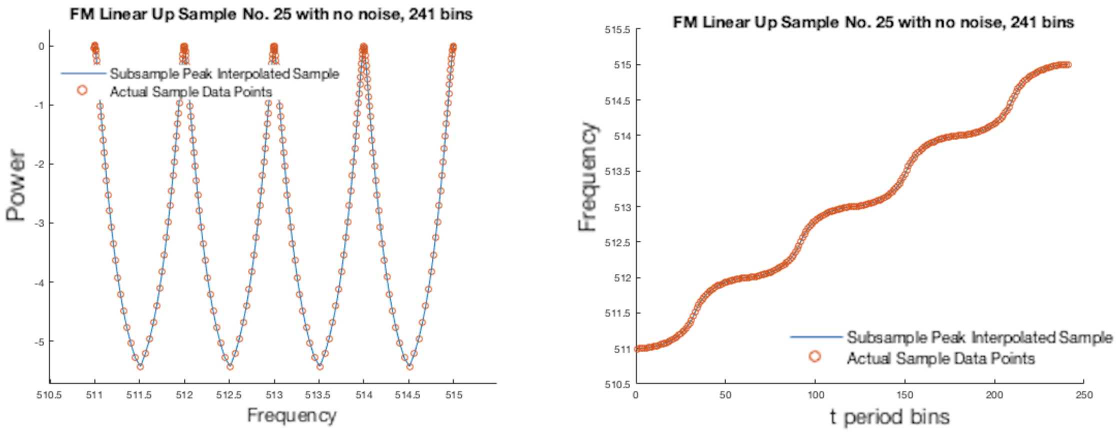


FIGURE 3.8: Data points, subsample peak interpolated (left), conversion to Frequency Domain (right).

the frequency isn't seen until the sample is converted into the frequency domain, as seen in Figure 3.4 (bottom right). Evidence of poor interpolation performance can be seen as undulations in the signal, evident in Figure 3.8 (right), with greater undulations in spectrogram samples containing fewer rows. Prediction accuracy was good despite these interpolation issues being uncorrected; however, future work may discover minor increases in performance with better subsample peak interpolation techniques using more than two adjacent samples to perform interpolation.

Conversion from Spectrogram Matrix to Frequency Domain

Each row in the spectrogram contains N_{fft} bins, representing the length of the signal, with each bin containing a magnitude, representing the signal power as decibels. After the intra-frequency magnitude is interpolated between frequency bins, as described in the Curvature κ step in Section 3.3.1, a matrix is created where each time interval row is sequentially ordered into a matrix of frequency values, producing a plot where frequency is plotted as the y axis and time as the x axis.

$$\begin{bmatrix} p_1 < a_{max-1}, a_{max}, a_{max+1} > \\ \cdots \\ p_n < a_{max-1}, a_{max}, a_{max+1} > \end{bmatrix} = \begin{bmatrix} f_{max1} \\ \cdots \\ f_{maxn} \end{bmatrix} = \begin{bmatrix} f_{max1}, \cdots, f_{maxn} \end{bmatrix} \tag{3.7}$$

Equation 3.7 shows a high level representation of spectrogram matrix time interval rows t where each row t is subsample peak interpolated parabolically using adjacent maxima values a_{max-1}, a_{max}, and a_{max+1} to create a $1xN$ matrix of sequential frequency rows. The 1xN matrix is then transposed to create a matrix of frequency maxima interpreted as the Frequency domain.

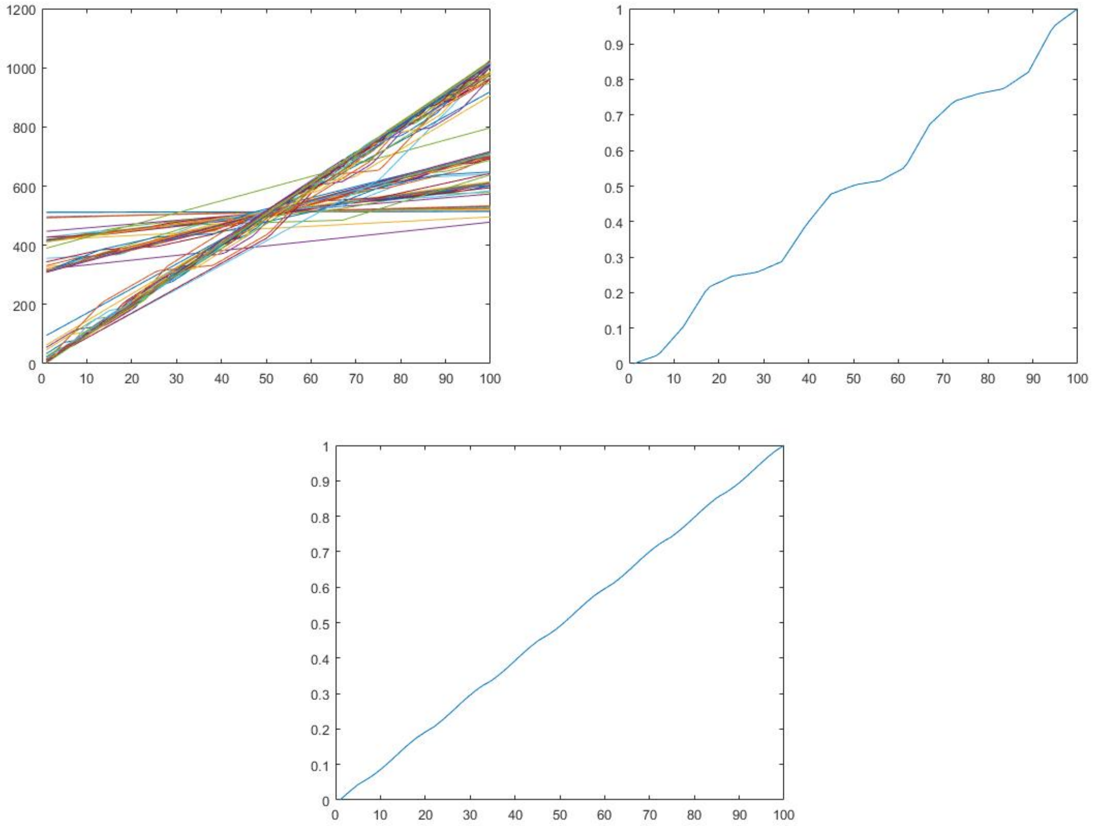


FIGURE 3.9: All FM Linear Up samples stacked (left), Frequency domain (middle, right) normalized.

Normalization and Generalization

After the spectrogram matrix is processed into the frequency domain using parabolic subsample peak interpolation, the frequency range is finally normalized from a range of 1 to $NFFT$ to floating point values from 0 to 1.

$$\begin{aligned} (x_1, \cdots, x_n) &= x \\ \frac{x_i - min(x)}{max(x) - min(x)} &= z_i \end{aligned} \tag{3.8}$$

Equation 3.8 illustrates the process where x represents the set of all frequencies in a sequence. The second line in Equation 3.8 solves for z_i as the i_{th} value in the series. The new normalized matrix will contain frequency f_{min} and f_{max} values between 0 and 1.

A plot (left) in Figure 3.9 of all FM Linear Up samples illustrates the variability and inseparability of the signal bin traversal seen in the frequency domain. After normalization, the plots (middle, right) in Figure 3.9 show approximate class geometric representation, with the exception of the middle plot where undulations from the signal side lobes influence the magnitude interpolation during subsampling. Another influencing factor is the signal movement from one frequency bin to the next. If more than one bin is traversed per row, the interpolation changes due to blurring inside multiple bins. The plot on the right in Figure 3.9 is straighter than the middle plot both because these undulations are scaled out due to its high number of time interval length and also because the interpolation methods do not best represent the subsample ranges. The middle plot spectrogram has 19 time intervals, the right plot has 48 time intervals. This means that the undulations in the plot on the right with the greater number of rows are less noticeable, in part, due to a higher time interval, making time interval an influencing factor of signal geometry.

Finally, feature values are extracted through interpolation into 9 points. The number of points were picked empirically after some tests found that fewer points lead to lost features around the beginning and ending of FM Symmetric Up, making it less separable to FM Linear; and more points introduced redundancy in the center of the signal, reducing its statistical significance during training. An odd number of points was chosen to capture the center point of mirrored features found in FM Symmetric Up and FM Triangle Up. Figure 3.13 illustrates an example (left) of a class with 9 designated feature locations represented as red circles on a plot.

3.4 Power Over Frequency Feature Extraction

Figure 3.4, bottom row, show frequency domain plots with the left illustrating a low feature space with only 5 time intervals with the right plot containing 81 time intervals and a much greater feature space, as illustrated by well defined starting and ending curves. The plot on the right will perform better for both prediction and training due to its increased feature space, particularly in the capture of curves during the start and end, characteristic to asymmetric signals. However, in the magnitude spectrum, Figure 3.4, top left, illustrates a feature rich plot containing only 5 time intervals.

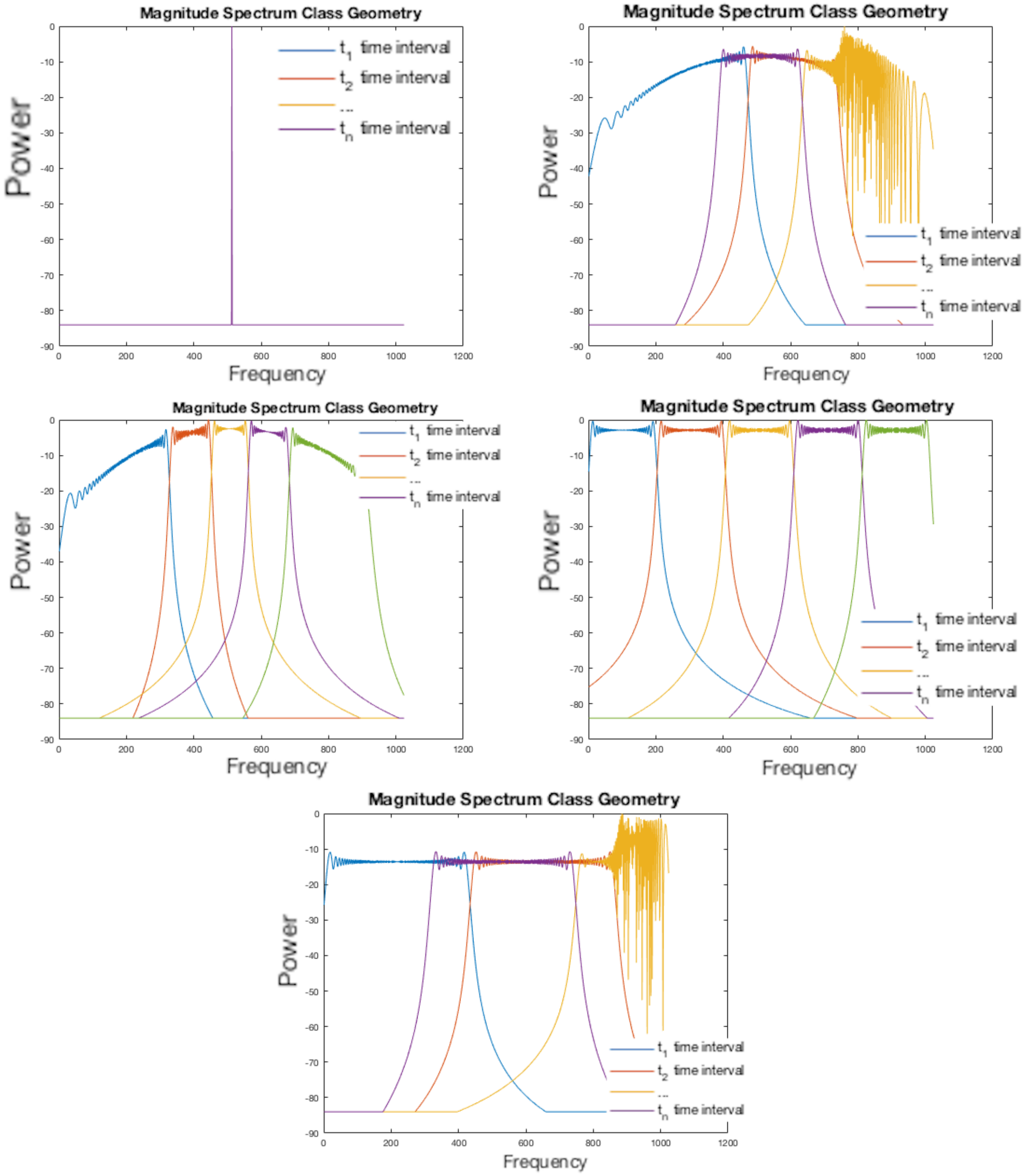


FIGURE 3.10: Magnitude spectra showing unique class geometry for CW (top left), FM Symmetric Up (top right), FM Asymmetric Up (left middle), FM Linear Up (middle right), and FM Triangle Up (bottom).

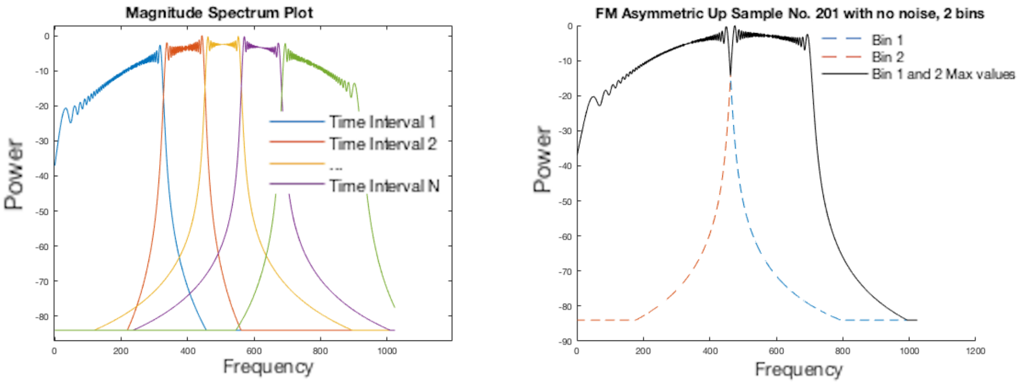


FIGURE 3.11: Magnitude spectrum (left) $\max\limits_{r \in rows} s_{rc}$ processed magnitude spectrum (right).

3.4.1 Defining Class Geometry using Max Smoothing

In order to plot changes in the signal, as it traverses the frequency range per time period, the spectrogram must be transposed, so that the row and column index for each element are interchanged. Equation 3.9 shows spectrum S being transposed before being processed to find the max signal power where the signals cross over each other per time interval.

$$\textbf{Spectrogram} = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ b_1 & b_2 & \cdots & b_n \end{bmatrix}^{\tau} = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ \cdots & \cdots \\ a_n & b_n \end{bmatrix} \tag{3.9}$$

$$Spectrogram = S, Rows = r, Columns = c$$

$$\max_{r \in rows} s_{rc}(i = 1, 2, \cdots, n)$$

Figure 3.10 illustrates the unique geometry for each class. The CW signal, being extremely separable from the other signals due to the consistent frequency travel over all samples, can be safely classified first using its distinctive continuous geometric identifier at peak magnitude. In the remaining 4 classes, two distinct geometric characteristics appear: firstly, the straight or curved travel over the frequency range, and secondly, the interference at the end of the signal. The unique geometry seen as interference at the end of magnitude spectrum samples for FM Triangle Up and FM Symmetric Up in Figure 3.10 is presented regardless of the number of time intervals, making

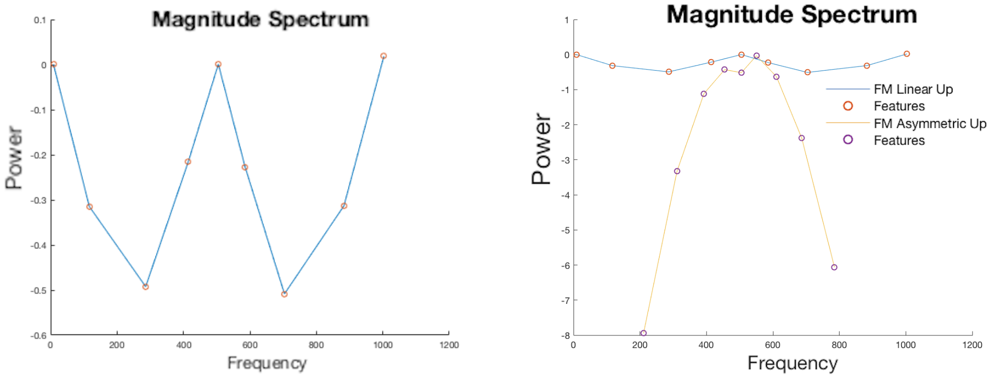


FIGURE 3.12: FM Linear Up scaled (left) FM Linear Up scaled relative to FM Asymmetric Up (right).

this a critical feature. Therefore, during feature extraction, normalization and interpolation should be sensitive to these attributes.

Conversion from Spectrogram Matrix to Magnitude Domain

After transposing the matrix as shown in Equation 3.9, the max of each time interval must be found. Transposing the signal is done in order to capture power on the Y axis while measuring frequency on the X axis. Time intervals are plotted as distinct and separate excursions across the frequency X axis with the intention that the peak and position will be generalized as a feature for training and prediction. If the interval peak is not located directly on a frequency bin, subsample peak interpolation will need to be utilized, as described in the previous Section 3.3.1 for spectrogram conversion into the frequency domain.

Figure 4.1 shows the unique and distinct excursion patterns differentiating the symmetrical signals from the others. The blue line in each plot represents the feature geometry and the red dots along the blue line represents 9 generalized features. FM Triangle Up (left, top) contains very little frequency detail and very few (7 rows) time intervals compared to the same class with high detail and low time intervals (7 rows). Typical signal excursion for FM Triangle Up on a magnitude spectrum is a straight line, with some interference at the end of the frequency range, and with terminal time intervals appearing in mid frequency as seen in the plot (right, top). However, due to the extremely fast excursion and low time interval, exhibiting very low detail, the plot (left, top) does not contain interference detail. However, excursion is seen in those plots (left and right, top) with feature points indicating that ending time intervals are seen mid frequency, hypothesizing that the indication of terminal feature values found in mid frequency is an indicator for signal symmetry, as seen in FM Symmetric Up and FM Triangle Up. This is opposed to plots (left and right, bottom) where ending feature values are positioned at the end of the frequency range, typically seen in asymmetrical signals like FM Asymmetric Up and FM Linear Up. These features will be critical for symmetrical geometry pattern detection and should yield good results with any classifier.

Normalization and Generalization

Plots (left, top and bottom) in figure 4.1 appear to contain a slight curve in the peak magnitudes. However, this is an artifact of the scale in the plot due to the low detail, low time interval, and intrapeak frequencies of the signal. When compared to other signals, these two plot features (left, top and bottom) appear relatively flat, as shown in Figure 3.12. These exaggerated features were even more pronounced after normalization, therefore it is concluded that in cases where time interval and detail is low, normalization is detrimental to feature enhancement, and was avoided.

3.5 Classification Methods

The two common Machine Learning classifiers used in this study are k-Nearest Neighbor (kNN) and Support Vector Machine (SVM). These two classifiers were chosen due to their ubiquity and evaluation simplicity. In this case, kNN has been shown to approach best possible performance within a factor of 2 as population size $N \mapsto \infty$ [46] while the binary nature of SVM can highlight strengths and weaknesses within the feature extraction separability methods[47].

3.5.1 Confusion Matrix

Insight will be gathered from a confusion matrix of prediction results tabulating false and true negative and positive predictions with a column for two standard deviations of accuracy per class, a 95% confidence interval for the prediction, as seen in raw data tabulations in the Appendix.

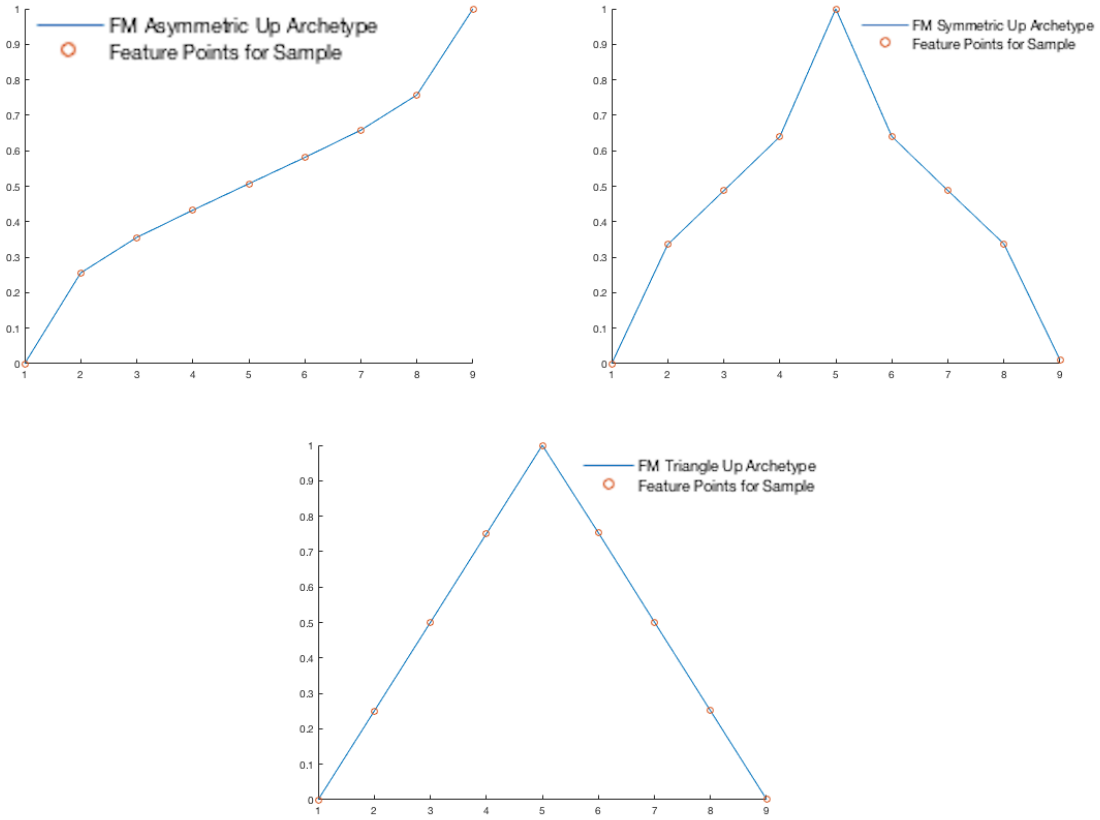


FIGURE 3.13: Features as red points are defined for each class sample.

Additionally, exploratory data analysis was done to identify gaps in the statistical relevance of features for spectrogram samples provided in the sample data with some edge cases discovered, which may lead to potentially confounding results in future work done with new samples. After potential confounds and false positives and negatives are illustrated, the resulting true positives and true negatives with standard deviations to two places will reveal method accuracy with confidence. Finally, Accuracy, Specificity, Sensitivity, and Precision metric will be defined and reported in the Results Chapter 5 of this thesis.

3.5.2 Monte Carlo Cross Validation

In order to evaluate the model, and thus evaluate the feature extraction methods, a less-optimistic testing process should be employed to mitigate the risks of over-fitting and bias during testing. A relatively simple procedure is to use Monte Carlo Cross Validation[48], where random subsampling is split and done N times. Here, the data set is randomly shuffled and split during each iteration, after which the model is trained using one sample set and validation is assessed using the other. Final results are averages over all N cross validation splits. Due to the random shuffling and splitting of test sample data, there is no dependency on a fixed number of iterations, as may be seen in other Cross Validation methods. However, a major limitation of the Monte Carlo method, due to inherent randomness, is that some observations have a probability $\frac{1}{2^n}$ of never being seen either as training or test cases. The probability of this occurring can be mitigated through many splitting and shuffling iterations, which lower the probability of observations not being using in a training or prediction iteration. Tests done in this thesis, using Monte Carlo, will first separate spectrogram data into two cohorts, one with less than 12 time intervals and one with 12 or more, to ensure that shuffled splits will only provide spectrograms appropriately sized to the correct time interval size prediction model.

3.5.3 k-Nearest Neighbor

kNN is a non-parametric algorithm where the input consists of adjacent samples in the feature space[49] and the output is inclusion to a class based on a majority vote among neighbors for the most common and closest class[38]. This classification method is particularly prone to degradation from features that lack relevance, for example excessive noise or redundancy, or if features scale inconsistently with their importance[38]. This makes feature extraction a critical step in kNN classifier viability. Computation is approximated locally and deferred until classification making this a relatively inexpensive process compared to SVM. As a side benefit, it is a relatively simple machine learning algorithm which can be used to reduce complexity in a recognition task by removing clustered samples that can be classified immediately.

Points to be classified are evaluated based on their distance to points inside a data set, in the simplest implementation of kNN, with k defined as the number of neighbors used to define the label. This paper uses Matlab's implementation of kNN, using Euclidean distance as the method of measurement between points, shown in Equation 3.10.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} \tag{3.10}$$

A total of 9 feature points are extracted from each of the 1000 samples using the methods outlined in this paper, making the multidimensional distance measurement seen in Equation 3.11.

$$d(p,q) = \sqrt{\sum_{i=9}^{1000} (p_i - q_i)^2} \tag{3.11}$$

This rudimentary implementation of kNN is used to focus attention on the paper's feature extraction techniques by remaining as agnostic to the data set as possible.

3.5.4 Support Vector Machine

Support Vector Machine (SVM) is used in many recent research papers on automatic radar emitter recognition[50, 51, 52, 53]. With its ability to perform nonlinear classifications using a "kernel trick" to move data to a higher dimensional space[38], SVM can make non-linear binary classifications on sparse data. Sparsity occurs as features increase without an increase in available data, diminishing the statistical significance of that data[54]. This limits the accuracy of SVM without large amounts of data to provide support vectors on the boundary of non-linearly separable samples. In order to mitigate data sparsity, spectrogram samples with lower time intervals will be analyzed using features extracted in the magnitude spectrum rather than in the frequency domain. Furthermore, to avoid the "Curse of Dimensionality" mentioned earlier in the chapter, spectrograms with a high number of time intervals will be analyzed using features extracted in the frequency domain; this will mitigate the diluted statistical significance of key features when using entire spectrogram data samples as input. Figure 3.4 illustrates how the magnitude spectrum on the top right can cause unwanted high dimensionality, and how the frequency domain can present sparse data in the bottom left, all within the same data set. Classifications in this thesis will use hard margin linear SVM where 9 features points per sample will make a feature vector used in training such that in group. The 9 feature points were chosen as the minimum number of equidistant points capable of capturing both starting and ending undulations in the signal geometry, as seen in FM Symmetric Up and FM Asymmetric Up, and also for the middle point significance in capturing signal symmetry, as seen in CW, FM Triangle Up, and FM Symmetric Up.

$$(\overrightarrow{x_1}, y_1), \ldots, (\overrightarrow{x_n}, y_n) \tag{3.12}$$

Shown in Equation 3.12, y_i represents a binary 1 or -1 depending on $\overrightarrow{x_i}$ class. This binary classification is the result of a "maximum-margin hyperplane" where the distance between the hyperplane and $\overrightarrow{x_i}$ is maximized satisfying Equation 3.13 where $\overrightarrow{w}$ is the hyperplane normal vector.

$$\overrightarrow{w} \cdot \overrightarrow{x} - b = 0 \tag{3.13}$$

If the SVM model contains data that is linearly separable, binary predictions made with Equation 3.13 will equal 1 or -1. The width of the margin, or distance between the two 1 and -1 boundaries, is $\frac{2}{\|\overrightarrow{w}\|}$ where b is the offset determined by $\frac{b}{\|\overrightarrow{w}\|}$ along the normal vector $\overrightarrow{w}$. If we want to maximize our margin to obtain the best classification result, we must minimize $\left\|\overrightarrow{w}\right\|$ subject to $y_i\left(\overrightarrow{w_i} \cdot \overrightarrow{x_i} \cdot -b\right) \geq 1$ where $i = 1, \ldots, n^n$ (with n^n representing the multidimensional hyperplane) to solve for $\overrightarrow{w}$ and b as shown in Duda[38] *et al*.

$$\overrightarrow{x} \mapsto sgn\left(\overrightarrow{w} \cdot \overrightarrow{x} - b\right) \tag{3.14}$$

Therefore, classification of feature vector $\overrightarrow{x}$ can be summarized as the signum function of the distance from the feature vector to the boundary, as shown in Equation 3.14.

$$sgn(x) := \left\{ \begin{matrix} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{matrix} \right\} \tag{3.15}$$

The signum function is defined in Equation 3.15 and is used to extract the sign of the real number result of $\overrightarrow{w} \cdot \overrightarrow{x} - b$.

3.5.5 One-versus-Rest SVM

One approach to multiple class binary prediction using SVM is through One-versus-Rest, where the predicted class is trained against negative examples of all other classes[55]. This requires each class prediction test to have a separate model where features outside the SVM margin represent all classes that are not the positive class. A large problem with this type of test is that with five classes, training will be highly unbalanced against the positive test class, with four representatives for a negative class for every one positive. However, high time interval samples with no noise displayed near perfect Accuracy with 0dB SNR showing weaknesses in FM Symmetric Up with an Accuracy score 0.96 and Precision of 0.92, as shown in Table 4.2. The sharp drop in precision is due to the comparatively small number of features used to describe the flaring geometry at the ends of the FM Symmetrical Up signal shape when compared to noise effects influencing shape geometry in other signals.

3.5.6 Hierarchical SVM

Similarly to One-versus-Rest SVM, Hierarchical SVM requires no more than two classes to be trained and tested against each other. However, the major weakness of One-versus-Rest training, where the positive class is outnumbered four to one, is mitigated through a cascading hierarchy of tests done in succession, as described in Section 3.5.4. This hierarchical sequence is based on strongest separability, with cohorts made from a sample's unique and distinct geometric features, as shown in Table 3.3. This sequence is started with the most separable sample, CW versus everything else. Both empirically and as shown through consistently high Accuracy and Precision metrics, CW prediction remains accurate through low and high time intervals between no noise and 0dB SNR, as shown in Table 4.3. Removing CW from the training set after prediction enables subsequent training to focus on either geometrically symmetrical or asymmetrical samples, following which training can focus on separability between either FM Symmetric Up versus FM Triangle Up, and FM Linear Up versus FM Asymmetric Up.

"Always mystify, mislead, and surprise the enemy, if possible;"

Sun Tzu, The Art of War

4 Results

At the highest level, a spectrogram is a measure of power and frequency information over time, derived from a Discrete Fourier Transformation (DFT) outputting frequency, phase, and amplitude encoding of the component sinusoids in a radar signal[11]. Spectrograms, including those in the sample data set, transmit various frequencies at full power, over N time intervals, in order to estimate the location of objects at a distance by measuring their return characteristics[4]. Therefore, training those unique characteristics using a classification algorithm will allow predicting what unknown sources populate a potential battlefield[5]. The radar spectrograms used as data sets in this thesis are simulations based on an algorithm developed by Australia's Defence Science and Technology Group (DST). Details of the algorithm are outside the scope of this thesis; however, an exploration of parameter details will yield a better understanding of the samples. Because this thesis is focused more on feature extraction than on classifier performance, shortcomings found in classifier metrics will be used to deduce where features do not provide sufficient information for classification instead of how classification can be improved.

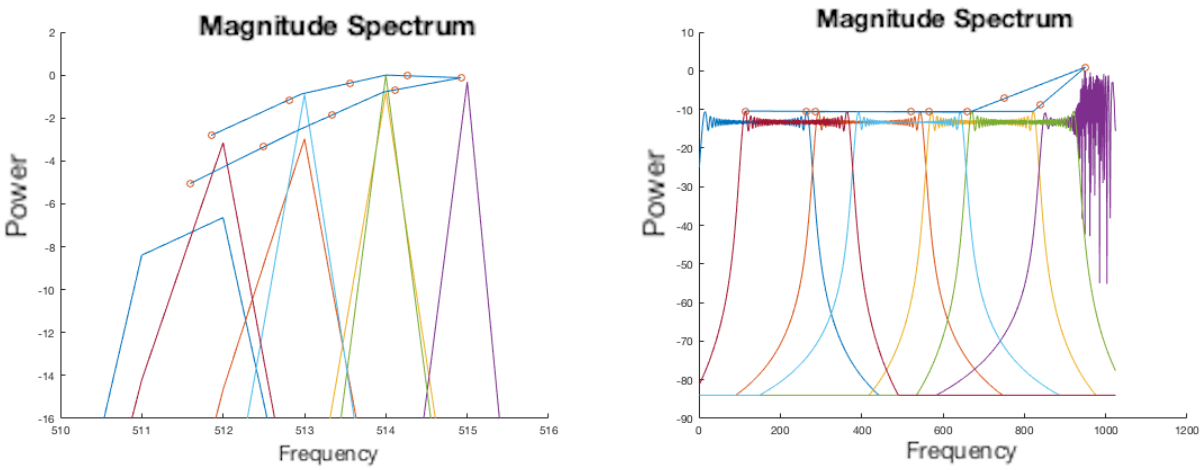


FIGURE 4.1: Low Time Interval FM Triangle Up with low detail (left) and high detail (right).

4.1 Testing Methods

Prediction is first attempted using k-Nearest Neighbor, using Monte Carlo Cross Validation described in Methods Chapter 3. Support Vector Machine testing will use a both One-versus-Rest and Hierarchical binary predictions. One-versus-Rest multiclass SVM testing uses the binary classification ability of SVM by learning positive predictions for each class one at a time with all other classes grouped as a single negative class – "the rest"[55]. Hierarchical SVM separates positive and negative predictions into more evenly populated cohorts. As shown in Figure 4.2, Step 1 shown by a blue circle starting with CW prediction. If One-versus-Rest SVM prediction for CW produces poor accuracy results, starting the hierarchy with this test would be a major liability for all other predictions, due to their dependence on CW exclusion in all subsequent tests. However, CW was shown to be highly separable in One-versus-Rest SVM prediction results, as shown in the raw metrics in Table A.5. Step 2 excludes CW from the SVM training model and combines FM Symmetric Up with FM Triangle Up for a positive geometrically Symmetrical cohort, and FM Asymmetric Up with FM Linear Up for a negative geometrically Asymmetrical cohort. Finally, having been classified as either symmetrical or not, in Step 3 training samples are separated into FM Asymmetric Up and FM Linear Up for geometrically asymmetrical prediction and FM Symmetric Up and FM Triangle Up for geometrically symmetrical prediction.

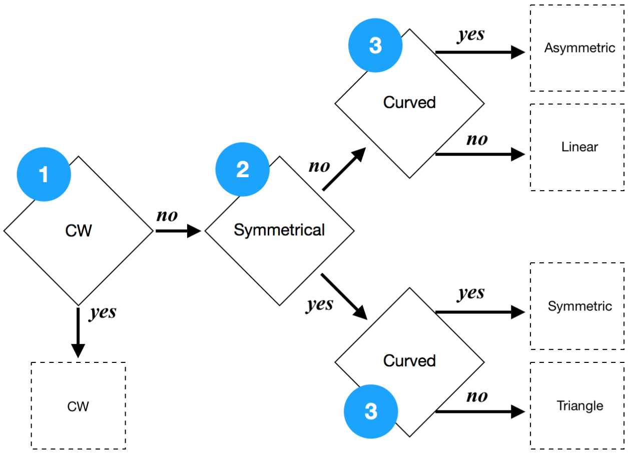


FIGURE 4.2: Hierarchical SVM decision tree showing steps to final prediction.

4.2 Confusion Matrix

In order to gain further insight into feature extraction performance, a confusion matrix is derived from a combination of classifier prediction results that are analyzed using ground truth versus observed results[56]. These results are further distilled into Accuracy, Sensitivity, Specificity, and Precision, to determine where weaknesses and strengths are found in the ways the feature extraction methods are exploited by the classifiers. Table 4.1, 4.2, and 4.3 separates these results into no noise, 0 dB SNR, low, and high time interval metrics. An explanation of these measures are outlined in Section 4.2.1 and raw data results can be found in the Appendix at the end of this thesis.

4.2.1 Definition of Metrics Used

Ground Truth and Observation

The two axes of every table in this thesis containing results from the classifier contains predictions vertically along the X axis and observations, as ground truth, horizontally along the Y axis. The ground truth in these tables represents the objective data set whereas the predictions represent output from the classifier. Generally, in machine learning, ground truth represents information provided through objective data or direct observation, as opposed to being inferred[37].

TABLE 4.1: kNN Performance Metric

	Ground Truth	Accuracy	Sensitivity	Specificity	Precision
kNN	CW HI No Noise	1.00	1.00	1.00	1.00
	HI 0dB SNR	0.99	0.97	1.00	1.00
	LI No Noise	0.98	1.00	0.97	0.91
	LI 0dB SNR	0.98	1.00	0.97	0.90
	Linear HI No Noise	1.00	1.00	1.00	1.00
	HI 0dB SNR	1.00	1.00	1.00	0.99
	LI No Noise	0.81	0.61	0.86	0.53
	LI 0dB SNR	0.81	0.57	0.87	0.52
	Symmetric HI No Noise	0.99	0.99	0.99	0.98
	HI 0dB SNR	0.99	0.97	0.99	0.96
	LI No Noise	0.87	0.59	0.94	0.72
	LI 0dB SNR	0.86	0.63	0.92	0.67
	Asymmetric HI No Noise	1.00	1.00	1.00	1.00
	HI 0dB SNR	1.00	1.00	1.00	0.99
	LI No Noise	0.88	0.70	0.93	0.71
	LI 0dB SNR	0.87	0.68	0.92	0.69
	Triangular HI No Noise	0.99	0.98	1.00	0.99
	HI 0dB SNR	0.99	0.98	0.99	0.96
	LI No Noise	0.86	0.63	0.92	0.67
	LI 0dB SNR	0.87	0.61	0.93	0.68

Positives and Negatives

The combination of ground truth versus observation can yield a true positive (TP), true negative (TN, false positive (FP), and false negative (FN). A true positive is where a prediction matches a positive observation whereas a true negative correctly identifies a negative observation. A false positive incorrectly predicts a match where there is none, whereas a false negative fails to predict a match[37]. In an example where radar detection is used in a stealth campaign, a false negative would be less desirable than a false positive, due to a false negative possibly leading to stealth failure whereas a false positive would lead to unnecessary evasive maneuvers.

Accuracy

$$Accuracy[37] = \frac{TP+TN}{TP+TN+FP+FN}$$

Sensitivity

$$Sensitivity[37] = \frac{TP}{TP+FN}$$

Specificity

$$Specificity[37] = \frac{TN}{TN+FP}$$

TABLE 4.2: SVM One-versus-Rest Performance Metrics.

	Ground Truth	Accuracy	Sensitivity	Specificity	Precision
One-versus-Rest	CW HI No Noise	1.00	1.00	1.00	1.00
SVM	HI 0dB SNR	0.99	0.94	1.00	1.00
	LI No Noise	0.97	1.00	0.97	0.88
	LI 0dB SNR	0.96	1.00	0.95	0.83
	Linear HI No Noise	1.00	1.00	1.00	1.00
	HI 0dB SNR	1.00	1.00	1.00	0.99
	LI No Noise	0.82	0.59	0.88	0.55
	LI 0dB SNR	0.78	0.53	0.85	0.46
	Symmetric HI No Noise	1.00	0.99	1.00	1.00
	HI 0dB SNR	0.96	0.88	0.98	0.92
	LI No Noise	0.81	0.55	0.87	0.52
	LI 0dB SNR	0.78	0.51	0.85	0.46
	Asymmetric HI No Noise	1.00	1.00	1.00	1.00
	HI 0dB SNR	0.98	0.99	0.98	0.91
	LI No Noise	0.82	0.59	0.88	0.56
	LI 0dB SNR	0.80	0.49	0.87	0.49
	Triangular HI No Noise	1.00	1.00	1.00	0.99
	HI 0dB SNR	0.99	1.00	0.99	0.96
	LI No Noise	0.79	0.49	0.87	0.46
	LI 0dB SNR	0.79	0.49	0.87	0.49

Precision

$$Precision[37] = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

4.3 Euclidean kNN Test Results

Testing feature extraction performance using techniques from Methods Chapter 3 with kNN begins with preprocessing of spectrograms into two cohorts with one less than 12 and the other more than 11 time intervals. Each test is repeated 30 times to reduce variability due to differences between testing cohorts[57]. Evidence of representative sample variability can be seen as low confidence interval numbers in the raw data found in Tables A.1, A.2, A.3, and A.4 as indicated by a 95% Confidence Interval value up to ±0.28 for High Time Interval kNN prediction with no noise. This reveals that the excellent results, all higher than 97.6%, are based on representative samples. Alternatively, a 95% Confidence Interval value as high as ±6.65 indicates that Low Time Interval 0dB SNR kNN samples displayed a much greater degree of variability and performance, even while implementing Monte Carlo Cross Validation described in Section 3.5.2. This is due to two factors; one is due to high detail variability seen in the frequency excursion of low time interval samples

TABLE 4.3: Hierarchical SVM Performance Metrics.

		Ground Truth	Accuracy	Sensitivity	Specificity	Precision
Hierarchical SVM	CW	HI No Noise	1.00	1.00	1.00	1.00
		HI 0dB SNR	0.98	0.92	1.00	1.00
		LI No Noise	0.96	1.00	0.95	0.83
		LI 0dB SNR	0.93	1.00	0.92	0.75
	Linear	HI No Noise	1.00	1.00	1.00	1.00
		HI 0dB SNR	1.00	1.00	1.00	0.99
		LI No Noise	0.87	0.67	0.92	0.68
		LI 0dB SNR	0.84	0.60	0.90	0.60
	Symmetric	HI No Noise	1.00	0.99	1.00	0.99
		HI 0dB SNR	0.98	0.97	0.99	0.95
		LI No Noise	0.82	0.57	0.88	0.54
		LI 0dB SNR	0.80	0.44	0.89	0.51
	Asymmetric	HI No Noise	1.00	1.00	1.00	1.00
		HI 0dB SNR	0.99	1.00	0.99	0.97
		LI No Noise	0.87	0.70	0.91	0.66
		LI 0dB SNR	0.85	0.66	0.90	0.63
	Triangular	HI No Noise	1.00	0.99	1.00	0.99
		HI 0dB SNR	0.99	0.98	0.99	0.97
		LI No Noise	0.83	0.41	0.93	0.60
		LI 0dB SNR	0.82	0.44	0.92	0.58

and the other is due to the low number of frequency peak shifts that can be exploited between time intervals, both illustrated in Table 4.1.

The least separability, in optimal conditions where there is both no noise and also high time intervals, was seen between FM Symmetric Up and FM Triangular Up. A lack of precision in FM Symmetric Up, as shown in Table 4.1, points to the feature extraction process failing to capture unique geometries separating it from FM Triangle Up. With the introduction of 0 dB SNR noise, precision remains consistent whereas FM Symmetric Up sensitivity falls, compared to FM Triangle Up, as shown in Table 4.1. This is due to noise influencing the geometry of other samples in the training set, making FM Symmetric samples less distinguishable. Sensitivity and Precision problems are compounded in samples with low time intervals; however, FM Linear Up displays a high degree of failure. Table 4.1 shows FM Linear Up Accuracy as being at least 80 percent, much like four out of five of the other samples. Where the sample fails the most is in Precision, where it scored 0.52 as shown in Table 4.1. This means features used to separate low time interval samples could not predict better than a little over half of relevant samples from all FM Linear Up samples in the testing cohort. Successes using this classifier include CW and FM Asymmetric Up, scoring 0.98 and 0.87 in Accuracy with 0dB SNR in low time interval sample cohort prediction, as shown in Table 4.1. Comparatively, CW Precision is 0.90 compared to FM Asymmetric Up at 0.69, indicating the robust separability for CW among the top two most separable samples in the low time interval cohort.

4.4 SVM Test Results

As defined in Methods Chapter 3, SVM is a binary classifier requiring training and testing between no more than two classes. Therefore, preprocessing before each trial requires first separating spectrograms into a low time interval cohort with less than 12 and a high cohort with more than 11 time intervals, and then consolidating samples in a way where the resulting prediction uses two classes.

4.4.1 One-versus-Rest SVM Test Results

FM Asymmetric Up performed the most poorly in the high time interval cohort with 0.91 Precision when under 0dB SNR, almost tied with second-to-last FM Symmetric Up Precision at 0.92, as shown in Table 4.2. Both of these samples have important separability features close to the beginning and end of their signal geometry which may be missed by the nine equally spaced feature markers interpolated in the feature extraction algorithm. The worst performing sample was FM Linear Up, with an Accuracy of 0.78 and Precision of 0.46, indicating an indiscriminate feature representation at low time interval and 0dB SNR, as shown in Table 4.2. This is not very surprising because the FM Linear Up geometry, being a straight diagonal line, is prone to distortion by the influence of noise and energy changes between frequency bins, as described in Section 3.3.1. CW is largely unperturbed by any of the previously described factors, with many results remaining in the 0.90s and dipping down to 0.83 under 0dB SNR at low time intervals; however, all other samples with low time intervals with 0dB SNR fail Precision metrics profoundly, with none achieving a score over 0.49, as shown in Table 4.2, pointing to a general lack of distinct separability of low time interval features in noisy conditions.

4.4.2 Hierarchical SVM Test Results

Under optimal conditions where time intervals are high, performance is high and consistent, with CW performing the lowest at 0.98 Accuracy, as shown in Table 4.3. Consistent high Accuracy under this classification method shows that feature separability is good when it is not confounded by superfluous features in the unbalanced negative training cohort, as shown in One-versus-Rest SVM. All metrics for high time interval samples are in the high 0.90s with the exception of CW Sensitivity at 0.92, as shown in Table 4.3, which is lowered due to CW being the first step in the hierarchical prediction tree. Unsurprisingly, high time interval samples with no noise for CW, FM Linear Up, and FM Asymmetric Up perform flawlessly across all performance metrics due to the classifier being able to scrutinize subtle distinguishing features once other classes are removed from both training and prediction sets. Surprisingly low performance is seen in low time interval 0dB SNR FM Triangle Up Sensitivity, at 0.41, as shown in Table 4.3. This is due to FM Triangle Up being misidentified during the sample symmetry prediction phase of the hierarchy. Some samples may have weak features exhibiting symmetry, preventing low time interval samples from passing into the detection phase in the hierarchy. FM Symmetric Up also suffers from this failure with a Sensitivity of 0.44 for 0dB and a slightly better score of 0.57 with no noise, as shown in Table 4.3. However, overall Accuracy for both of these classes remains somewhat good in the low 0.80s despite poor Sensitivity.

4.5 Analyses

Comparatively, kNN, One-versus-Rest SVM, and Hierarchical SVM all contained highest and lowest performing metrics in at least one category, as shown in Tables 4.1, 4.2, and 4.3. However, kNN yielded the greatest number of high performing metrics while One-versus-Rest SVM had the most poor performing ones. The poor performance of One-versus-Rest SVM among multiclass SVM classifiers is known[55] due to its unbalanced ratio of positive to negative samples. To illustrate this unbalance, in the tests for this thesis, 100 samples were used in a positive cohort while 400 samples are used to test "the rest" of the samples. This means that a much more detailed margin exists defining negative samples, making Precision work very well with high time interval samples with no noise due to the well defined negative margin, but very poor in all other cases where low time intervals and 0 dB make the well defined margin difficult to pass due to noise artifacts and interpolation of frequency peaks over low time interval samples. This led to a great deal of difficulty over all metrics in low time interval tests yet best results among high interval samples with no noise for FM Symmetric Up and FM Triangle Up. Improving feature extraction methods in this thesis to better suit One-versus-Rest SVM would require increasing geometric separability of FM Symmetric Up and FM Asymmetric Up from all other samples, perhaps by scrutinizing undulating features at the start and end of the signal.

kNN performed consistently well in all tests compared to the other two SVM methods, in part due to the importance of geometric proximity of the extracted features. Most noticeably, kNN performed best at predicting low time interval 0 dB signals with its poorest performance being FM Linear Up Precision 0.53, Accuracy 0.81, and Specificity 0.86, compared to Hierarchical SVM Precision 0.68, Accuracy 0.87, and Specificity 0.92. The metric that best hints why kNN performed so poorly with FM Linear Up and so well elsewhere is in the failure in Precision. In kNN testing the FM Linear Up sample, both in high and low time interval feature extraction, contains poor geometric feature separability compared to other samples, making it prone to being misidentified more frequently. This challenge became even greater with low time interval samples where some features describing peaks are heavily interpolated. With Hierarchical SVM, FM Linear Up only needed to be separated from FM Asymmetric Up, making their sparse features easier to separate.

Hierarchical SVM leverages CW separability to classify and remove it from further tests as it divides and classifies samples into equal cohorts to facilitate separability. This is most evident in its superior ability to classify FM Linear Up, beating all metrics in low time interval classification versus the two other classifiers. This indicates that when features are compared between two equally populated cohorts, features extracted using the methods described in this paper will produce good results. Alternatively, poor results seen in low time interval FM Symmetric Up and FM Asymmetric Up in both Hierarchical SVM and One-versus-Rest SVM in Sensitivity point to the Euclidean bias of the equidistantly interpolated feature extraction methods. Improving SVM performance may require introducing a normalization method and additional feature points that favor entering and exiting undulations seen in the frequency excursion of FM Asymmetric Up and FM Symmetric Up.

"Rouse him, and learn the principle of his activity or inactivity. Force him to reveal himself, so as to find out his vulnerable spots."

Sun Tzu, The Art of War

5 Conclusion

The aim of the thesis to develop classifier agnostic feature extraction is achieved, as evidenced by high accuracy and good precision using three classification approaches and two classifiers under noisy conditions. Contrasting and comparing performance between classifiers has provided additional insight into potential improvements that could be made to the feature extraction methods, as well as strategies to best implement these features using approaches designed to exploit them. The strong prediction accuracy made from the calculated mean values derived through kNN training coupled with a 95% Confidence Interval no greater than ±0.37 in high and ±6.86 in low time intervals reveals the robust geometrically separable characteristics of the feature extraction methods developed in this thesis.

5.1 Limitations

Low separability of the signal geometry made some samples prone to prediction problems using kNN as compared to SVM, as seen with FM Linear Up. In kNN, the mean values of its features

TABLE 5.1: High Interval, SVM, Symmetrical vs Asymmetrical, 95% CI.

Ground Truth	84dB SNR	Not	0dB SNR	Not
Linear	0.00	100.00	0.00	100.00
Symmetric	100.00	0.00	100.00	0.00
Asymmetric	0.00	100.00	0.00	100.00
Triangular	100.00	0.00	100.00	0.00

were compared to all other class features, whereas in Hierarchical SVM, they were compared only to FM Asymmetric Up. Performance will also improve with better interpolation methods used to mitigate the undulations seen in subsample interpolation, as mentioned in Section 3.3.1. Future work incorporating neural network learning methods can be used to interpolate the subsample peaks with greater accuracy than the parabolic interpolation used in this thesis.

5.2 Future Work

Theoretically, a hierarchical mixed-classifier approach could pair each classification task with the most appropriate classifier specifically trained to optimize prediction accuracy. This would improve performance without changing the feature extraction methods outlined in this thesis, with the exception of one step. First, kNN would be used to separate CW from non-CW samples, due to the high level of accuracy observed for separating both low and high interval CW samples from all others. Second, SVM would be used to separate geometrically Symmetric and Asymmetric samples by grouping FM Symmetric Up with FM Triangle Up and FM Linear Up with FM Asymmetric Up. Test results in Table 5.1 show that symmetrical geometry has a high degree of separability. Finally, SVM would be used to do high interval detection and a Convolutional Neural Network for differentiating one low interval sample from another, within previously predicted symmetrical and asymmetrical cohorts.

5.3 Conclusion

Hierarchical SVM metrics performed better than most One-versus-Rest SVM results; but even though it did not out-perform kNN, the approach shows promise. With results in both kNN and

One-versus-Rest SVM showing CW to be the most separable sample compared against all others, any classifier that exploits this advantage will see an improvement in separability of remaining classes after CW is excluded from successive training models. In the frequency domain, CW accuracy is improved with the selection of an odd number of feature points, allowing the central peak to be fully exploited by feature number 5, features 1-4 and 6-9 capturing the relatively flat geometry found in CW.

The new way of thinking about RER feature extraction, introduced by methods contained in this thesis, using classifier agnostic hierarchical methods for specialized feature extraction of high and low time interval spectrogram samples, proves that the methods are robust against SNR up to 0 dB, and introduces many options for future researchers to further explore this approach using hybrid classification.

A

Appendix

A.1 Raw Data Results

TABLE A.1: High Interval, kNN, no noise, 95% CI.

Ground Truth	CW	Linear	Symmetric	Asymmetric	Triangular
CW	100.00	0.00	0.00	0.00	0.00
Linear	0.00	100.00	0.00	0.00	0.00
Symmetric	0.00	0.00	98.61 ± 0.25	0.00	1.39 ± 0.25
Asymmetric	0.00	0.00	0.00	100.00	0.00
Triangular	0.00	0.00	2.15 ± 0.28	0.00	97.85 ± 0.28

TABLE A.2: High Interval, kNN, 0dB SNR, 95% CI.

Ground Truth	CW	Linear	Symmetric	Asymmetric	Triangular
CW	96.95 ± 0.35	0.33 ± 0.11	1.56 ± 0.22	0.66 ± 0.17	0.50 ± 0.14
Linear	0.00	99.72 ± 0.088	0.00	0.28 ± 0.088	0.00
Symmetric	0.00	0.00	96.63 ± 0.37	0.00	3.37 ± 0.37
Asymmetric	0.00	0.49 ± 0.09	0.00	99.51 ± 0.10	0.00
Triangular	0.00	0.00	2.37 ± 0.35	0.00	97.63 ± 0.35

TABLE A.3: Low Interval, kNN, no noise, 95% CI.

Truth	CW	Linear	Symmetric	Asymmetric	Triangular
CW	100 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Linear	8.67 ± 2.78	61 ± 6.86	7.33 ± 3.25	13.67 ± 4.15	9.33 ± 3.63
Symmetric	0 ± 0	16 ± 5.03	58.67 ± 6.76	9 ± 4.34	16.33 ± 5.27
Asymmetric	1.67 ± 1.36	20.67 ± 4.97	1.67 ± 1.36	70.33 ± 5.36	5.67 ± 2.6
Triangular	0 ± 0	17.67 ± 3.95	13.33 ± 4.13	6.33 ± 2.74	62.67 ± 5.63

TABLE A.4: Low Interval kNN, 0dB SNR, 95% CI.

Truth	CW	Linear	Symmetric	Asymmetric	Triangular
CW	100 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Linear	9.33 ± 2.65	57 ± 6.65	8.33 ± 3.77	17 ± 4.71	8.33 ± 2.98
Symmetric	0 ± 0	17 ± 4.22	59 ± 5.82	9 ± 3.02	15 ± 4.07
Asymmetric	2.33 ± 2.03	17.33 ± 5.23	7 ± 3.89	68.33 ± 6.91	5 ± 2.78
Triangular	0 ± 0	19 ± 3.68	14.33 ± 4.48	5.33 ± 1.82	61.33 ± 5.04

Table A.5: High Interval, SVM, CW vs Rest, 95% CI.

Truth	84dB SNR	Not	0dB SNR	Not
CW	100.00	0.00	93.80 ± 1.33	6.20 ± 1.33
Linear	0.00	100.00	0.00	100.00
Symmetric	0.00	100.00	0.30 ± 0.30	99.70 ± 0.30
Asymmetric	0.00	100.00	0.00	100.00
Triangular	0.00	100.00	0.00	100.00

Table A.6: High Interval, SVM, Symmetric vs Rest, 95% CI.

Ground Truth	84dB SNR	Not	0dB SNR	Not
CW	0.00	100.00	7.77 ± 0.94	92.23 ± 0.95
Linear	0.00	100.00	0.00	100.00
Symmetric	98.90 ± 0.21	1.10 ± 0.22	87.70 ± 1.41	12.30 ± 1.41
Asymmetric	0.00	100.00	0.00	100.00
Triangular	0.47 ± 0.31	99.53 ± 0.31	0.13 ± 0.12	99.87 ± 0.12

Table A.7: High Interval, SVM, Triangle vs Rest, 95% CI.

Ground Truth	84dB SNR	Not	0dB SNR	Not
CW	0.00	100.00	1.13 ± 0.34	98.87 ± 0.34
Linear	0.00	100.00	0.00	100.00
Symmetric	1.33 ± 0.36	98.67 ± 0.36	3.10 ± 0.61	96.90 ± 0.61
Asymmetric	0.00	100.00	0.00	100.00
Triangular	99.53 ± 0.20	0.47 ± 0.20	99.70 ± 0.17	0.30 ± 0.17

TABLE A.8: High Interval, SVM, Linear vs Rest, 95% CI.

Ground Truth	84dB SNR	Not	0dB SNR	Not
CW	0.00	100.00	1.03 ± 0.46	98.97 ± 0.46
Linear	100.00	0.00	100.00	0.00
Symmetric	0.00	100.00	0.00	100.00
Asymmetric	0.00	100.00	0.37 ± 0.18	99.63 ± 0.18
Triangular	0.00	100.00	0.00	100.00

TABLE A.9: High Interval, SVM, Asymmetric vs Rest, 95% CI.

Ground Truth	84dB SNR	Not	0dB SNR	Not
CW	0.00	100.00	9.50 ± 0.98	90.50 ± 0.98
Linear	0.47 ± 0.18	99.53 ± 0.18	0.03 ± 0.07	99.97 ± 0.07
Symmetric	0.00	100.00	0.00	100.00
Asymmetric	100.00	0.00	98.67 ± 0.32	1.33 ± 0.32
Triangular	0.00	100.00	0.00	100.00

TABLE A.10: High Interval, SVM, CW vs Rest, 95% CI.

Truth	84dB SNR	Not	0dB SNR	Not
CW	100.00	0.00	93.80 ± 1.33	6.20 ± 1.33
Linear	0.00	100.00	0.00	100.00
Symmetric	0.00	100.00	0.30 ± 0.30	99.70 ± 0.30
Asymmetric	0.00	100.00	0.00	100.00
Triangular	0.00	100.00	0.00	100.00

Table A.11: High Interval, SVM, Symmetrical vs Asymmetrical, 95% CI.

Ground Truth	84dB SNR	Not	0dB SNR	Not
Linear	0.00	100.00	0.00	100.00
Symmetric	100.00	0.00	100.00	0.00
Asymmetric	0.00	100.00	0.00	100.00
Triangular	100.00	0.00	100.00	0.00

Table A.12: High Interval, SVM, Symmetric vs Triangle, 95% CI.

Ground Truth	84dB SNR	Not	0dB SNR	Not
Symmetric	99.10 ± 0.24	0.90 ± 0.24	96.90 ± 0.69	3.10 ± 0.69
Triangular	1.13 ± 0.41	98.87 ± 0.41	1.83 ± 0.53	98.16 ± 0.53

Table A.13: High Interval, SVM, Linear vs Asymmetric, 95% CI.

Ground Truth	84dB SNR	Not	0dB SNR	Not
Linear	99.57 ± 0.18	0.43 ± 0.18	99.80 ± 0.15	0.20 ± 0.15
Asymmetric	0.00	100.00	0.50 ± 0.18	99.50 ± 0.18

Table A.14: High Interval, Hierarchical SVM, no noise, 95% CI.

Ground Truth	CW	Linear	Symmetric	Asymmetric	Triangular
CW	100 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Linear	0 ± 0	99.77 ± 0.15	0 ± 0	0.23 ± 0.15	0 ± 0
Symmetric	0 ± 0	0 ± 0	99 ± 0.25	0 ± 0	1 ± 0.25
Asymmetric	0 ± 0	0 ± 0	0 ± 0	100 ± 0	0 ± 0
Triangular	0 ± 0	0 ± 0	1.23 ± 0.5	0 ± 0	98.77 ± 0.5

Table A.15: High Interval, Hierarchical SVM, 0dB SNR, 95% CI.

Ground Truth	CW	Linear	Symmetric	Asymmetric	Triangular
CW	92.47 ± 0.78	0.77 ± 0.31	3.33 ± 0.64	2.23 ± 0.45	1.2 ± 0.42
Linear	0 ± 0	99.6 ± 0.18	0 ± 0	0.4 ± 0.18	0 ± 0
Symmetric	0.3 ± 0.17	0 ± 0	97.43 ± 0.62	0 ± 0	2.27 ± 0.58
Asymmetric	0 ± 0	0.33 ± 0.2	0 ± 0	99.67 ± 0.2	0 ± 0
Triangular	0 ± 0	0 ± 0	1.83 ± 0.47	0 ± 0	98.17 ± 0.47

Table A.16: Low Interval, SVM, CW vs Rest, 95% CI.

Truth	84dB SNR	Not	0dB SNR	Not
CW	100 ± 0	0 ± 0	100 ± 0	0 ± 0
Linear	1 ± 1.96	99 ± 1.96	3 ± 2.13	97 ± 2.13
Symmetric	2 ± 2.61	98 ± 2.61	4.33 ± 2.43	95.67 ± 2.43
Asymmetric	3 ± 4.18	97 ± 4.18	5 ± 2.93	95 ± 2.93
Triangular	8 ± 6.4	92 ± 6.4	8.33 ± 3.65	91.67 ± 3.65

Table A.17: Low Interval, SVM, Linear vs Rest, 95% CI.

Truth	84dB SNR	Not	0dB SNR	Not
CW	0 ± 0	100 ± 0	0 ± 0	100 ± 0
Linear	59 ± 7.42	41 ± 7.42	52.67 ± 8.24	47.33 ± 8.24
Symmetric	18 ± 10.85	82 ± 10.85	11.67 ± 4.42	88.33 ± 4.42
Asymmetric	16 ± 8.86	84 ± 8.86	35.33 ± 7.96	64.67 ± 7.96
Triangular	14 ± 10.21	86 ± 10.21	15 ± 4.68	85 ± 4.68

TABLE A.18: Low Interval, SVM, Symmetric vs Rest, 95% CI.

Truth	84dB SNR	Not	0dB SNR	Not
CW	0 ± 0	100 ± 0	0 ± 0	100 ± 0
Linear	10.33 ± 4.36	89.67 ± 4.36	7 ± 2.84	93 ± 2.84
Symmetric	55 ± 6.07	45 ± 6.07	51.33 ± 6.56	48.67 ± 6.56
Asymmetric	2.67 ± 1.86	97.33 ± 1.86	4.67 ± 2.61	95.33 ± 2.61
Triangular	38 ± 5.18	62 ± 5.18	48.33 ± 7.46	51.67 ± 7.46

TABLE A.19: Low Interval, SVM, Asymmetric vs Rest, 95% CI.

Truth	84dB SNR	Not	0dB SNR	Not
CW	0 ± 0	100 ± 0	0.33 ± 0.65	99.67 ± 0.65
Linear	23 ± 5.65	77 ± 5.65	36.33 ± 5.9	63.67 ± 5.9
Symmetric	10.33 ± 4.36	89.67 ± 4.36	5 ± 3.08	95 ± 3.08
Asymmetric	58.67 ± 6.76	41.33 ± 6.76	49.33 ± 7.34	50.67 ± 7.34
Triangular	13.33 ± 4.34	86.67 ± 4.34	9.33 ± 4.3	90.67 ± 4.3

TABLE A.20: Low Interval, SVM, Triangle vs Rest, 95% CI.

Truth	84dB SNR	Not	0dB SNR	Not
CW	0 ± 0	100 ± 0	0 ± 0	100 ± 0
Linear	7 ± 3.14	93 ± 3.14	9 ± 3.17	91 ± 3.17
Symmetric	33.67 ± 6.27	66.33 ± 6.27	33.67 ± 5.52	66.33 ± 5.52
Asymmetric	11.67 ± 4.89	88.33 ± 4.89	8.67 ± 3.73	91.33 ± 3.73
Triangular	45.33 ± 4.86	54.67 ± 4.86	48.67 ± 5.21	51.33 ± 5.21

TABLE A.21: Low Interval, Groupwise, SVM, CW vs Rest, 95% CI.

Truth	84dB SNR	Not	0dB SNR	Not
CW	100 ± 0	0 ± 0	100 ± 0	0 ± 0
Linear	1 ± 1.96	99 ± 1.96	3 ± 2.13	97 ± 2.13
Symmetric	2 ± 2.61	98 ± 2.61	4.33 ± 2.43	95.67 ± 2.43
Asymmetric	3 ± 4.18	97 ± 4.18	5 ± 2.93	95 ± 2.93
Triangular	8 ± 6.4	92 ± 6.4	8.33 ± 3.65	91.67 ± 3.65

TABLE A.22: Low Interval, Groupwise, SVM, Symmetrical vs Asymmetrical, 95% CI.

Truth	84dB SNR	Not	0dB SNR	Not
Linear	22.67 ± 5.71	77.33 ± 5.71	9.67 ± 4.46	90.33 ± 4.46
Symmetric	89 ± 4.24	11 ± 4.24	90 ± 2.97	10 ± 2.97
Asymmetric	8.67 ± 3.84	91.33 ± 3.84	8.67 ± 4.38	91.33 ± 4.38
Triangular	85.33 ± 3.84	14.67 ± 3.84	86.67 ± 3.8	13.33 ± 3.8

TABLE A.23: Low Interval, Groupwise, SVM, Symmetric vs Triangle, 95% CI.

Truth	84dB SNR	Not	0dB SNR	Not
Symmetric	63.33 ± 5.74	36.67 ± 5.74	53.33 ± 7.3	46.67 ± 7.3
Triangular	50.33 ± 6.13	49.67 ± 6.13	56 ± 6.13	44 ± 6.13

TABLE A.24: Low Interval, Groupwise, SVM, Linear vs Asymmetric, 95% CI.

Truth	84dB SNR	Not	0dB SNR	Not
Symmetric	72.33 ± 6.62	27.67 ± 6.62	61 ± 5.9	39 ± 5.9
Triangular	30 ± 5.72	70 ± 5.72	42.33 ± 5.92	57.67 ± 5.92

Table A.25: Low Interval, Hierarchical, SVM, no noise, 95% CI.

Truth	CW	Linear	Symmetric	Asymmetric	Triangular
CW	100 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Linear	1.67 ± 1.36	67 ± 6.45	13.33 ± 4.44	15.33 ± 5.29	2.67 ± 2.47
Symmetric	2.33 ± 1.8	11.67 ± 4.61	57.33 ± 7.51	8.33 ± 3.27	20.33 ± 8.33
Asymmetric	13 ± 3.78	7 ± 3.28	5.33 ± 2.78	70 ± 5.64	4.67 ± 3.22
Triangular	4 ± 2.22	13 ± 3.41	30.67 ± 5.14	11.67 ± 3.53	40.67 ± 5.63

Table A.26: Low Interval, Hierarchical, SVM, 0dB SNR, 95% CI.

Truth	CW	Linear	Symmetric	Asymmetric	Triangular
CW	100 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Linear	7.33 ± 3.38	60.33 ± 6.27	7.33 ± 3.63	20.67 ± 5.47	4.33 ± 2.03
Symmetric	9.67 ± 2.89	10.67 ± 4.3	44.33 ± 7.44	10 ± 3.26	25.33 ± 4.95
Asymmetric	10.33 ± 3.32	19.67 ± 4.65	1.67 ± 1.36	66 ± 5.45	2.33 ± 2.03
Triangular	5.33 ± 2.25	9.33 ± 2.81	33.33 ± 6.11	8 ± 3.17	44 ± 5.84

References

[1] Prokofiev, “Estimating the box-counting dimension of the coast of great britain,” 2010, [Online; accessed 1-October-2018]. [Online]. Available: https://commons.wikimedia.org/wiki/File:Great_Britain_Box.svg viii, 9

[2] Q. Fu, C. Li, F. Cai, W. Wang, and S. Xiao, “Emitter signal sorting based on fractal dimensions of pulse envelope’s front edge,” in *2016 International Conference on Military Communications and Information Systems (ICMCIS)*, May 2016, pp. 1–5. viii, 3, 10, 11

[3] G. Stimson, *Introduction to Airborne Radar*, ser. Aerospace & Radar Systems. SciTech Pub., 1998. [Online]. Available: https://books.google.com.au/books?id=RIxTAAAAMAAJ 1, 2, 6, 20

[4] M. I. Skolnik, *Radar Handbook, Third Edition*, ser. Electronics electrical engineering. McGraw-Hill Education, 2008. [Online]. Available: https://books.google.com.au/books?id=76uF2Xebm-gC 1, 2, 5, 6, 15, 39

[5] H. R. D., “Origins of radar,” *Eos, Transactions American Geophysical Union*, vol. 71, no. 27, pp. 781–786, 1990. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/90EO00226 1, 39

[6] N. G. E. Systems, “The radar warning story,” *Asian Defence Journal*, vol. V. 72, 1982. [Online]. Available: https://www.aef.se/Avionik/Artiklar/Motmedel/Nya_hotbilder/RadarWarnStory.pdf 2, 5, 6

[7] Q. Guo, P. Nan, X. Zhang, Y. Zhao, and J. Wan, “Recognition of radar emitter signals based on svd and af main ridge slice,” *Journal of Communications and Networks*, vol. 17, no. 5, pp. 491–498, Oct 2015. 2, 7, 12

[8] K. Li and T. Edwards, “Statistical models for automatic language identification,” in *ICASSP ’80. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, Apr 1980, pp. 884–887. 3

[9] S. Xu, L. Xu, Z. Xu, and B. Huang, “Individual radio transmitter identification based on spurious modulation characteristics of signal envelop,” in *MILCOM 2008 - 2008 IEEE Military Communications Conference*, Nov 2008, pp. 1–5. 3, 9, 10, 12

[10] C. Taowei, L. Chunhong, S. Li, and Y. Zhibing, “Feature extraction using autocorrelation function for radar emitter signals,” in *Proceedings of 2011 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference*, vol. 2, July 2011, pp. 1371–1374. 3

[11] S. W. Smith, *The Scientist and Engineer’s Guide to Digital Signal Processing*. San Diego, CA, USA: California Technical Publishing, 1997. 3, 15, 39

[12] O. Lowenschuss, “Electronic countermeasure system.” US Patent US 4 217 580A, 07 22, 1976. [Online]. Available: https://patents.google.com/patent/US4217580A/en 5

[13] D. J. Milojevic and B. M. Popovic, “Improved algorithm for the deinterleaving of radar pulses,” *IEE Proceedings F - Radar and Signal Processing*, vol. 139, no. 1, pp. 98–104, Feb 1992. 6

[14] P. M. Grant and J. H. Collins, “Introduction to electronic warfare,” *Communications, Radar and Signal Processing, IEE Proceedings F*, vol. 129, no. 3, pp. 113–132, June 1982. 6

[15] Y. M. Chen, C.-M. Lin, and C.-S. Hsueh, “Emitter identification of electronic intelligence system using type-2 fuzzy classifier,” *Systems Science & Control Engineering*, vol. 2, no. 1, pp. 389–397, 2014. [Online]. Available: https://doi.org/10.1080/21642583.2014.912569 6

[16] M.-Q. Ren, Y.-Q. Zhu, Y. Mao, and J. Han, “Radar emitter signals classification using kernel principle component analysis and fuzzy support vector machines,” in *2007 International Conference on Wavelet Analysis and Pattern Recognition*, vol. 3, Nov 2007, pp. 1442–1446. 8

[17] A. Kawalec and R. Owczarek, “Radar emitter recognition using intrapulse data,” in *15th International Conference on Microwaves, Radar and Wireless Communications (IEEE Cat. No.04EX824)*, vol. 2, May 2004, pp. 435–438 Vol.2. 8

[18] M. Giess, S. Sarafadeen, Q. He, and B. Tang, “A new radar recognition method based on the wigner trispectrum,” in *2014 7th International Congress on Image and Signal Processing*, Oct 2014, pp. 893–896. 8, 9

[19] J. Xie, “Robust intra-pulse modulation recognition via sparse representation,” in *2016 CIE International Conference on Radar (RADAR)*, Oct 2016, pp. 1–4. 9

[20] T. Vicsek, *Fractal Growth Phenomena*, 2nd ed. WORLD SCIENTIFIC, 1992. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/1407 9

[21] M. R. (Manfred Robert) Schroeder, *Fractals, chaos, power laws : minutes from an infinite paradise / Manfred Schroeder*. SERBIULA (sistema Librum 2.0), 10 2018. 9

[22] S. Xu, B. Huang, Z. Xu, and Y. Huang, “A new feature vector using local surrounding-line integral bispectra for identifying radio transmitters,” in *2007 9th International Symposium on Signal Processing and Its Applications*, Feb 2007, pp. 1–4. 10

[23] G. Zhang, L. Hu, and W. Jin, “Complexity feature extraction of radar emitter signals,” in *Asia-Pacific Conference on Environmental Electromagnetics, 2003. CEEM 2003. Proceedings.*, Nov 2003, pp. 495–498. 10

[24] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ser. ICCV ’99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1150–. [Online]. Available: http://dl.acm.org/citation.cfm?id=850924.851523 11

[25] S. Liu, X. Yan, P. Li, X. Hao, and K. Wang, “Radar emitter recognition based on sift position and scale features,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2018. 11

[26] P. Chen, G. Li, K. Xu, and J. Wan, "Applying the frechet distance to the specific emitter identification," in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, Nov 2016, pp. 1027–1030. 11

[27] R. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Heidelberg, Berlin, New York: Springer Verlag, 1998. 11

[28] *Speeding Up All-Pairwise Dynamic Time Warping Matrix Calculation*, 06 2016. 11

[29] J. Li and Y. Ying, "Radar signal recognition algorithm based on entropy theory," in *The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014)*, Nov 2014, pp. 718–723. 12

[30] J. W. Gibbs, *Elementary Principles in Statistical Mechanics: Developed with Especial Reference to the Rational Foundation of Thermodynamics*, ser. Cambridge Library Collection - Mathematics. Cambridge University Press, 2010. 12

[31] X. Zhang, P. Luo, and X. Hu, "A hybrid method for classification and identification of emitter signals," in *2017 4th International Conference on Systems and Informatics (ICSAI)*, Nov 2017, pp. 1060–1065. 12

[32] X. Ru, C. Gao, Z. Liu, Z. Huang, and W. Jiang, "Emitter identification based on the structure of unintentional modulation," in *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Oct 2016, pp. 998–1002. 12

[33] S. D'Agostino, "Specific emitter identification based on amplitude features," in *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, Oct 2015, pp. 350–354. 12

[34] S. D'Agostino, G. Foglia, and D. Pistoia, "Specific emitter identification: Analysis on real radar signal data," in *2009 European Radar Conference (EuRAD)*, Sept 2009, pp. 242–245. 12

[35] J. Samuel and W. du Plessis, "Specific emitter identification for enhanced access control security," *SAIEE Africa Research Journal*, vol. 108, pp. 71 – 79, 06

2017. [Online]. Available: http://www.scielo.org.za/scielo.php?script=sci_arttext&pid=S1991-16962017000200003&nrm=iso 12

[36] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, Heidelberg, Dordrecht, London: Springer, 2013. [Online]. Available: https://dl.dropboxusercontent.com/u/108263707/_book/KuhnJohnson2013apm.pdf 14

[37] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. 14, 41, 42, 43

[38] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000. 14, 35, 36, 37

[39] *Time-Frequency Analysis of Radar Signals*, 03 2012. 15

[40] A. V. Oppenheim, "Speech spectrograms using the fast fourier transform," *IEEE Spectrum*, vol. 7, no. 8, pp. 57–62, Aug 1970. 15

[41] B. Audone, R. Colombo, I. Marziali, and O. Losito, "The short time fourier transform and the spectrograms to characterize emi emissions," in *2016 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, Sept 2016, pp. 882–888. 15

[42] M. Komorowski, D. Marshall, J. Salciccioli, and Y. Crutain, *Exploratory Data Analysis*. Springer, Cham, 09 2016. 18

[43] R. Bellman, *Dynamic Programming*, ser. Dover Books on Computer Science Series. Dover Publications, 2003. [Online]. Available: https://books.google.com.au/books?id=fyVtp3EMxasC 18

[44] P. F. Evangelista, M. J. Embrechts, and B. K. Szymanski, "Taming the curse of dimensionality in kernels and novelty detection," in *Applied Soft Computing Technologies: The Challenge of Complexity*, A. Abraham, B. de Baets, M. Köppen, and B. Nickolay, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 425–438. 18

[45] I. J. SCHOENBERG, "Contributions to the problem of approximation of equidistant data by analytic functions: Part a.âĂŤon the problem of smoothing or graduation. a first class of

analytic approximation formulae," *Quarterly of Applied Mathematics*, vol. 4, no. 1, pp. 45–99, 1946. [Online]. Available: http://www.jstor.org/stable/43633538 24

[46] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, January 1967. 33

[47] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: https://doi.org/10.1023/A:1022627411411 33

[48] Q.-S. Xu, Y.-Z. Liang, and Y.-P. Du, "Monte carlo cross-validation for selecting a model and estimating the prediction error in multivariate calibration," *Journal of Chemometrics*, vol. 18, no. 2, pp. 112–120, 2004. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.858 34

[49] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992. [Online]. Available: http://www.jstor.org/stable/2685209 35

[50] L. Li, H. Ji, and L. Wang, "Specific radar emitter recognition based on wavelet packet transform and probabilistic svm," in *2009 International Conference on Information and Automation*, June 2009, pp. 1308–1313. 36

[51] G. Zhang, W. Jin, and L. Hu, "Radar emitter signal recognition based on support vector machines," in *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.*, vol. 2, Dec 2004, pp. 826–831 Vol. 2. 36

[52] Y. Sun, X. Hu, S. Yin, and J. Liu, "A new method of electromagnetic radiant sources based on support vector machine," in *2016 Progress in Electromagnetic Research Symposium (PIERS)*, Aug 2016, pp. 3060–3063. 36

[53] L. Li, H. Ji, and L. Wang, "Specific radar emitter recognition based on wavelet packet transform and probabilistic svm," in *2009 International Conference on Information and Automation*, June 2009, pp. 1308–1313. 36

[54] R. Bellman and R. Bellman, *Adaptive Control Processes: A Guided Tour*, ser. Princeton Legacy Library. Princeton University Press, 1961. [Online]. Available: https://books.google.com.au/books?id=POAmAAAAMAAJ 36

[55] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. 37, 40, 46

[56] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. 41

[57] R. Hogg and E. Tanis, *Probability and Statistical Inference*. Pearson Prentice Hall, 2006. [Online]. Available: https://books.google.com.au/books?id=MKl1AAAACAAJ 43