Application of Dual-Tree ℂomplex Wavelet Transforms to Burst Detection and RF Fingerprint Classification

DISSERTATION

Randall W. Klein, Major, USAF

AFIT/DEE/ENG/09-12

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/DEE/ENG/09-12

Application of Dual-Tree ℂomplex Wavelet Transforms to Burst Detection and RF Fingerprint Classification

DISSERTATION

Presented to the Faculty
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

Randall W. Klein, B.S.E.E., M.S.E.E.
Major, USAF

September 2009

AFIT/DEE/ENG/09-12

Application of Dual-Tree Complex Wavelet Transforms to Burst Detection and RF Fingerprint Classification

Randall W. Klein, B.S.E.E., M.S.E.E.

Major, USAF

Approved:

	18 Aug Ø9
Dr. Michael A. Temple Chairman	Date
	18-AUG-2009
Maj. Michael J. Mendenhall, PhD Member	Date
	18 Aug 2009
Dr. Richard G. Cobb Member	Date

Accepted:

MU Thomas	24 Aug 2009
M. U. Thomas	Date

Dean, Graduate School of Engineering and Management

AFIT/DEE/ENG/09-12

Abstract

The continued proliferation of affordable RF communication devices has greatly increased wireless user exposure and the need for improved security to protect against spoofing. This work addresses various Open Systems Interconnection (OSI) Physical (PHY) layer mechanisms to extract and exploit RF waveform features ("fingerprints") that are inherently unique to specific devices and that may be used for reliable device classification to provide hardware specific identification (manufacturer, model, and/or serial number). Automatically detecting, identifying and locating RF communication devices remains a challenging technical problem and consists of: 1) the selection and generation of fundamental signal characteristics (amplitude, phase, and/or frequency), 2) the feasibility and repeatability of detecting and locating the start of a burst using selected waveform feature(s) amidst channel noise, 3) the identification and robust extraction of distinguishable fingerprints–features that uniquely characterize the unintentional modulation of a device, and 4) the performance of signal classification under varying channel conditions and Signal-to-Noise Ratio (SNR). This challenge is addressed by applying a Dual-Tree $\mathbb{C}$omplex Wavelet Transform (DT-$\mathbb{C}$WT) to improve burst detection and RF fingerprint classification.

Two burst detection techniques are analyzed under varying channel SNR conditions, the Fractal Bayesian Step Change Detector (Fractal-BSCD) and Traditional Variance Trajectory (VT). Performance of both techniques are consistent with perfect burst location at higher SNRs ($10 \leq SNR \leq 30$ dB) but diverged at lower SNRs ($-3 \leq SNR \leq 10$ dB). Traditional VT performance is most consistent with perfect results for $6 \leq SNR \leq 30$ dB, under performs perfect results for $-3 \leq SNR \leq 6$ dB, and outperforms Fractal-BSCD considerably for $-3 \leq SNR \leq 18$ dB. A "Denoised VT" technique is introduced to improve performance at lower SNRs, with denoising implemented using a DT-$\mathbb{C}$WT decomposition prior to Traditional VT processing. This proves to be effective and provides more robust burst detection for $-3 \leq SNR \leq 10$ dB.

Performance of a newly developed Wavelet Domain (WD) fingerprinting technique is presented using statistical WD fingerprints with Multiple Discriminant Analysis/Maximum Likelihood (MDA/ML) classification. The statistical fingerprint features are extracted from coefficients of a DT-$\mathbb{C}$WT decomposition. Relative to previous Time Domain (TD) results, the enhanced WD statistical features provide improved device classification performance. Improvement is characterized using a "gain" metric defined as the difference in required SNR in dB ($SNR_{WD} - SNR_{TD}$) for the two techniques to achieve a given classification performance. Accounting for all intra-manufacturer and inter-manufacturer device discrimination scenarios, the WD technique provides 2–7 dB of gain for 80% correct classification performance at 2 dB $< SNR_{WD} <$ 11 dB. Additional performance sensitivity results are presented to demonstrate WD fingerprinting robustness for variation in burst location error, MDA/ML training and classification SNRs, and MDA/ML training and classification signal types. For all cases considered, the WD technique proves to be more robust and exhibited less sensitivity when compared with the TD technique.

Acknowledgements

I would like to first and foremost thank my family for all the love and support they provided me. Without them, this journey would have been all but impossible and not nearly as fun.

Most students would agree that choosing the right advisor is one of the most important decisions you'll make. I would like to thank Dr. Temple for being my advisor, again, and for all the right guidance and direction he provided. I know I made the right decision and would do it all over again.

I would also like to thank my committee members whose assistance and encouragement was greatly appreciated.

I would also like to thank my fellow students whose numerous nuggets of information always proved timely and helpful.

Lastly, I would to thank the US Air Force for granting me the time and opportunity to pursue this degree.

Randall W. Klein

Table of Contents

List of Figures

Figure Page

Figure Page

List of Tables

List of Abbreviations

Abbreviation		Page
RF	Radio Frequency	1
OSI	Open Systems Interconnection	2
MAC	Medium Access Control (OSI Layer)	2
PHY	Physical (OSI Layer)	2
RSS	Received Signal Strength	2
SEI	Specific Emitter Identification	3
SNR	Signal-to-Noise Ratio	4
OFDM	Orthogonal Frequency Division Multiplexing	4
SDR	Software Defined Radio	5
CR	Cognitive Radio	5
DWT	Discrete Wavelet Transform	5
AWGN	Additive White Gaussian Noise	5
STFT	Short Time Fourier Transform	5
FFT	Fast Fourier Transform	5
DT-ℂWT	Dual-Tree ℂomplex Wavelet Transform	6
WD	Wavelet Domain	8
TD	Time Domain	8
MDA/ML	Multiple Discriminant Analysis/Maximum Likelihood	10
BSCD	Bayesian Step Change Detector	10
VT	Variance Trajectory	10
RFSICS	RF Signal Intercept and Collection System	12
PDF	Probability Distribution Function	15
GRLVQI	Gen Relevance Learning Vec Quant Improved	75
WEKA	Waikato Environment for Knowledge Analysis	75

Application of Dual-Tree ℂomplex Wavelet Transforms to Burst Detection and RF Fingerprint Classification

I. Introduction

This chapter introduces the dissertation research and its documentation. The motivation for conducting the research is first provided in Section 1.1 which includes the Operational Motivation factors in Section 1.1.1 and Technical Motivation factors in Section 1.1.2. This is followed by a summary of Research Contributions in Section 1.2 which provides a relational mapping in Table 1.1 to highlight contributions of this work relative to what had been previously accomplished. The chapter concludes with a Dissertation Overview in Section 1.3.

1.1 Research Motivation

1.1.1 Operational Motivation. The continued proliferation of inexpensive wireless Radio Frequency (RF) devices provides worldwide communication connectivity to virtually every individual. Within a geographically localized region, the fundamental emissions from these devices, i.e., the intentionally radiated emissions designed to support the intended purpose, may be remotely intercepted by unintended recipients. The intended communicators are generally unaware that this is occurring and the intent of the unauthorized listener varies. The interceptor may remain passive and simply "listen" with the intent of monitoring, recording, analyzing, etc., the communication activity. This type of passive activity is very difficult to detect. In other cases, the interceptor may become active and "join" in the communication activity. This may take the form of "spoofing" or "man-in-the-middle" type attacks whereby an identity compromise occurs and the unintended party is able to freely inject traffic into the system. This activity is generally detectable given that interceptor RF emissions are present. To mitigate this activity, there is a pressing need to improve both pre-attack security and post-attack digital forensics.

1.1.2 Technical Motivation.

1.1.2.1 PHY Layer Network Security. Much research has focused on traditional bit-level algorithmic approaches to improve network security and mitigate spoofing. More recently consideration has been given to detecting and mitigating spoofing near or at the bottom of the Open Systems Interconnection (OSI) network stack. One such work includes the addition of a "lightweight security layer" hosted within the Medium Access Control (MAC) layer to detect spoofing and anomalous traffic [35]. Other recent efforts have focused on Physical (PHY) layer implementations with a goal of exploiting RF characteristics (radio and environmental) that are difficult to mimic, thus minimizing the opportunity for spoofing. Two such efforts investigated the use of Received Signal Strength (RSS) as a means for detecting the presence of a spoofing node [5, 53]. Although related in their use of RSS, it is not entirely clear that these works are comparable one-to-one given that the experiments were conducted using different hardware being operated in different physical environments. Under dissimilar conditions such as these, it is expected that statistics of the power-based RSS metric would vary. This variation is not unique to wireless communications and is encountered in other applications employing power-based metrics, especially when all system and environmental interactions are accounted for (antenna patterns, multipath, background noise, etc.).

The authors in [53] introduce RF fingerprinting in [24] as an alternative technique to detect and mitigate spoofing through PHY layer mechanisms. However, they readily dismiss this alternative for "scale" reasons. Assuming this conclusion is based on the fact that RSS is currently supported and provided with most manufactured devices, the authors' position is supportable. This is particularly true when constraining PHY layer anti-spoofing mechanisms to reside on PC-sized cards. However, there may be applications where the size constraints are much more relaxed and RF fingerprinting becomes a viable alternative. Consistent with related work in [54, 55], these applications were addressed through the fundamental research goal that involved demonstrating radar-like Specific Emitter Identification (SEI) capability similar to what is used to distinguish between radar emitters [6,9,11,12,34,40,46,60].

1.1.2.2 Specific Emitter Identification. Radar-based SEI research spans nearly twenty years and has considered both conventional and non-conventional parameters for identification. Conventional radar parameters generally include those which are based on *intentional* modulation which may be applied across multiple pulses (inter-pulse modulation) or within a given pulse (intra-pulse modulation). These modulations are introduced to improve some aspect of overall radar performance (tracking accuracy, ambiguity resolution, clutter suppression, etc.). There are other *unintentional* modulations that may be induced by the hardware used to implement the system [30, 34]. These unintentional modulations may result from any number of hardware issues, including poor system design (device incompatibility), improper operation (over/under voltage), physical device limitations (operating temperature range), etc. When viewed at the waveform level, many of these features are similar to what currently exist in modern wireless communication systems that typically transmit burst-like waveforms representing various forms of digital information (symbols, bits, packets, etc.). Communication researchers have recognized these similarities and have begun to address the question: **“Can existing SEI methods be employed with wireless communication signals to achieve radar-like SEI capability?”**

1.1.2.3 RF Fingerprinting. The task of automatically detecting, identifying and locating commercial RF communication devices remains a challenging technical problem. The work presented here addresses four main aspects of this problem, including: 1) the selection and generation of fundamental signal characteristics (amplitude, phase, and/or frequency), 2) the feasibility and repeatability of detecting and locating the start of a burst using selected waveform feature(s) amidst channel noise, 3) the identification and robust extraction of distinguishable fingerprints–features that uniquely characterize the unintentional modulation of a device, and 4) the performance of signal classification under varying channel conditions and Signal-to-Noise Ratio (SNR). The ultimate goal is to demonstrate an end-to-end process to accurately classify commercially-available RF communication devices using signal features extracted from collected emissions. Relevant research in wireless network security and RF fingerprinting suggests that information in fundamental emissions and unintentionally modulated regions provides the most effective means for identifying transmitters. Collectively, related works in RF fingerprinting, electromagnetic signatures, intrapulse modulation, and unintentional modulation [11, 23, 24, 30, 34, 51, 64, 66, 68], form a solid basis for developing techniques that may be applicable to commercial communication devices. If the inherent RF fingerprints are repeatedly extractable and sufficiently unique, they are potentially useful for determining the specific make, model, and/or serial number of a given device.

Previous work highlighted signal structure uniqueness and attributed inter-device differences to various manufacturing, aging, and environmental factors [68]. While several processing steps are required to effectively exploit the unique RF fingerprints, burst location is arguably the most important [23,66]. In this context, burst location includes determining both the burst start time and the subsequent signal region(s) from which fingerprints are extracted. Burst detection, burst start location and signal region(s) selection for fingerprint extraction are all important given that improper determination of the burst start location and imprudent selection the signal region(s) can adversely bias processing in favor of channel noise or undesired signal features [68].

The most relevant published results to date for this research are found in [54,55] and is based on experimentally collected 802.11A signals. As with this previous work, the choice of using Orthogonal Frequency Division Multiplexing (OFDM)-based signals for RF fingerprinting demonstration was driven by two factors: 1) consistency with previous related 802.11A work that has been extensively published [42, 54, 55, 63, 67], and 2) the continued emergence of OFDM-based signals as envisioned for 4G Software Defined Radio (SDR) and Cognitive Radio (CR) communications [21,26,48, 72]. While the fundamental fingerprinting and classification techniques in this work are believed to be broadly applicable to other signal types, the challenges posed by OFDM-based signals is of near-term interest.

1.1.2.4 Signal Denoising. In some applications the desired level of performance cannot be achieved due to inherent noise contributions in the environment. The effective mitigation of such adverse noise effects has been demonstrated in numerous applications by "denoising" the signal of interest prior to processing to remove undesired noise contributions. This can be accomplished using a Discrete Wavelet Transform (DWT) by exploiting differences in the distribution of signal burst energy and the Additive White Gaussian Noise (AWGN) in which it is embedded [4, 7, 8, 14, 15, 17–19, 44, 61]. The common approach to wavelet denoising includes: 1) transforming the input signal with the desired transform,2) comparing coefficient magnitudes with a pre-defined threshold, 3) zeroing-out all coefficients having magnitudes less than the threshold while retaining those above the threshold, 4) inverse transforming the thresholded set of coefficients, and 5) processing the resultant denoised signal.

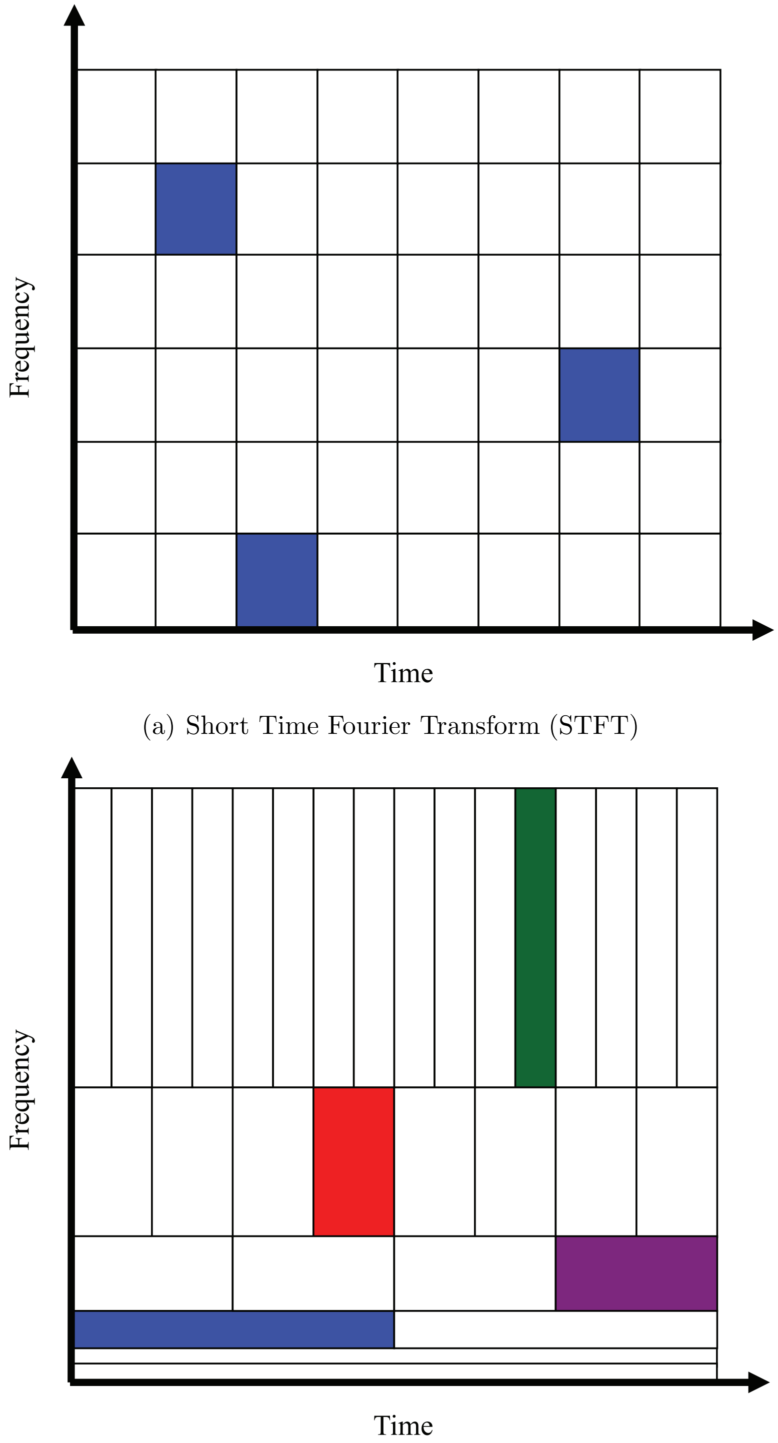
One distinct disadvantage of the DWT is the lack of shift invariance. i.e., for a given time shift in the input signal the transformation yields a different set of coefficients. For burst detection, this problem has the consequence of complicating the computation of reasonable thresholds for signal denoising. One shift invariant (when properly implemented) alternative is the Short Time Fourier Transform (STFT) [13]. The STFT is a Fast Fourier Transform (FFT) done over a series of short contiguous time intervals spanning the signal of interest [47]. Ideally, the intervals are short enough to maintain piece-wise stationarity across the signal while at the same time long enough to capture sufficient spectral energy. This trade-off represents a compromise between achievable time resolution (better with a shorter interval) and achievable frequency resolution (better with a longer interval)–the Heisenberg inequality [44]. One drawback is that for a given STFT interval length, which generally remains fixed throughout the signal duration, the resolution in both time and frequency is uniform across the domains. This is illustrated in Figure 1.1(a) using Heisenberg uncertainty boxes.

The DWT achieves non-uniform multi-resolution capability by effectively scaling the time interval inversely proportional to frequency such that a relatively narrow interval is used to capture high frequency content. This is illustrated in Figure 1.1(b) which shows representative Heisenberg uncertainty boxes for an arbitrary DWT. As indicated, the higher frequency content regions have higher time resolution (narrower box widths across time) and lower frequency content regions have lower time resolution (wider box widths across time). This type of multi-resolution time-frequency decomposition works best if the signal is composed of high frequency components of short duration plus low frequency components of long duration, characteristics which most signals possess [47].

An alternative wavelet transform that possesses both the DWT's multi-resolution capability and the STFT's shift invariance is the Dual-Tree $\mathbb{C}$omplex Wavelet Transform (DT-$\mathbb{C}$WT) [2, 50]. The DT-$\mathbb{C}$WT is a DWT extension that is "nearly shift-invariant," i.e., the DT-$\mathbb{C}$WT coefficients are independent of time domain shift and more strongly dependent on inter-scale and intra-scale neighborhoods [50]. Furthermore, the DT-$\mathbb{C}$WT magnitude response exhibits reduced ringing in the wavelet domain due to high-frequency noise and sharp discontinuities, which makes the denoising process more reliable by ensuring consistent threshold calculations [50].

1.2 Research Contributions

Table 1.1 provides a list of various *Technical Areas* (concepts, techniques, attributes, metrics, etc.) and the relational mapping between *Previous* related work and the *Current* research presented in this dissertation. As summarized in the fol-



(a) Short Time Fourier Transform (STFT)

(b) Discrete Wavelet Transform (DWT)

Figure 1.1: Tiling of Heisenberg uncertainty boxes in the time-frequency plane for STFT and DWT decompositions. The width and height of a given box is related to its time and frequency resolution, respectively [47].

lowing subsections, there have been contributions made to each of the technical areas identified in the first five rows of the table.

1.2.1 Performance Criteria. Experimental setup and execution can differ from one research activity to another, even when using identical or similar equipment and processes. This can make direct comparison of new results with previous results difficult and care must be taken to ensure that 1) previous contributions are fairly represented and 2) new contributions are sufficiently supported–this is the case for work presented here. For all previous AFIT-based works referenced in Table 1.1, there were no firm performance goals or criteria in place at the time the work was conducted. Rather, proof-of-concept demonstration was the main objective and "As Achieved" performance was reported as noted in Table 1.1.

While as achieved results are presented in this document as well, and based on many combinations of parameters and parameter values, specific performance criteria was introduced to help highlight performance differences (poorer and better) across the numerous scenarios considered. The "Reasonable" criteria used here and shown in Table 1.1 is somewhat arbitrary and based on achieving 80% or better classification accuracy at $SNR \leq 20$ dB. Using this reasonable operating point of 80% classification accuracy, performance comparisons are made throughout Chapter 4 based on the "gain" provided by Wavelet Domain (WD) techniques relative to what is provided by Time Domain (TD) techniques. This gain is defined here as the reduction in required SNR, in dB, for the WD fingerprinting technique to achieve the same classification performance as the TD fingerprinting technique.

1.2.2 TD Fingerprint Classification. Prior to assessing WD classification performance, it was necessary to replicate TD results in [54] to form a baseline for comparison. Upon replicating these earlier results, it became evident that the post-collection filter bandwidth BW_{PC} was a very important parameter and that all earlier TD results were based on using a fixed value. While the fixed bandwidth approach was sound and the selected bandwidth was reasonably based on sound engineering prac-

Table 1.1: Relational mapping between *Technical Areas* in *Previous* related work and *Current* research contributions.

Technical Area	Previous		Current	
	Addressed	Ref #	Addressed	Ref #
TD Fingerprinting	×	[23, 24, 54, 55, 68]	×	[31–33]
WD Fingerprinting			×	[32]
SNR Sensitivity	×	[54, 55]	×	[31–33]
Burst Detection			×	[31, 33]
Signal Type / Modulation				
802.11A / OFDM	×	[20, 42, 54, 55, 63, 67]	×	[31–33]
802.11B / DSSS	×	[24, 66, 68]		
802.11G / OFDM			×	
GSM / GMSK	×	[3]		
Bluetooth / GFSK	×	[23, 71]		
Instantaneous Signal Characteristics				
Amplitude	×	[20, 24, 42, 51] [63, 64, 66–68]	×	[31–33]
Phase	×	[20, 23, 24]	×	[31–33]
Frequency	×	[20, 24]	×	[31–33]
RF Fingerprint Features and Metrics				
Std Deviation	×	[20, 24]		
Variance	×	[20, 23]	×	[31–33]
Skewness	×	[54]	×	[31–33]
Kurtosis	×	[54]	×	[31–33]
Classification Method and Performance Criteria				
Bayesian MDA/ML	×	[20, 54]	×	[31–33]
"As Achieved"	×	[23, 24, 51, 54, 68]		
"Reasonable"			×	[32]

tices, the earlier works provided no bandwidth sensitivity analysis. This analysis was subsequently carried out under this research and a bandwidth of $BW_{PC} = 7.7$ MHz was used for generating all comparative TD and WD results. This particular value enabled comparison of both techniques at their best overall performance levels, with TD having an approximate 2% advantage in device classification at higher SNRs–an advantage that rapidly diminishes at lower SNRs that are more consistent with operational environments [31–33].

1.2.3 WD Fingerprint Classification. Relative to TD fingerprint classification, enhanced fingerprint classification is demonstrated here using improved fingerprint features. Specifically, this work represents the first application of a DT-$\mathbb{C}$WT decomposition to enhance features of statistical RF fingerprints. Considerable performance improvement or gain is realized using the enhanced WD feature set with identical post-collection filter bandwidth and Multiple Discriminant Analysis/Maximum Likelihood (MDA/ML) processing [32].

1.2.4 SNR Sensitivity Analysis. With the exception of results generated under this research and documented in [31–33], a majority of the works cited in Section 1.1.2 lack any form of sensitivity analysis in terms of assessing burst detection and/or fingerprint classification performance under varying channel noise conditions. The two exceptions are the most recent related works in [54, 55]. Noise sensitivity analysis is imperative for determining acceptable SNR levels for achieving consistent and reliable classification results. Classification sensitivity to channel noise and burst-to-burst detection variability has been analyzed using experimentally collected 802.11A signals in [33]. With respect to burst location estimation, both Fractal-Bayesian Step Change Detector (BSCD) and Traditional Variance Trajectory (VT) techniques provided results that were consistent with "perfect" burst location (a start location based on visual inspection of each collected burst) at higher SNRs ($10 \leq SNR \leq 30$ dB). However, performance for both techniques diverged at lower SNRs ($-3 \leq SNR \leq 10$ dB) [33]. With respect to the burst location estimation error impact to classification performance, the Traditional VT technique was consistent with perfect estimation for $6 \leq SNR \leq 30$ dB but underperformed for $-3 \leq SNR \leq 6$ dB. Traditional VT also provided considerable improvement when compared with the Fractal-BSCD technique at lower $SNRs$ ($-3 \leq SNR \leq 18$ dB), i.e., for a given classification accuracy in the range of 50%–80% the required SNR for Traditional VT is 3-6 dB lower than what is required for Fractal-BSCD. This shortfall provided an impetus for subsequent burst detection research aimed at improved performance at low SNRs [31].

1.2.5 Burst Detection at Lower SNR. As published in [31] and presented in this dissertation, signal denoising with the DT-$\mathbb{C}$WT prior to Traditional VT burst detection (introduced here as Denoised VT processing) is more effective and provides more robust burst detection and location at lower SNRs ($-3 \leq SNR \leq 10$ dB). Relative to results for perfect burst detection and location, the Denoised VT process achieves nearly 34% of the available performance improvement–when used with MDA/ML processing, there is little more to be gained in overall classification performance by improving burst detection and location accuracy [31].

1.3 Dissertation Overview

This document is divided into five chapters and contains one appendix. Chapter 2 presents relevant technical background information on major concepts and techniques used to conduct the research. Sufficient technical detail is presented such that the fundamental research approach is repeatable and the key contributions are verifiable. The major concepts and techniques are presented as functionally implemented in the overall demonstration process.

Chapter 3 provides the overall demonstration process used for generating results and conducting analysis. A detailed description is included for both the "Signal Collection" hardware and "Post-Collection Processing" software processes. The primary hardware used for signal collection was AFIT's RF Signal Intercept and Collection

System (RFSICS) with subsequent data processing accomplished exclusively in a MATLAB® environment.

Chapter 4 provides modeling, simulation and analysis results that were generated using the processes detailed in Chapter 3. The research involved hundreds of simulations, each requiring tens of hours of processing time in some cases. For brevity and to ensure succinctness, only a subset of representative results are presented from selected scenarios to fully support key research findings and contributions.

Chapter 5 concludes the main document by providing an overall summary of research activities, a summary of key findings, and recommendations for subsequent research. This is followed by an appendix that provides some of the developmental MATLAB® code used to support the research.

II. Background

This chapter presents relevant technical background information on major concepts and techniques used to conduct the research. The material here supports subsequent material presented in the methodology, results and conclusion chapters of the document. This chapter is not presented as a complete tutorial, but rather, intended to provide sufficient detail such that the fundamental research approach is repeatable and the key contributions are verifiable. For convenience, the major concepts and techniques are presented as functionally implemented in the overall demonstration process. Burst Detection and Location is first presented in Section 2.1 which provides details on the two specific techniques considered, including the Fractal-Bayesian Step Change Detector (Fractal-BSCD) in Section 2.1.1 and the Traditional Variance Trajectory (Traditional VT) technique in Section 2.1.2. Lastly, the Dual-Tree $\mathbb{C}$omplex Wavelet Transform (DT-$\mathbb{C}$WT) is presented in Section 2.3.

2.1 Burst Detection and Location

Discriminating a burst-like signal response from background noise can be a difficult task as many burst responses can appear noise-like. In some respects, detecting a burst response is akin to separating noise from noise [52]. Related research has focused on exploiting two different properties to discriminate between signal and background noise contributions, including *inherent signal structure* and *instantaneous signal characteristics*. Inherent signal structure has been successfully exploited using a Fractal-Bayesian Step Change Detector (Fractal-BSCD) while instantaneous signal characteristics have been exploited using Traditional Variance Trajectory (VT). The details for these approaches are provided in the following subsections.

2.1.1 Fractal-Bayesian Step Change Detector.

The Fractal-Bayesian Step Change Detector (Fractal-BSCD) has been used to exploit inherent signal structure to discriminate between signal and channel background responses. As time progresses, random Additive White Gaussian Noise (AWGN) exhibits no structure amongst samples while a deterministic signal does. When a time series transitions from a region containing only noise to a region containing both noise and signal its inherent structure changes. This change can be detected using fractal dimensions. However, a burst response in such a region is non-stationary and therefore, not a pure fractal, i.e., its fractality is a function of time and thus it cannot be self-similar [52]. Otherwise, the calculated fractal dimension would yield the same value regardless of the signal time and duration used, which does not describe a non-stationary signal. Yet on a smaller scale, a transient can have local stationary fractality and can be modeled as a series of piece-wise fractals through multi-fractality analysis. The local fractal dimensions are calculated using a sliding window [52].

It has been demonstrated that burst start location can be accomplished using the fractal dimension [64] measure followed by a Bayesian Step Change Detector [42, 63, 64, 67]. This process is denoted here as Fractal-BSCD. The fractal derivation can be found in [27] and can be calculated using the following Higuchi method. Given a windowed data time series $\{X(1),\ X(2),\ ...,\ X(N_x)\}$, the curve length is defined as:

$$L_m(k) = \frac{\bar{X}(N_x - 1)}{k^2 N_L} \ , \tag{2.1}$$

$$\bar{X} = \sum_{i=1}^{N_L} \left| X\left(m + ik\right) - X\left(m + \left(i-1\right)k\right) \right| \ ,$$

where $N_L = \lfloor (N_x - m)/k \rfloor$, $\lfloor \bullet \rfloor$ is the floor operator, k is the interval index number, and $m \in [1, k]$ is the start time index number.

The average of $L_m\left(k\right)$ over m is denoted as $\langle L\left(k\right)\rangle$ and defines the curve length for time interval k. By varying k over $[1, k_{max}]$ and plotting $\langle L\left(k\right)\rangle$ versus k on a log-log scale, the data ideally forms a straight line, with a proper selection of k_{max}. The fractal dimension d is defined as the negative of the line slope, which can be calculated using a least squares method. Furthermore, k_{max} is empirically chosen. If it is too large, the data plotted on the log-log scale will not be linear. If it is too small, there will not be enough data points for an accurate linear fit. For this work, a value of $k_{max} = 10$ is chosen for all fractal calculations.

Using the fractal dimension vector $\mathbf{d}$ formed across all data windows, BSCD is applied to determine the *a-posteriori* probability that a given fractal dimension $d_m \in \mathbf{d}$ represents the data change point corresponding to the burst start. The *a-posteriori* Probability Distribution Function (PDF) for m given $\mathbf{d}$ is [41]

$$p\left(\{m\}\left|\mathbf{d}, I\right.\right) \propto \left[\sqrt{m\left(N_F - m\right)} \times \bar{d}^{\,\left(\frac{N_F-2}{2}\right)}\right]^{-1} , \tag{2.2}$$

$$\bar{d} = \sum_{i=1}^{N_F} d(i)^2 - \frac{1}{m}\left[\sum_{i=1}^{m} d(i)\right]^2 - \frac{1}{N_F - m}\left[\sum_{i=m+1}^{N_F} d(i)\right]^2 ,$$

where N_F is the length of $\mathbf{d}$, $\lfloor \bullet \rfloor$ is the floor operator, I denotes prior information, and m is the potential change point being evaluated. The value of m corresponding to $max[p\left(\{m\}\left|\mathbf{d}, I\right.\right)]$ establishes the burst start sample number. Representative responses for Fractal-BSCD processing are shown in Figure 2.1 where the circled region highlights the burst start location at $t = 0$. As illustrated in the bottom *a-posteriori* PDF response there is a distinct peak that corresponds to the burst start time.

Work in [23, 68] shows that abrupt, non-gradual feature changes are important for the Fractal-BSCD process to work effectively. Signals having more gradual ramp-like versus impulse-like responses are problematic and require alternate methods of detection. Similar BSCD-based methods have been considered to address the increased challenge, e.g., Bayesian Ramp Change Detection [63, 67]. However, as detailed in the next section there are alternatives to BSCD-based methods that have proven effective as well.

2.1.2 Traditional Variance Trajectory.

The Traditional Variance Trajectory (Traditional VT) alternative to burst detection exploits instantaneous signal characteristics to discriminate between signal and channel background responses. While the Traditional VT process can be applied to any arbitrary sequence of data, it has

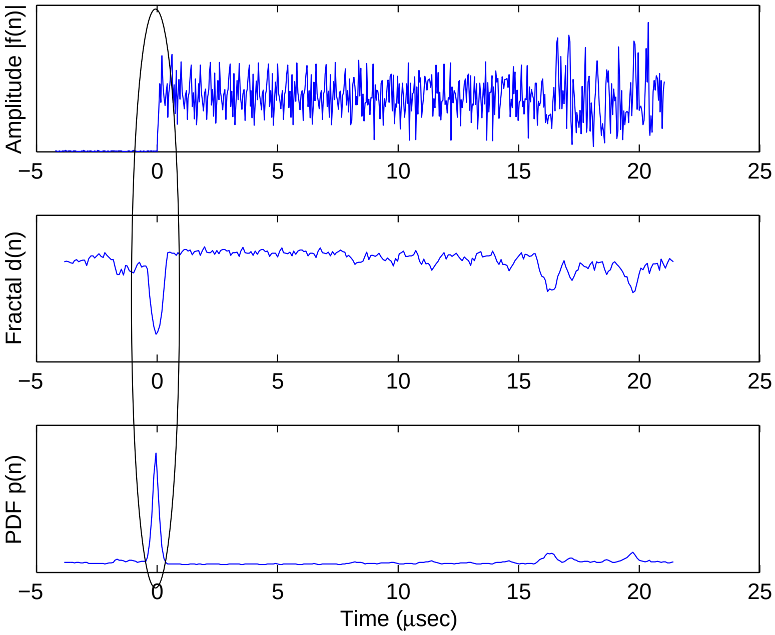


Figure 2.1: Representative responses for Fractal-BSCD processing: (Top) Instantaneous Amplitude, (Middle) Fractal d, and (Bottom) *A-Posteriori* PDF. The circled region highlights the burst start location at $t = 0$.

previously been used for burst detection with both instantaneous phase [23] and instantaneous amplitude characteristics [55]. Given an arbitrary input sequence, the Traditional VT process consists of 1) dividing the input sequence into sequential subsequences, or windows of data, which may or may not overlap, 2) calculating the variance over each window of data, and 3) forming the "trajectory" sequence as the difference between consecutive window variances. Given arbitrary sequence $\{x(k)\}$, $k = 1, 2, ..., N_x$, the *variance trajectory* of $\{x(k)\}$ is denoted as the sequence $\{VT_x(i)\}$ where the i^{th} element is given by [55]

$$VT_x(i) = |W_x(i) - W_x(i+1)| \ , \tag{2.3}$$

$$i = 1, 2, ..., L_w - 1 \ ,$$

$$W_x(m) = \frac{1}{N_w} \sum_{k=1+(m-1)N_s}^{1+(m-1)N_s+N_w} \left[x(k) - \mu_w\right]^2 \ , \tag{2.4}$$

$$m = 1, 2, ..., L_w \ ,$$

where N_w is the window extent, and N_s is the number of samples the window advances between sequential calculations. The μ_w factor in (2.4) is the sample mean of $\{x_w(k)\}$ which is the subsequence of consecutive elements from $\{x(k)\}$ contained in window w.

Figure 2.2 shows representative responses for Traditional VT processing where the top plot is the magnitude response of $\{x(k)\}$ and the other two plots are the corresponding responses for Traditional VT at $SNR = 40$ dB and $SNR = 0$ dB. The circled region highlights the burst start location at $t = 0$. As seen in the $SNR =$ 40 dB response, there is a distinct peak corresponding to the burst start time near $t = 0$. The sensitivity of Traditional VT processing to SNR variation is evident in the $SNR = 0$ dB response where the peak response near the burst start time is virtually indistinguishable from earlier ($t < 0$) peaks. As used here and in other previous work with instantaneous signal characteristics, the degradation of Traditional VT performance at lower SNRs directly impacts burst detection and location error and subsequent classification performance.

2.2 RF Fingerprint Classification

There has been considerable work in previous years involving the exploitation of RF signal characteristics to classify signals and identify the devices producing them [23,54,55,64,66,68]. Collectively, these works embody the field of RF Fingerprint Classification which fundamentally requires two processes, including: 1) fingerprint generation and 2) fingerprint classification. Fingerprint generation requires the selection and extraction of features that enable signal/device discrimination. Desirable properties of the selected feature set include: 1) reduced dimensionality to minimize

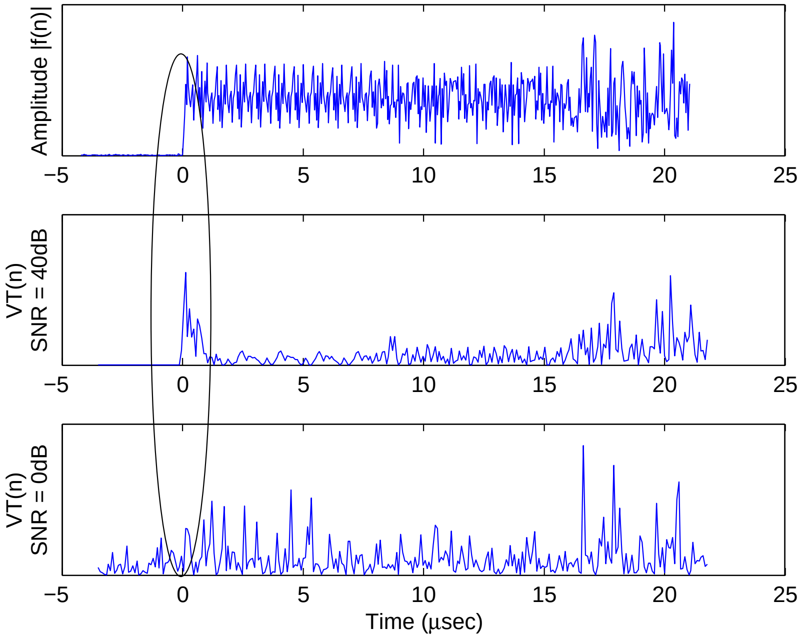


Figure 2.2: Representative responses for VT processing: (Top) Instantaneous Amplitude, (Middle) $SNR = 40$ dB, and (Bottom) $SNR = 0$ dB. The circled region highlights the burst start location at $t = 0$.

processing and storage requirements, 2) intra-device repeatability, and 3) inter-device uniqueness. For this work, the classification features are statistics of instantaneous signal characteristics per the details provided in Section 2.2.1 and Section 2.2.2. The resultant RF Statistical Fingerprints are then used for signal/device classification per the details provided in Section 2.2.3.

2.2.1 Instantaneous Signal Characteristics. While there are many signal characteristics that could be used for device identification (instantaneous responses, peak responses, average responses, amplitude, phase, frequency, power, etc.), a majority of earlier related works have predominantly focused on instantaneous amplitude and instantaneous phase characteristics [23, 64, 66, 68]. The most recent research has exploited instantaneous frequency characteristics as well [54, 55]. As adopted for consistency with these previous work, the following development of instantaneous signal characteristics is provided for completeness.

Samples of a complex time domain (TD) signal having in-phase and quadrature components of $I_{TD}(n)$ and $Q_{TD}(n)$, respectively, can be expressed as

$$s_{TD}(n) = I_{TD}(n) + jQ_{TD}(n) \ , \tag{2.5}$$

and have corresponding instantaneous amplitude, $a(n)$, instantaneous phase, $\phi(n)$, and instantaneous frequency, $f(n)$, responses are given by

$$a\left(n\right) = \sqrt{I_{TD}^{2}\left(n\right) + Q_{TD}^{2}\left(n\right)} \ , \tag{2.6}$$

$$\phi\left(n\right) = \tan^{-1}\left[\frac{Q_{TD}\left(n\right)}{I_{TD}\left(n\right)}\right] \ , \tag{2.7}$$

$$f(n) = \frac{1}{2\pi}\left[\frac{\phi(n) - \phi(n-1)}{\Delta n}\right] \ . \tag{2.8}$$

In practice, each characteristic response is "centered" (mean removed) to remove collection system biases that may unduly influence subsequent processing. The instantaneous amplitude and frequency responses are simply centered using

$$a_c(n) = a(n) - \mu_a \ , \tag{2.9}$$

$$f_c(n) = f(n) - \mu_f \ , \tag{2.10}$$

where $n = 1, 2, 3, \ldots, N_M$, N_M is the total number of samples in the sampled signal, and μ_a and μ_f are amplitude and frequency means calculated across N_M samples of (2.6) and (2.8), respectively.

The phase centering process is somewhat more involved and includes removal of a linear phase component prior to centering. This component may be due to collection receiver coloration or result from inexact frequency estimation during post-collection down-conversion. Given the phase response in (2.7), the *non-linear phase* response is given by

$$\phi_{nl}(n) = \phi(n) - 2\pi\mu_f(n)\Delta_t \ , \tag{2.11}$$

where μ_f is the frequency mean used in (2.10) and Δ_t is the time sample spacing. As a final step, the mean of ϕ_{nl} is removed to yield the desired *centered non-linear* phase which is given by

$$\phi_{cnl}(n) = \phi_{nl}(n) - \mu_{\phi_{nl}} \ , \tag{2.12}$$

where $\mu_{\phi_{nl}}$ is the mean of $\phi_{nl}(n)$ in (2.11). The centering of signal characteristics in (2.9)–(2.12) is consistent with previous fingerprint classification work that successfully employed similar procedures [54, 55].

2.2.2 Statistical Feature Metrics. Direct use of signal characteristics such as those presented in Section 2.2.1 for classification features can be prohibitive in terms of data storage memory requirements and computational processing time. The computational burden can be eased by reducing the feature dimensionality used for fingerprint classification. This was successfully accomplished in previous work using inherent statistical behavior of the signal characteristics vice the signal characteristics themselves [54, 55]. As adopted from this earlier work, the statistics of interest here included the variance (σ^2), skewness (γ), and kurtosis (κ). For arbitrary sequence $\{x(k)\}$, $k = 1, 2, ..., N_x$, these statistics are defined as [36]:

$$\sigma_x^2 = \frac{1}{N_x}\sum_{k=1}^{N_x}\left[x(k) - \bar{x}\right]^2 \ , \tag{2.13}$$

$$\gamma_x = \frac{\frac{1}{N_x}\sum_{k=1}^{N_x}\left[x(k)-\bar{x}\right]^3}{\left\{\frac{1}{N_x}\sum_{k=1}^{N_x}\left[x(k)-\bar{x}\right]^2\right\}^{3/2}} \ , \tag{2.14}$$

$$\kappa_x = \frac{\frac{1}{N_x}\sum_{k=1}^{N_x}\left[x(k)-\bar{x}\right]^4}{\left\{\frac{1}{N_x}\sum_{k=1}^{N_x}\left[x(k)-\bar{x}\right]^2\right\}^{2}} \ , \tag{2.15}$$

where $\bar{x}$ is the sample mean of $\{x(k)\}$. The final RF statistical fingerprints are formed by calculating these statistics for the appropriate centered instantaneous signal characteristic(s) in Section 2.2.1, i.e., setting $\{x(k)\}$ equal to $\{a_c(n)\}$ with elements from (2.9), setting $\{x(k)\}$ equal to $\{f_c(n)\}$ with elements from (2.10), and/or setting $\{x(k)\}$ equal to $\{\phi_{cnl}(n)\}$ with elements from (2.12).

2.2.3 MDA/ML Classification. While many different techniques have been researched and are available for classification, they all employ two fundamental processes: training and classification. That is, they *train* the classifier using a subset of the input data and then *classify* using the remaining data. For the most part, these techniques are oblivious to what the input data actually represents and their performance is predominantly driven by the statistical behavior of the data. With regard to RF fingerprint classification, there has been little novelty in developing specialized classification techniques and most researchers have opted for well-established techniques. The predominant techniques of choice have been based on neural networks [45,51,52,57–59,62,63,65], with some limited additional work based on Kalman filtering and/or a Hotelling statistic [22, 28].

Multiple Discriminant Analysis/Maximum Likelihood (MDA/ML) classification has emerged as a viable alternative and successfully used for RF fingerprint classification [54]. Multiple Discriminant Analysis (MDA) is an extension of Fisher's Linear Discriminant (FLD) process for more than two classes [16]. For a 3-class problem, the

MDA process projects higher-dimensional data onto a 2-dimensional "Fisher plane" that maximizes inter-class distances while simultaneously minimizing intra-class distances. In principle, this method cannot improve classification potential. However, it provides good class separation and visualization of data having input dimensionality greater than three. Using this lower-dimensional data, decision boundaries calculated from ML distributions are determined assuming normally distributed input data, equal costs and uniform prior probabilities. In general, to discriminate c classes using d-dimensional input data, the input vector $\mathbf{x}$ is linearly projected onto a $(d-1)$-dimensional space using

$$\mathbf{y} = \mathbf{W}^T\mathbf{x} \ , \tag{2.16}$$

where $\mathbf{y}$ is the vector of projected values and $\mathbf{W}$ is a $d \times (c-1)$ projection matrix. Classification is performed using unknown data and the trained 2-dimensional decision boundaries calculated from ML distributions. The process classifies each "unknown" input data set by projecting it onto the trained Fisher plane according to (2.16). Projected points falling within the correct region are correctly classified while those falling outside the correct region are misclassified. The percentage of correct classification is determined based on the total number of unknown trials. A more complete description of the MDA/ML process is provided in [10].

2.3 Dual-Tree ℂomplex Wavelet Transform

Device classification can be performed using a Discrete Wavelet Transform (DWT), with one popular method using a subset of the largest DWT coefficient magnitudes as the classification features [44]. As mentioned in Section 1.1.2.4, one distinct disadvantage of DWT-based approaches is that the DWT is not shift invariant. As with signal denoising, this presents a problem for RF fingerprinting applications given that robust classification performance relies on the fingerprint features being unique, repeatable and stable. These properties cannot be assured if the underlying features (DWT coefficients) vary dramatically throughout the processing interval of interest. For example, variation in burst detection and start location error generally translates

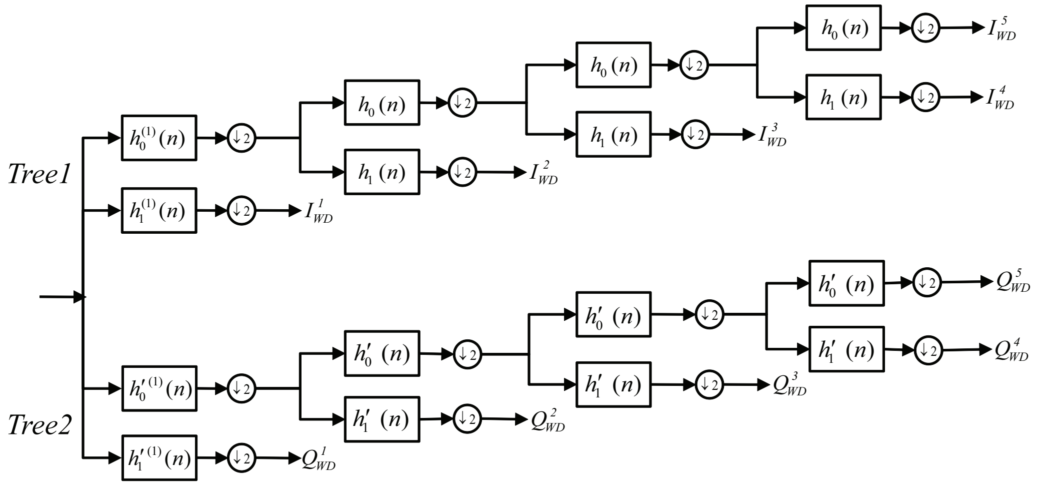


Figure 2.3: Four Stage Dual-Tree ℂomplex Wavelet Transform (DT-ℂWT) [2].

to greater variation in fingerprint features. To address the lack of shift invariance in DWT processing, a Dual-Tree ℂomplex Wavelet Transform (DT-ℂWT) is considered.

The DT-ℂWT is a DWT extension that is "nearly shift-invariant," i.e., the DT-ℂWT coefficients are independent of time domain shift and more strongly dependent on interscale and intrascale neighborhoods [50]. This shift invariance has been previously exploited to improve classification performance for hyperspectral images [38]. Furthermore, the DT-ℂWT magnitude response exhibits reduced ringing that is generally induced by high-frequency noise and sharp discontinuities [50].

The DT-ℂWT is commonly implemented using two real-valued filter banks. These are denoted as *Tree1* and *Tree2* in Figure 2.3 which shows one common architecture for DT-ℂWT implementation [2]. The scaling and wavelet functions for *Tree1* are symmetric (even functions) while *Tree2* has scaling and wavelet functions that are anti-symmetric (odd functions). The wavelet and scaling functions, $\psi(t)$ and $\phi(t)$ respectively, for the *Tree1* filter bank are given by [2, 50],

$$\psi(t) = \sqrt{2}\sum_{n} h_1(n)\phi(2t-n) \ , \tag{2.17}$$

$$\phi(t) = \sqrt{2} \sum_n h_0(n)\phi(2t-n) \ , \tag{2.18}$$

where the filter coefficients $h_1(n)$ and $h_0(n)$ are implemented directly as the Analysis Filters (AF) given in [49] (see Section A.5). Ideally, the corresponding functions for the *Tree2* filter bank are the Hilbert transforms of (2.17) and (2.18), expressed as

$$\psi^{'}(t) = \sqrt{2} \sum_n h_1^{'}(n)\phi'(2t-n) \ , \tag{2.19}$$

$$\phi^{'}(t) = \sqrt{2} \sum_n h_0^{'}(n)\phi'(2t-n) \ , \tag{2.20}$$

where the filter coefficients $h_1^{'}(n)$ and $h_0^{'}(n)$ are implemented directly as the Analysis Filters (AF) given in [49] (see Section A.5).

As shown in Figure 2.3, the first stage filters for both *Tree1* and *Tree2* have different coefficients when compared to the later stage filters and are denoted as $h_1^{(1)}(n)$, $h_0^{(1)}(n)$, $h_1^{'(1)}(n)$, and $h_0^{'(1)}(n)$, respectively. The first stage filter coefficients are implemented directly as the First Analysis Filters (FAF) given in [49] (see Section A.5).

For real-valued input signals, the *Tree1* and *Tree2* filter banks yield real-valued wavelet domain (WD) coefficients representing real (I_{WD}^l) and imaginary (Q_{WD}^l) components of complex coefficients [50]. These components can be functionally combined in a form similar to (2.5) and expressed as

$$s_{WD}^l(n) = I_{WD}^l(n) + jQ_{WD}^l(n) \ . \tag{2.21}$$

Using $s_{WD}^l(n)$ elements from (2.21), the sequence $\{s_{WD}(n)\}$ of all elements can be interpreted as what may be called a "complex sampled WD signal." Given the similar structure of this WD signal and the TD signal in (2.5), WD fingerprint classification can be performed using the process in Section 2.2. In this case, the WD signal in (2.21) can be used in (2.6)–(2.12) to generate WD signal characteristics and statistics calculated per (2.13)–(2.15) to form statistical WD fingerprints.

2.4 Denoising

Wavelet transforms, and in particular the DWT, have been used to denoise signals by exploiting differences in the distribution of signal and embedded noise contributions in the wavelet domain [4, 7, 8, 14, 15, 17–19, 44, 61]. In the case of an AWGN channel, the noise contribution remains Gaussian in the wavelet domain and thus uniformly distributed with respect to scale [43]. However, burst signal contributions are non-uniformly distributed in the wavelet domain and significant signal content is generally manifested in large wavelet coefficient magnitudes. Thus, one common approach for denoising using wavelets involves 1) transforming the time domain signal into the wavelet domain, 2) thresholding the wavelet coefficient magnitudes, 3) zeroing-out all coefficients with magnitudes less than the threshold and retaining the others, and 4) inverse transforming the thresholded coefficient set to yield the denoised time domain signal [4,7,8,14,15,17–19,44,61]. The effectiveness of this approach is based on selecting a threshold value that 1) retains coefficients containing a majority of desired signal contributions while 2) zeroing-out coefficients that are dominated by noise contributions. Due to the compaction property of the wavelet transform, there are relatively few large magnitude coefficients. Thus, a majority of the coefficients can be zeroed-out which minimizes the remaining noise contribution in the denoised response.

Summary

This chapter presented the relevant technical background information on Burst Detection and Location, RF Fingerprinting, DT-ℂWT, and Denoising. The information here supports subsequent material presented in the document.

III. Methodology

This chapter provides the overall demonstration process used for generating results and conducting analysis. A detailed description is included for both the "Signal Collection" (Section 3.2) hardware and "Post-Collection Processing" (Section 3.3) software processes. The primary hardware used for signal collection was AFIT's RF Signal Intercept and Collection System (RFSICS) with subsequent data processing accomplished exclusively in a MATLAB® environment. Denoising using the DT-ℂWT is described in Section 3.4. Threshold determination for the various processes is described in Section 3.5.

3.1 Overall Demonstration Process

Figure 3.1 shows the overall demonstration process that was used for generating all results presented in Chapter 4. The dashed boundaries denote the processes that are primarily conducted in hardware and software. The "Signal Collection" hardware process consisted of placing communication devices the RFSICS in a given electromagnetic environment and making signal collections. The collected signal data (a series of complex valued samples) is passed along for subsequent Post-Collection Processing which was accomplished exclusively in a MATLAB® environment. The implementation and functionality of various processes in Figure 3.1 is discussed in the following sections.

3.2 Signal Collection Process

Classification performance was demonstrated for two cases, including: 1) *Intra-manufacturer* where all devices are from a given manufacturer and have different serial numbers, and 2) *Inter-manufacturer* where at least one of the devices is from a different manufacturer. A summary of manufacturers, device serial numbers and signals considered is provided in Table 3.1. Consistent with the overall research objective, the table shows that results were not generated for all combinations of manufactur-

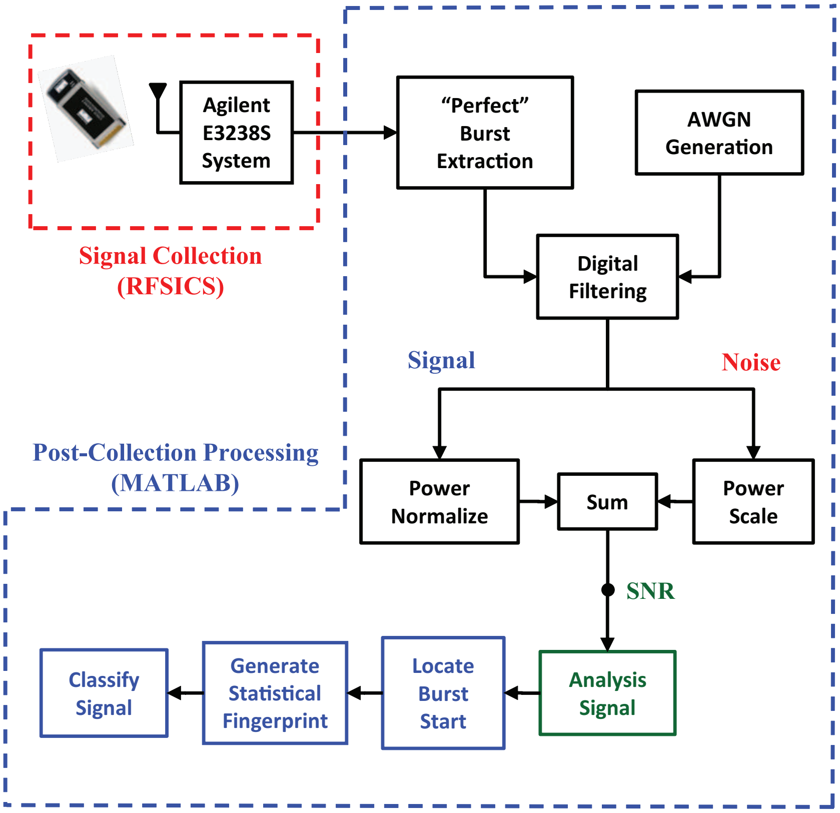


Figure 3.1: Overall demonstration process for signal collection, analysis signal generation, burst detection and start location, fingerprint extraction, and classification.

ers, devices and signals. Rather, selected combinations were used to generate results to sufficiently support final research conclusions–it is believed that results from an exhaustive analysis would not fundamentally change these conclusions. From an operational perspective, the potential number of combinations that may be of interest to the broader technical community is nearly limitless and based on tens of manufacturers, tens of device types per manufacturer, and hundreds of serial numbers per device type. Considering various combinations of alternatives remains an area of interest for future research and is subject to technical community interest.

For all results presented, the signals were collected with both the device under test and the RFSICS in an anechoic chamber. Basic functionality of the RFSICS is provided by Agilent's E3238S system [1]. This includes an RF front-end collection range of 20.0 MHz to 6.0 GHz from which a band of interest is selected using a

Table 3.1: Device manufacturers, serial numbers, and signal types (802.11A and 802.11G) used for generating Chapter 4 results.

Manu	Serial Number / Signal Type			
Cisco	N4U9 / A&G	N4UD / A&G	N4UW / A&G	N4PX / A&G
Linksys	0306 / A&G	0307 / A		
Netgear	0273 / A	0217 / A		
Dell	BTA4 / A			
Airmag	2C01 / G			

tunable RF filter with fixed bandwidth of 36.0 MHz. The selected RF band is down-converted to an Intermediate Frequency (IF) of 70.0 MHz and passed to a digitizer. The digitizing process consists of down-conversion (near baseband), 12-bit analog-to-digital conversion at 95 M samples-per-second (sps), digital filtering (user defined bandwidth), Nyquist compliant sub-sampling, and data storage as complex In-phase (I) and Quadrature (Q) components. A digital filter bandwidth of 18.56 MHz was selected for all 802.11A signals collected for this work. This resulted in the RFSICS automatically applying a sub-sampling factor of four, for a final sample rate of $f_s =$ 23.75 Msps and corresponding sample interval of $T_s = 1/f_s \approx 42.1$ $n$sec per sample. The typical *collected* SNR for the chamber collected signals is on the order of $SNR =$ 40 dB.

3.3 Post-Collection Processing

Post-Collection Processing in Figure 3.1 is accomplished exclusively in a MATLAB® environment using the near-baseband, complex I-Q data from RFSIC collections. Post-collection processing includes analysis signal generation, burst detection and start location, statistical fingerprint generation and signal classification. The functionality and implementation of each of these processes is discussed in the following subsections.

3.3.1 Analysis Signal Generation. The first post-collection process of "Perfect" Burst Extraction uses the near-baseband, complex I-Q data from the RFSIC collections. Extraction is accomplished through a combination of automated amplitude threshold detection followed by visual analysis and manual alignment to accurately identify the sample number corresponding to the burst start. The extracted burst responses are digitally filtered using a baseband filter and power-normalized. A 6th-order Chebyshev digital filter was implemented having a –3 dB bandwidth of 7.7 MHz. At this point, the sample frequency of the filtered signal is $f_s = 23.75$ Msps which effectively represent oversampling by a factor of approximately 1.5 times Nyquist. Provided that the RFSICS collection and subsequent post-processing is identical for all signals, it is reasonable to assume that "recording coloration" (variation in amplitude, phase and/or frequency characteristics) induced by the RFSICS and post-processing prior to burst start location, statistical fingerprint generation and signal classification is approximately identical. This is important in the overall process and ensures that final results are based on as received signal characteristics and features versus being unduly influenced by signal-dependent collection and post-processing coloration.

The desired "Analysis Signal" is intended to simulate varying SNR conditions that typically exist in an operational environments. This signal is generated by adding like-filtered, power-scaled noise to the digitally filtered, power-normalized signal. This is done by generating random complex AWGN that is filtered using the same digital filter as used for the signal. The filtered noise signal is then power-scaled to achieve the desired analysis SNR when added to the filtered signal. A representative instantaneous amplitude response from a collected 802.11A RF burst is shown in Figure 3.2 for analysis SNRs of $SNR = 10$ dB and $SNR = 0$ dB.

3.3.2 Burst Detection and Start Location. The sequential burst detection and burst start location process is implemented relative to what may occur in an operational collection system, i.e., a real-time system samples the environment, detects the "presence" of bursts and locates the burst start point (sample number) within the turn-on region. This process was functionally implemented in the Locate Burst Start block in Figure 3.1. The specific burst detection and location techniques

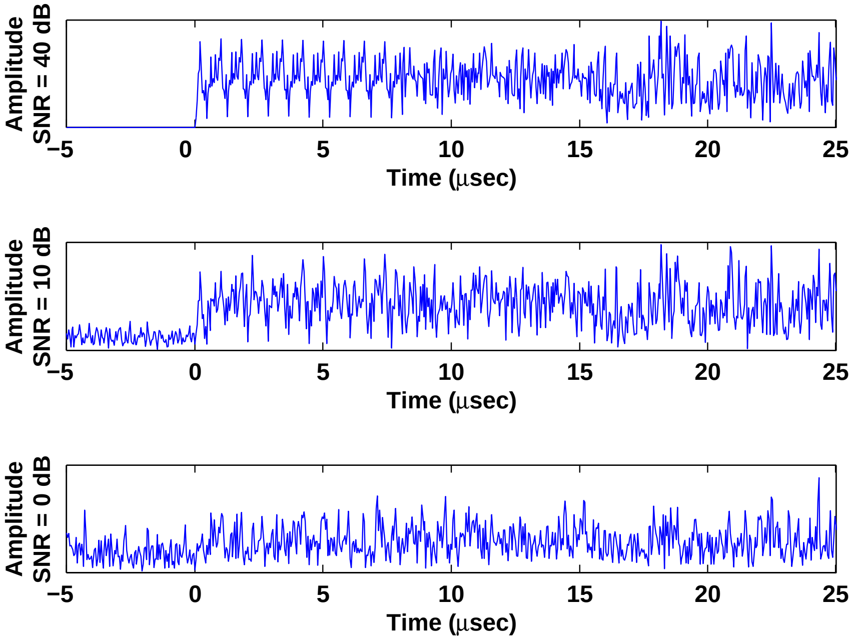


Figure 3.2: Instantaneous amplitude responses for collected 802.11A signal: (Top) *Collected* Signal, (Middle) *Filtered* Signal-plus-AWGN at $SNR = 10$ dB and (Bottom) *Filtered* Signal-plus-AWGN at $SNR = 0$ dB.

that were implemented in this block include Fractal-Bayesian Step Change Detector (Fractal-BSCD) (Section 2.1.1), Traditional Variance Trajectory (VT) (Section 2.1.2) and Denoised VT (Section 3.4). As presented in Chapter 4, results were generated using each of these techniques to characterize 1) their burst detection and location error performance, 2) their performance relative to each other, and 3) their corresponding error impact on subsequent fingerprint classification performance. It became evident throughout the research that reliable comparison of error impact on classification performance could only be accomplished if all the same bursts were used for classification following detection and location. To ensure a fair comparison, the concept of "dual-convergent" bursts was developed as explained next.

Undetected bursts are those which are actually received yet their presence is not declared. Detected bursts are those which are received and their presence is declared. The focus of this work is on detected bursts with subsequent algorithmic processing used to determine burst start location. For those cases where the burst start location algorithm does not converge in accordance with prescribed criteria (number of iterations, parametric tolerance, etc.), the detected bursts are designated as "non-convergent" and a default burst start location value assigned. When algorithm convergence occurs, the bursts are designated as "convergent" and the estimated location assigned. When algorithm convergence occurs for identical bursts with two different burst location techniques, the bursts are designated as "dual convergent."

3.3.2.1 Burst Detection. This process is similar to coarse burst detection that is accomplished in an the RF environment to detect the *presence* of RF bursts. The input analysis signal is first segmented into contiguous, non-overlapping sub-sections or windows such that $N_s = N_w$ in (2.4). While not a requirement, non-overlapping windows are used to minimize processing time. This has the disadvantage of producing coarser estimates of where the actual burst response starts, while at the same time capturing more signal power within each window and improving detectability. For all results presented in Chapter 4, a window size of $N_w = 512$ signal samples (21.6 μsec) is used.

Two burst detection methods, Traditional VT and Denoised VT as described in Section 2.1.2 and Section 3.4 respectively, are applied to the windowed signal data and an *a-priori* coarse detection threshold t_{Det} used to declare detection. Once a coarse detection occurs, the corresponding segment of windowed signal data is passed on for start location determination where it is assumed that an actual burst start occurs within the window. However, as with all coarse signal detection approaches, false alarms can occur with bursts falsely declared present. Coarse detection performance results are provided in Section 4.2.1.

3.3.2.2 Burst Start Location. This process is similar to coarse burst detection in that the Traditional VT and Denoised VT techniques are reapplied to determine the final start location. In addition, the Fractal-BSCD technique in Section 2.1.1 is considered as well. For the Traditional VT and Denoised VT techniques, the precise start location is indicated by the *time* (sample number) at which an abrupt change occurs in the VT_a response of (2.3). For the Fractal-BSCD technique the precise start time location corresponds to the *time* (sample number) at which a maximum occurs in the *a-posteriori* PDF of (2.2). The effectiveness of these approaches is based on an implicit assumption that bursts of OFDM-based signals can be modeled as having a step change response in/near the turn-on transient region. This assumed response is consistent with 802.11A specifications [29] and has been successfully exploited in related research [42, 63, 67].

For all three techniques, the segment of windowed data that is passed from the coarse detection process is further sub-segmented using much narrower and highly overlapped windows. The overlapping windows allow for better location accuracy at the expense of increased processing time. For this work, a window size of $N_w = 20$ samples (0.84 μsec) is used with a shift of $N_s = 2$ samples (84.2 $nsec$) between consecutive windows.

For demonstrating performance of the Traditional VT and Denoised VT techniques, an *a-priori* location threshold t_{Loc} ($t_{Loc} \neq t_{Det}$) is used to automatically estimate the burst start location based on a significant peak response occurring in VT_a of (2.3). When a significant peak is located the signal is passed on for subsequent fingerprint generation. In some cases no significant peak is found and the algorithm does not converge to a solution. This non-convergent condition can occur if there is no burst present (coarse burst detection false alarm) or if the threshold is set too high for the burst under evaluation. There are two options for dealing with non-convergent bursts, including: 1) the burst can be discarded without subsequent processing, or 2) a default start location value can be assigned and subsequent processing performed. In an operational environment where the system has access to a large number of bursts, discarding burst may be a reasonable choice and have minimal impact on final system performance. For this work the probability of coarse detection is effectively 100% given that collected signals are first passed through "Perfect" Burst Extraction (via a visual and manual inspection of each burst) according to Figure 3.1. Given this and data collection limitations, a default location is assigned to non-convergent bursts that produce no significant peak in the $VT_a(i)$ response. The default location time (sample number) is chosen to correspond with the last sample in the window of data passed by the coarse detection process. In presenting results in Chapter 4, non-convergent pulses are only included when characterizing detection and start location error performance of the three techniques considered. As explained earlier, they are not included when assessing the impact of this error on end-to-end signal classification performance. A performance comparison of Traditional VT and Denoised VT burst start location performance is presented in Section 4.2.2.4.

In assessing performance of the Fractal-BSCD technique in Section 2.1.1 it was found that there were no non-convergent bursts. The 100% convergence of Fractal-BSCD processing is ensured given the maximum operator in (2.2). Relative to the Traditional VT and Denoised VT techniques, this could be an operational disadvantage as there is no inherent back-up capability for detecting bad pulses (false alarms). A performance comparison of Fractal-BSCD and Traditional VT burst start location performance is presented in Section 4.2.2.

3.3.3 Statistical Fingerprint Generation. Following burst detection and start location, the RF statistical fingerprints are generated using the process shown in Figure 3.3. As indicated within the dashed lines, the Characteristics and Statistics generating functions are identical for both the time domain (TD) and wavelet domain (WD) techniques. A signal region of interest is selected from the input analysis signal and parsed into a predefined number of sub-regions for fingerprint generation. For the 802.11A/G signals considered here, the burst preamble is the region of interest. This choice was based on 1) previous works which successfully exploited the preamble [20,54,55], and 2) the preamble sequences being identical for all bursts per the 802.11 standard [29]. Figure 3.4 shows the modulated signal response for the standard preamble comprised of 10 short followed by 2 long symbols. For all results presented in Chapter 4, a total of $N_r = 3$ fingerprint regions were used as highlighted in Figure 3.4. The three different fingerprint regions include 1) the first 8.0 μsec which corresponds to ten short OFDM symbols, 2) the last 8.0 μsec which corresponds to two long OFDM symbols, and 3) the entire 16.0 μsec preamble (both short and long symbols).

For TD feature classification, the centered subregion characteristics are calculated using (2.6)–(2.12) and statistical classification features calculated using (2.13), (2.14), and (2.15) for each resultant characteristic response. The resultant TD RF fingerprint (feature vector) consists of 27 total features per collected burst (3 subregions × 3 signal characteristics × 3 statistics). The TD fingerprint for burst b, from device (class) c, in subregion r is given by

$$\begin{aligned}\mathbf{F}_r^{b,c} = [\ &\sigma_r^2(a),\ \sigma_r^2(\phi),\ \sigma_r^2(f),\\ &\gamma_r(a),\ \gamma_r(\phi),\ \gamma_r(f),\\ &\kappa_r(a),\ \kappa_r(\phi),\ \kappa_r(f)\ ]\ ,\end{aligned} \tag{3.1}$$

where $b = 1, 2, 3, \ldots, N_b$ with N_b being the total number bursts, $r = 1, 2, 3, \ldots, N_r$ with N_r being the total number of subregions, and $c = 1, 2, 3$ is the class index. Considering the $N_r = 3$ subregions as used here, the composite TD classification feature vector (1×27) is formed using (3.1) and is given by

$$\mathbf{F}_{TD}^{b,c} = \left[\mathbf{F}_1^{b,c}\ \mathbf{F}_2^{b,c}\ \mathbf{F}_3^{b,c}\right]\ . \tag{3.2}$$

For WD feature classification, the processing is identical to TD processing except that a Dual-Tree ℂomplex Wavelet Transform (DT-ℂWT) decomposition is performed in each subregion. As depicted in Figure 2.3, the DT-ℂWT decomposes each subregion into five levels associated with different wavelet scales. The "complex WD signal" samples are calculated using (2.21), followed by characteristic generation and centering using (2.6)–(2.12). The statistical classification features are calculated using (2.13), (2.14), and (2.15). The resultant WD RF fingerprint (feature vector) consists of 135 total features per collected burst (3 subregions × 5 DT-ℂWT decomposition

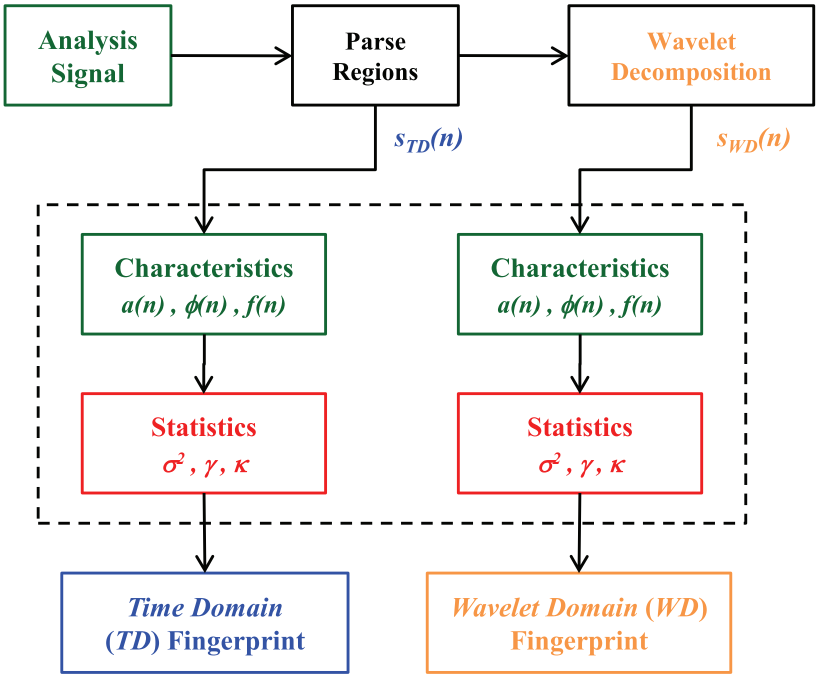

Analysis Signal
Parse Regions
Wavelet Decomposition
$s_{TD}(n)$
$s_{WD}(n)$
Characteristics
$a(n)$, $\phi(n)$, $f(n)$
Characteristics
$a(n)$, $\phi(n)$, $f(n)$
Statistics
σ^2 , γ, κ
Statistics
σ^2 , γ, κ
Time Domain (*TD*) Fingerprint
Wavelet Domain (*WD*) Fingerprint

Figure 3.3: Generation process for statistical RF fingerprints. The Characteristics and Statistics generating functions are identical for both the TD and WD techniques and implemented using (2.6)–(2.12) and (2.13)–(2.15), respectively [32].

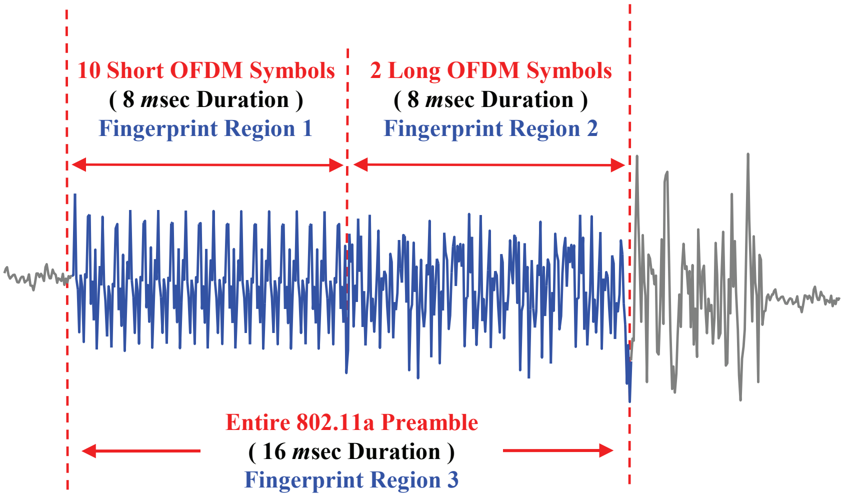


Figure 3.4: 802.11A preamble structure showing OFDM modulated signal response and fingerprint regions.

levels per subregion × 3 signal characteristics × 3 statistics). Paralleling the TD development, the WD fingerprint for burst b, from device c, in subregion r which has been decomposed into l DT-$\mathbb{C}$WT levels is given by

$$\begin{aligned}\mathbf{F}_{r,l}^{b,c} = [\ & \sigma_{r,l}^2(a),\ \sigma_{r,l}^2(\phi),\ \sigma_{r,l}^2(f),\\ & \gamma_{r,l}(a),\ \gamma_{r,l}(\phi),\ \gamma_{r,l}(f),\\ & \kappa_{r,l}(a),\ \kappa_{r,l}(\phi),\ \kappa_{r,l}(f)\ ]\ ,\end{aligned} \tag{3.3}$$

where $l = 1, 2, 3, \ldots, N_l$ with N_l being the total number of DT-$\mathbb{C}$WT decomposition levels per subregion. Considering $N_r = 3$ subregions with $N_l = 5$ levels as used here, the composite WD classification feature vector (1×135) is formed using (3.3) and is given by

$$\begin{aligned}\mathbf{F}_{WD}^{b,c} = \Big[\ & \mathbf{F}_{1,1}^{b,c}\ \mathbf{F}_{1,2}^{b,c}\ \mathbf{F}_{1,3}^{b,c}\ \mathbf{F}_{1,4}^{b,c}\ \mathbf{F}_{1,5}^{b,c}\\ & \mathbf{F}_{2,1}^{b,c}\ \mathbf{F}_{2,2}^{b,c}\ \mathbf{F}_{2,3}^{b,c}\ \mathbf{F}_{2,4}^{b,c}\ \mathbf{F}_{2,5}^{b,c}\\ & \mathbf{F}_{3,1}^{b,c}\ \mathbf{F}_{3,2}^{b,c}\ \mathbf{F}_{3,3}^{b,c}\ \mathbf{F}_{3,4}^{b,c}\ \mathbf{F}_{3,5}^{b,c}\ \Big]\ .\end{aligned} \tag{3.4}$$

3.3.4 MDA/ML Signal Classification. Signal classification is performed using the Multiple Discriminant Analysis/Maximum Likelihood (MDA/ML) process described in Section 2.2.3. For all MDA/ML classification results presented in Chapter 4, a total of $N_b = 2000$ bursts were used from $N_d = 3$ different 802.11A/G devices, with each device denoted as Class A, Class B, and Class C. Fingerprints from each class (device) were used to form a single composite fingerprint matrix for classification. As indicated in the following expressions, the composite matrix is formed by vertically concatenating the feature vectors for either TD using (3.2) or WD using (3.4). The formation of these matrices can be represented as

$$\mathbf{F}_{TD} = \begin{bmatrix} \left[\mathbf{F}_{TD}^{1,1}\ \mathbf{F}_{TD}^{2,1} \ldots \mathbf{F}_{TD}^{N_b,1}\right]^{T_V} \\ \left[\mathbf{F}_{TD}^{1,2}\ \mathbf{F}_{TD}^{2,2} \ldots \mathbf{F}_{WD}^{N_b,2}\right]^{T_V} \\ \left[\mathbf{F}_{TD}^{1,3}\ \mathbf{F}_{TD}^{2,3} \ldots \mathbf{F}_{TD}^{N_b,3}\right]^{T_V} \end{bmatrix}, \tag{3.5}$$

$$\mathbf{F}_{WD} = \begin{bmatrix} \left[\mathbf{F}_{WD}^{1,1}\ \mathbf{F}_{WD}^{2,1} \ldots \mathbf{F}_{WD}^{N_b,1}\right]^{T_V} \\ \left[\mathbf{F}_{WD}^{1,2}\ \mathbf{F}_{WD}^{2,2} \ldots \mathbf{F}_{WD}^{N_b,2}\right]^{T_V} \\ \left[\mathbf{F}_{WD}^{1,3}\ \mathbf{F}_{WD}^{2,3} \ldots \mathbf{F}_{WD}^{N_b,3}\right]^{T_V} \end{bmatrix}, \tag{3.6}$$

where T_V is used here to denote *vector transposition*, i.e., the vectors are transposed with the order of elements within each vector maintained. For $N_b = 2000$ bursts per class, the resultant composite $\mathbf{F}_{TD}$ matrix has dimension 6000×27 and the resultant composite $\mathbf{F}_{WD}$ matrix has dimension 6000×135. The composite fingerprint matrices in (3.5) and (3.6) are column-wise (i.e. per feature) centered and normalized to unit standard deviation. The centering and normalizing processes only aid in fingerprint visualization and do not impact subsequent MDA/ML classification performance. The impact of feature selection (TD and WD) on signal classification performance is demonstrated using the resultant centered and normalized RF fingerprints input to the MDA/ML process.

Monte Carlo simulation and K-fold cross validation processes are used with MDA/ML signal classification. Monte Carlo simulation is used to ensure statistical significance and K-fold cross validation is used to generalize the prediction error to an independent data set [25]. While the required value of K can vary as a function of data "behavior," values of $K = 5$ and $K = 10$ are common choices for cross validation [25]. Using $K = 5$ with $N_b = 2000$ bursts (fingerprints) per device, the input fingerprints are partitioned into $K = 5$ equal subsets (400 each), with $K-1 = 4$ subsets (1600 fingerprints) used for training and the remaining "held-out" subset (400 fingerprints) used for classification [25].

The overall process for MDA/ML classification with K-fold cross validation is shown in Figure 3.5. Accounting for a total of N_{MC} independent Monte Carlo noise realizations, the process for generating average classification results includes the following steps. Note that the Fold Iteration Accumulator in Figure 3.5 is cleared prior to the start of this process.

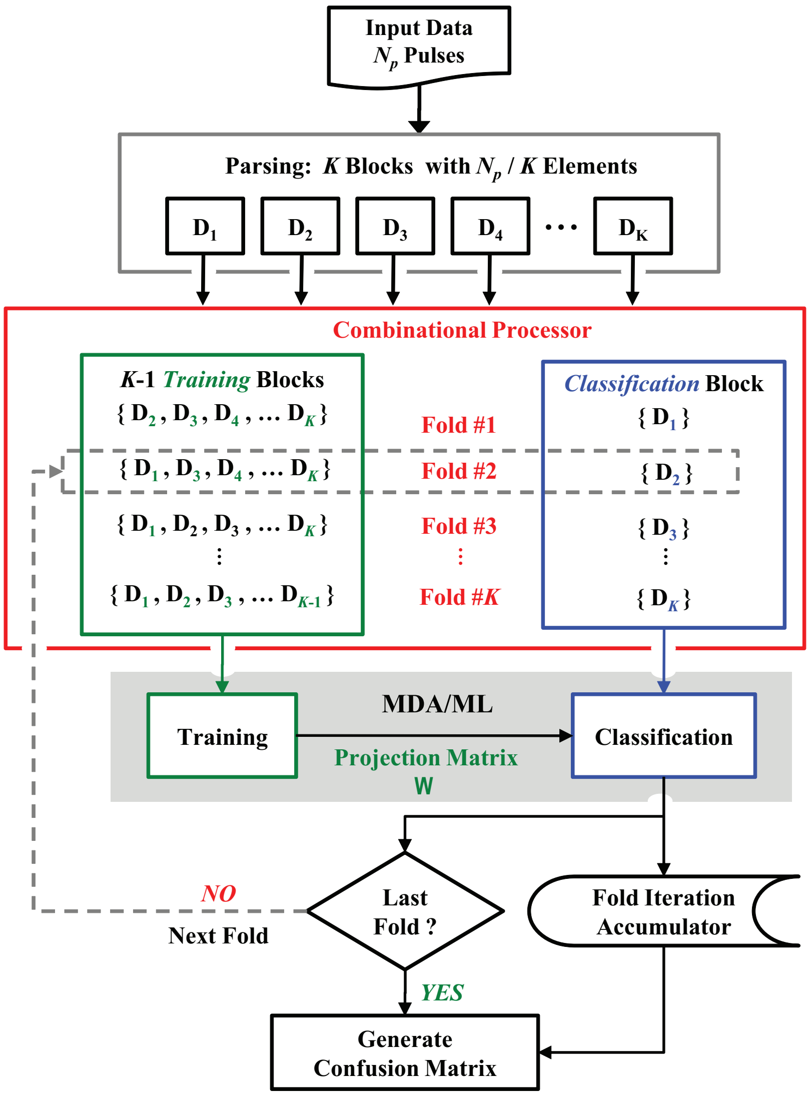


Figure 3.5: MDA/ML classification process with K-fold cross validation [71].

1. Generating the analysis signal for a given SNR per Section 3.3.1

2. Performing burst detection and start location per Section 3.3.2

3. Generating statistical fingerprints per Section 3.3.3 for the technique under evaluation (TD or WD)

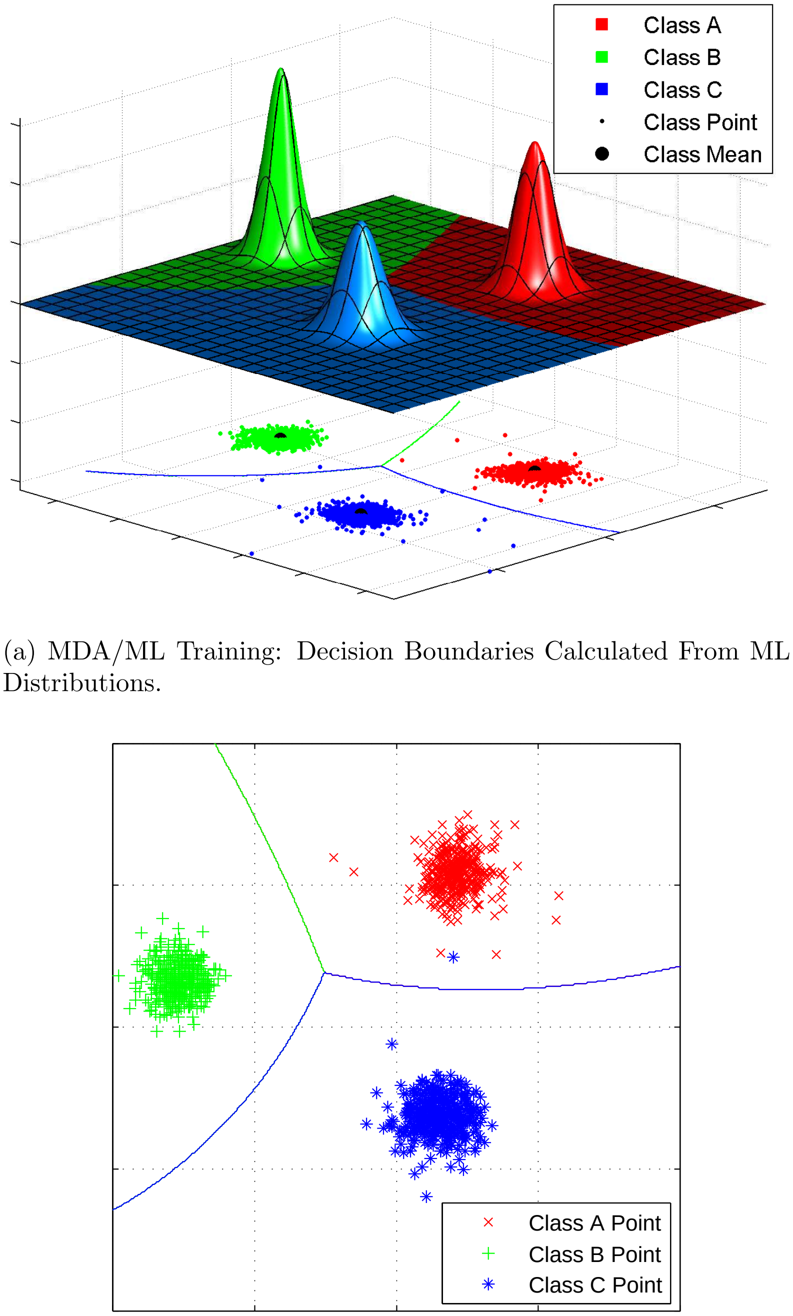
4. Generating projection matrix **W** per (2.16) using $K - 1 = 4$ subsets (80% of the fingerprints) from each device for training and ML classifier parameter calculation

5. Transforming the “held-out” subset (20% of the fingerprints) from each device as “unknown” inputs using $\mathbf{W}$ and classifying each per ML criteria
6. Accumulating the current fold classification results
7. Selecting the next $K-1=4$ blocks for the next fold
8. Repeating Step 4 – Step 7 for $K-1=4$ additional folds
9. Repeating Step 1 – Step 8 a total of N_{MC} times using different independent AWGN realizations for each iteration (Fold Iteration Accumulator not cleared)
10. Averaging Fold Iteration Accumulator results to obtain average classification performance (Accounting for all factors, the final average is based on a total of $N_{MC} \times N_b \times 3$ independent classification decisions.)
11. Repeating the process for each desired analysis SNR

Representative MDA-transformed training fingerprints and trained decision boundaries calculated from ML distributions are shown in Figure 3.6(a) for 802.11A signals at $SNR = 40$ dB. The corresponding projection of “unknown” MDA-transformed fingerprints are shown in Figure 3.6(b) overlayed with trained decision boundaries from Figure 3.6(a). Note that even under these high SNR conditions incorrect classification is possible. For example, one of the Class C ($*$ markers) fingerprints is clearly projected into the Class A ($\times$ markers) ML decision region and would be incorrectly classified.

3.4 DT-ℂWT Denoising Process

Denoising is accomplished using the DT-ℂWT described in Section 2.3 with the process illustrated in Figure 3.7. The complex input signal $f(n)$ is transformed using the DT-ℂWT which outputs complex-valued wavelet coefficients from the *Tree1* and *Tree2* filter banks. These outputs are combined to form real-valued coefficients *d(n)* according to



(a) MDA/ML Training: Decision Boundaries Calculated From ML Distributions.

(b) MDA/ML Classification: Projected Fingerprints.

Figure 3.6: MDA/ML (a) *Training* and (b) *Classification* for 802.11A signals at $SNR = 40$ dB. Lower surface of (a) shows MDA fingerprint projections and trained decision boundaries.

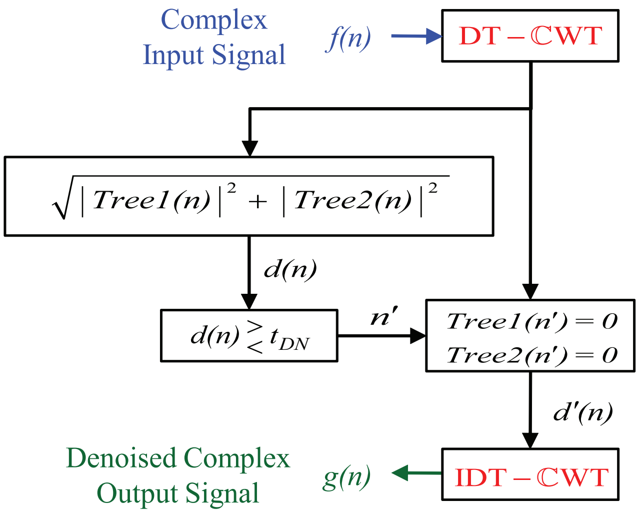


Figure 3.7: Denoising process using the DT-ℂWT in Section 2.3 [31].

$$d(n) = \sqrt{|Tree1(n)|^2 + |Tree2(n)|^2}\ . \tag{3.7}$$

The $d(n)$ coefficients in (3.7) are compared with the denoising threshold t_{DN} and a punctured set of coefficients $d'(n)$ produced by setting all coefficients below t_{DN} to zero and retaining those above t_{DN}, i.e., $\forall n'$ where $d(n') < t_{DN}$, $Tree\mathit{1}(n') = 0$ and $Tree\mathit{2}(n') = 0$. An Inverse DT-ℂWT (IDT-ℂWT) is then applied to $d'(n)$ to produce the denoised complex output signal $g(n)$. The denoised coefficients are subsequently processed using the Traditional VT technique in Section 2.1.2 to generate Denoised VT results.

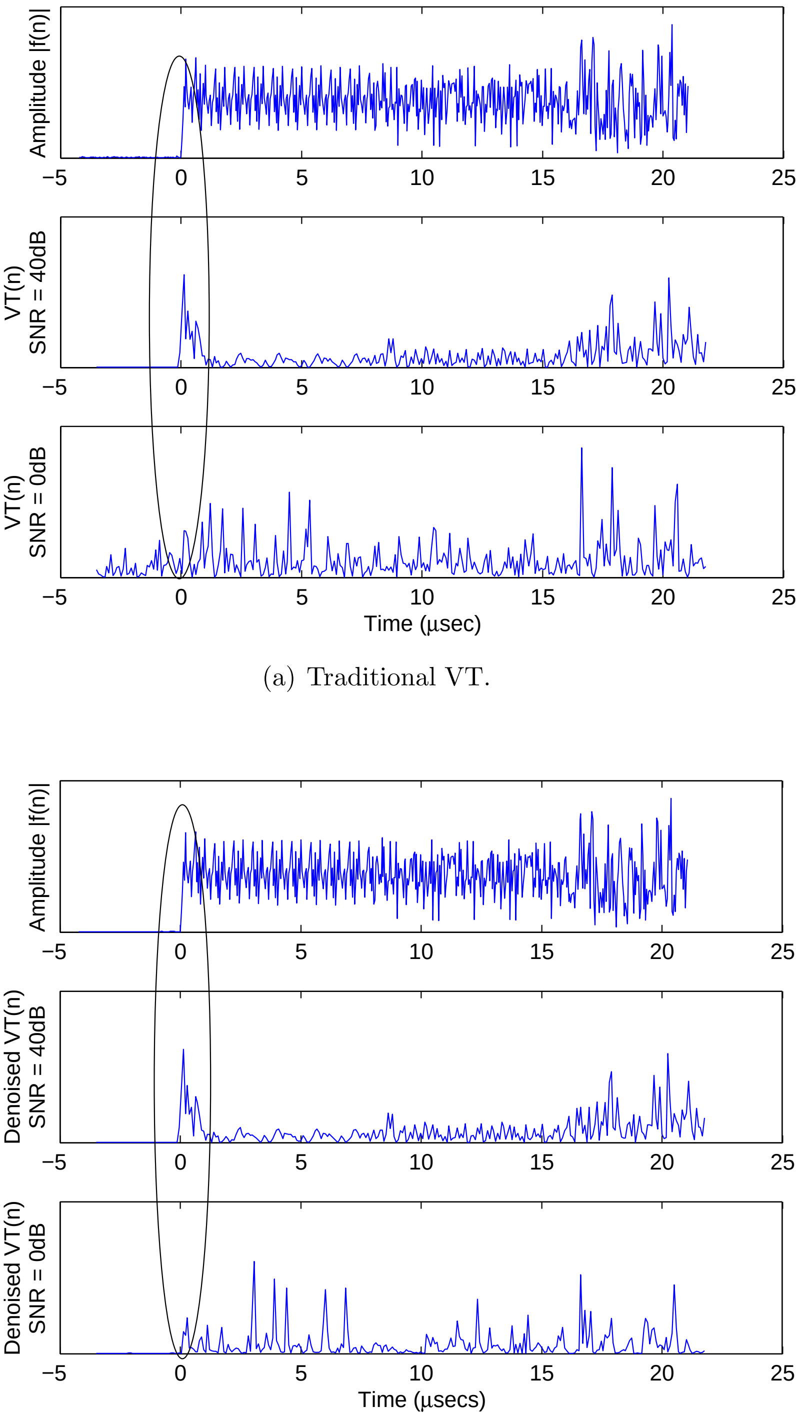
The impact of denoising is demonstrated by comparing Traditional VT results with Denoised VT results in Figure 3.8. Note that the circled region highlights the burst start location at $t = 0$. The representative amplitude response $|f(n)|$ is from an 802.11A burst at $SNR = 40$ dB and is identical in both figures. As a side note, the 16.0 μsec preamble response is clearly evident in the amplitude response. This burst was processed along with an $SNR = 0$ dB scaled version to generate the VT(n) results shown for each technique. As indicated, both techniques produce nearly identical

VT(n) responses at $SNR = 40$ dB with a clear distinct peak coinciding with the burst start time at $t = 0$. The effect of denoising is most evident in the $SNR = 0$ dB results by comparing the VT(n) responses just prior to $t = 0$, the actual burst start time. The $t < 0$ region of collected signals only contains background noise contributions. Upon close inspection of the $SNR = 0$ dB responses for $t < 0$, it is evident that DT-ℂWT denoising has effectively reduced the background noise response. While both VT(n) responses at $SNR = 0$ dB have a peak near $t = 0$, only the Denoised VT response has the desired step change response that is required for effective threshold detection and burst location.

3.5 Threshold Determination Process

Three distinct threshold values are required, including: 1) t_{Det} for coarse burst detection per Section 3.3.2.1, 2) t_{Loc} for burst location per Section 3.3.2.2, and 3) t_{DN} for denoising per Section 3.4. All SNR dependent threshold values were determined *a-priori* based on noise-only analysis using 100,000 AWGN realizations. The random noise realizations were generated, filtered, and scaled for the desired analysis SNR. For determining t_{Det} and t_{Loc} thresholds the resultant colored noise was analyzed using appropriate window parameters for a given technique. In determining t_{DN} for DT-ℂWT denoising, the resultant colored noise was transformed by the DT-ℂWT and coefficients retained for threshold determination. In all cases, results from the 100,000 noise-only iterations were histogrammed and the threshold value empirically chosen.

In selecting a t_{Loc} value for burst location, a trade-off is made between the number of early burst location estimates and the number of non-convergent solutions produced by the algorithm. The final t_{Loc} values were selected to ensure that both of these conditions are present and observable in the data. When comparing Traditional VT and Denoised VT performance, the t_{Loc} value is further constrained to provide a similar number of early burst location (10%) for both techniques to illicit a more fair comparison.



(a) Traditional VT.

(b) Denoised VT.

Figure 3.8: Instantaneous amplitude response of an 802.11A burst and (a) *Traditional VT* and (b) *Denoised VT* responses for $SNR = 40$ db and $SNR = 0$ dB. The circled region highlights the burst start location at $t = 0$. [31].

For DT-$\mathbb{C}$WT denoising, the value of t_{DN} is empirically chosen and based on the histogram bin value below which 95% of the noise-only values occur. The value of t_{Det} is chosen using conventional noise-only analysis of Probability of False Alarm (P_{fa}) and Probability of Detection (P_d) as represented on a Receiver Operating Characteristic (ROC) curve. Results of this analysis are reported in Section 4.2.1.

Summary

This chapter provided implementation details for Signal Collection and Post-Collection Processing, Statistical Fingerprint Generation, MDA/ML Signal Classification, DT-CWT Denoising and Threshold Determination. The results from implementing these processes are provided in Chapter 4.

IV. Results

This chapter provides modeling, simulation and analysis results that were generated using the processes detailed in Chapter 3. The research involved hundreds of simulations, requiring hundreds of hours of processing time in some cases. For brevity and to ensure succinctness, only a subset of representative results are presented from selected scenarios to fully support key research findings and contributions. Results for each contribution area introduced in Section 1.2 are presented in the following subsections: Bandwidth Sensitivity in Section 4.1, Burst Detection and Location in Section 4.2, MDA/ML Classification in Section 4.3, and Performance Sensitivity Analysis in Section 4.4.

4.1 Bandwidth Sensitivity

Prior to assessing burst detection and device classification performance, there was one important parameter that needed to be analyzed – the post-collection filter bandwidth (BW_{PC}). As shown in Figure 3.1 of Section 3.1, the collected burst responses and simulated noise are digitally filtered prior to forming the desired analysis signal. In previous related works using 802.11 signals, this filter bandwidth was simply fixed at a reasonable value based on common engineering practice [31, 33, 54, 55].

Intra-manufacturer classification accuracy using three Cisco devices is presented versus post-collection filter bandwidth in Figure 4.1 for both TD and WD techniques using 802.11A signals at $SNR = 40$ dB. While the best case WD classification performance is approximately 2% poorer than best case TD performance, the WD technique is more robust and classification performance varies by less than 2% over the range of bandwidths considered. The TD technique is much more sensitive and exhibits classification variation of nearly 6%, with poorest TD classification performance occurring at $BW_{PC} = 6.3$ MHz.

To highlight one potential cause for increased TD sensitivity, a few filter responses for different bandwidths are shown overlayed with a representative 802.11A signal PSD in Figure 4.2. The three filter bandwidths chosen for illustration include $BW_{PC} = 5.0$ MHz, $BW_{PC} = 6.3$ MHz (worst case TD classification performance), and $BW_{PC} = 7.7$ MHz. Of particular note is how each of the filters impact the 802.11A OFDM subcarrier response that exists near 7.5 MHz. Clearly, the $BW_{PC} = 7.7$ MHz filter effectively passes this carrier unaltered while each of the other two filters induce some degree of attenuation. This suggests there may be additional information in the higher frequency components that the MDA/ML classification process is more effectively exploiting. However, signal attenuation alone cannot account for all the TD performance differences in Figure 4.1 given that the poorest performing $BW_{PC} = 6.3$ MHz filter actually attenuates the 7.5 MHz carrier component less than the better performing $BW_{PC} = 5.0$ MHz filter (-5.0 dB versus -16 dB). Thus, the filter impact on noise (attenuation and spectral distribution) must be considered a contributing factor as well.

To enable comparison of both techniques at their best performance levels, a post-collection bandwidth of $BW_{PC} = 7.7$ MHz was used for generating all the subsequent burst detection and device classification results presented in Section 4.2 and Section 4.3, respectively. This particular bandwidth choice gives the TD technique an approximate 2% advantage in device classification. This will be considered when presenting, comparing and analyzing subsequent results.

4.2 Burst Detection and Location

This section discusses how traditional burst detection and burst start location techniques are sensitive to varying noise conditions and how this sensitivity impacts overall classification performance. Analysis indicates that improving the accuracy of burst detection and location can lead to improved device classification.

4.2.1 Burst Detection. Receiver Operating Characteristic (ROC) curves were generated using the process in Section 3.3.2.1 to characterize performance differences between the two burst detection techniques – Traditional VT and Denoised

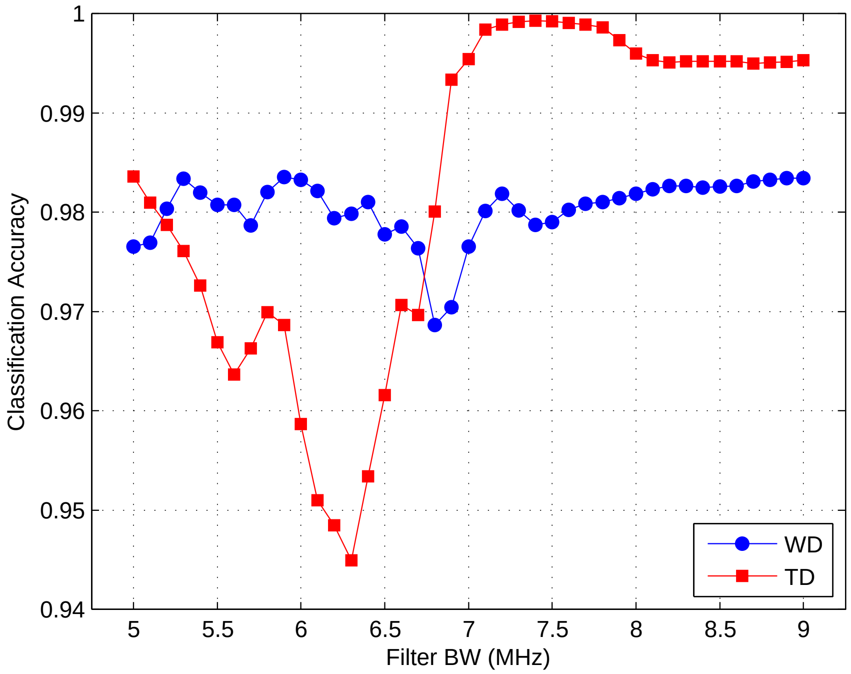


Figure 4.1: Intra-manufacturer classification accuracy versus post-collection filter bandwidth for TD and WD techniques using 802.11A signals at $SNR = 40$ dB.

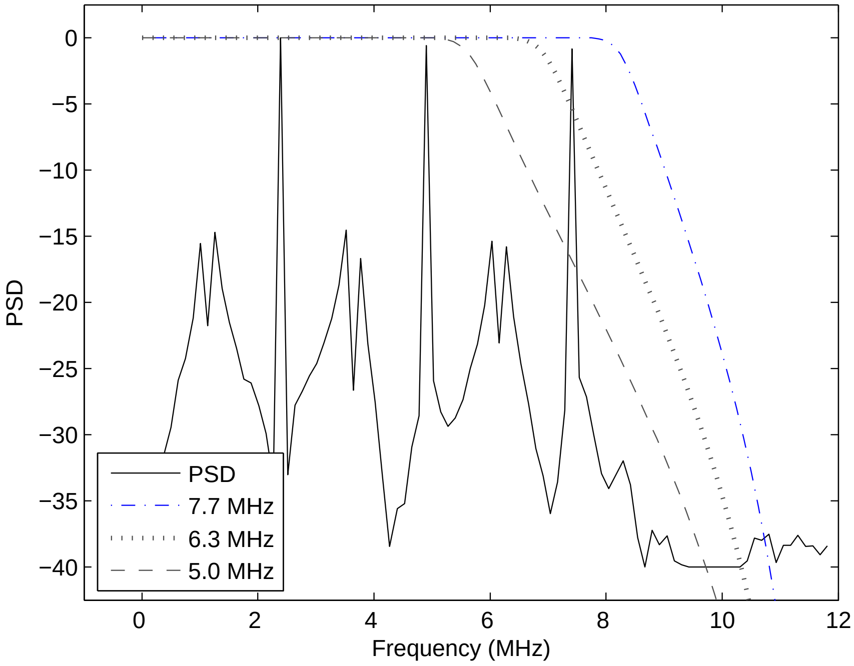


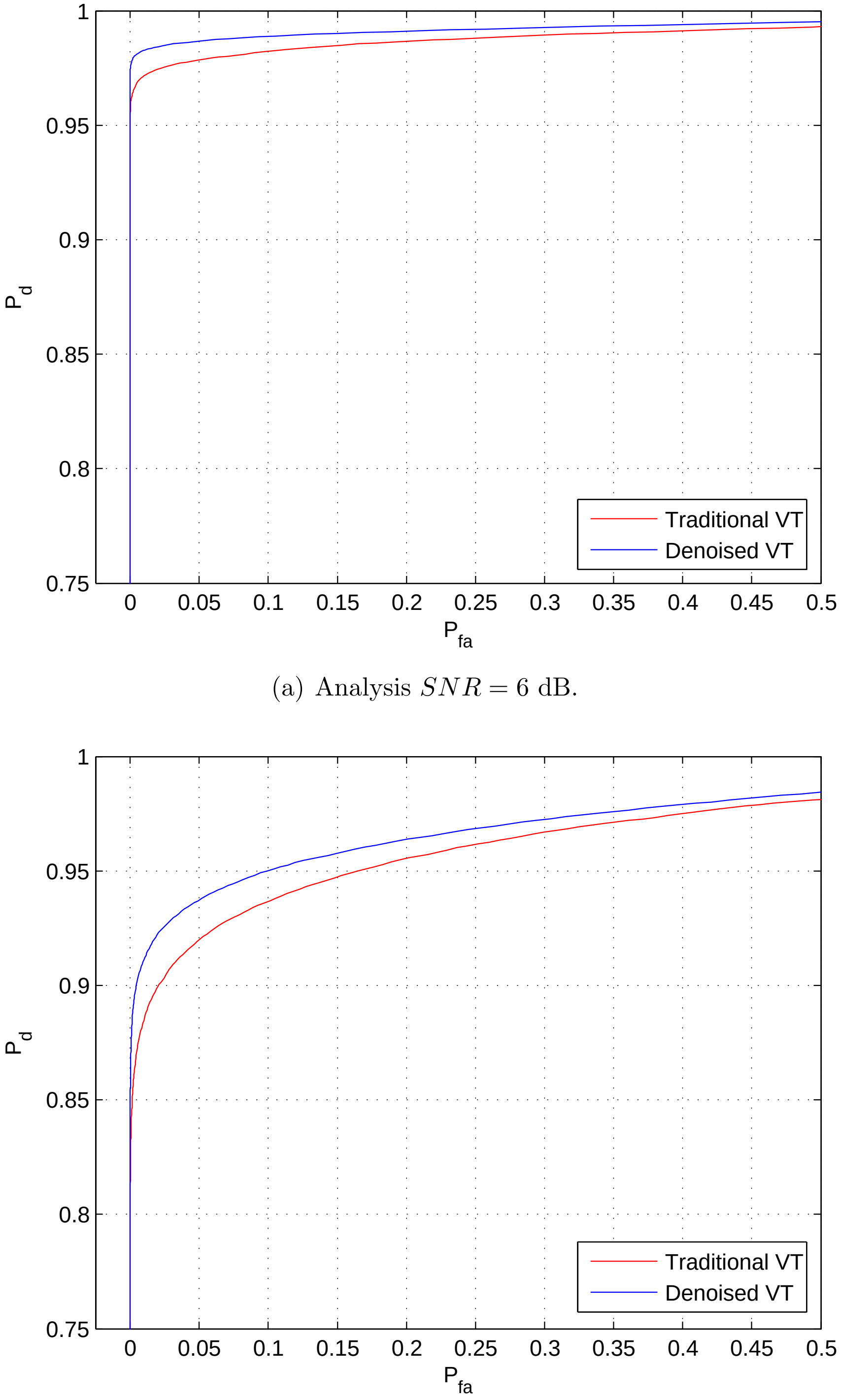
Figure 4.2: Overlay of representative 802.11A signal PSD and post-collection filter responses for various BW_{PC}.

VT. Results in Figure 4.3 show that at $SNR = 6$ dB and $SNR = 0$ dB, the Denoised VT technique provides a higher probability of detection (P_d) for a given probability of false alarm (P_{fa}). With a higher P_d for a given P_{fa}, the Denoised VT technique detects and outputs more bursts for subsequent processing when compared with the Traditional VT technique. With more bursts being detected and forwarded, it is possible to correctly classify the device in less time and have a higher confidence in the classification.

4.2.2 Burst Start Location. To isolate the effects of burst location accuracy from the effects of burst detection error, the 802.11A RF bursts were manually detected prior to burst location analysis. Thus, there is no noise-only data input to this process to generate false alarms and $P_d = 100\%$. All histogram results in this section share two common attributes, including: 1) the correct burst locations occur at $t = 0$ sec and 2) the default non-convergent solutions occur at $t = 16$ μsec (see Section 3.3.2.2 for discussion on non-convergent solutions).

4.2.2.1 Channel Noise Variability. These results illustrate the effect of channel noise variation for a given 802.11A RF burst and 200 AWGN realizations that are generated, filtered, scaled and added to achieve the desired analysis SNRs. Fractal-BSCD and Traditional VT estimation results are shown in Figure 4.4.

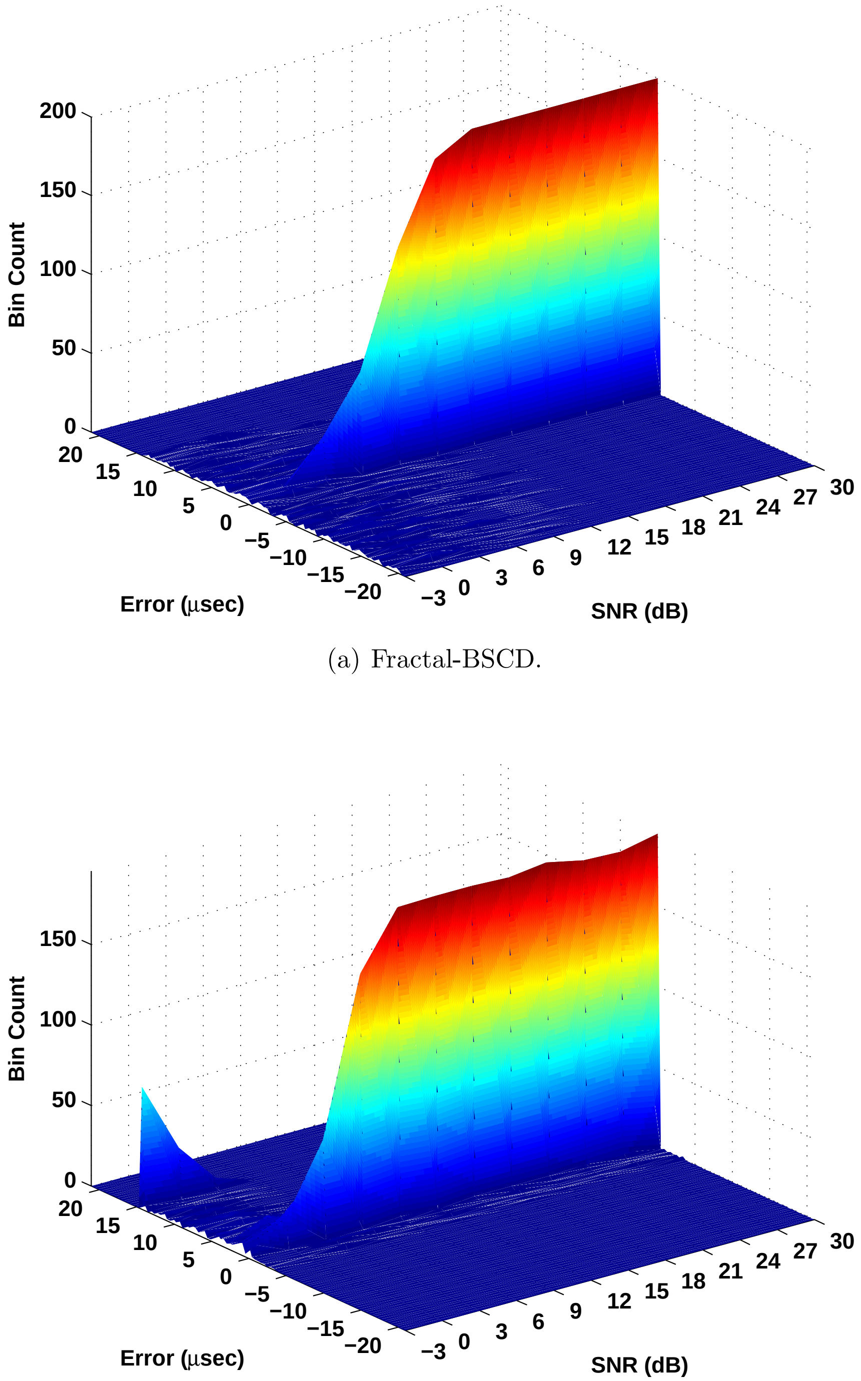
At higher SNRs the two methods perform similarly as the noise power varies, with primary differences beginning at $SNR = 9$ dB. Fractal-BSCD degradation is directly attributed to the *a-posteriori* PDF degradation, as calculated per (2.2) and shown in Figure 2.1. The strong peak response in the PDF diminishes and becomes more uniformly distributed as noise power increases. Traditional VT degradation is attributed to, and affected by, threshold selection criterion. For the non-optimum method implemented here, the threshold criterion is not always satisfied and a default start value is assigned – a *missed* detection or non-convergent solution. This is shown in Figure 4.4(b) as a peak forming at $t = 16$ μsec. The number of missed detections



(a) Analysis $SNR = 6$ dB.

(b) Analysis $SNR = 0$ dB.

Figure 4.3: Probability of False Alarm (P_{fa}) versus Probability of Detection (P_d) ROC curves for Traditional VT and Denoised VT techniques at (a) $SNR = 6$ dB and (b) $SNR = 0$ dB. [31].



(a) Fractal-BSCD.

(b) Traditional VT.

Figure 4.4: Impact of *Channel Noise Variation* on burst location error using (a) Fractal-BSCD and (b) Traditional VT. Histogram for 200 independently generated, filtered and scaled AWGN realizations with a given 802.11A RF burst [33].

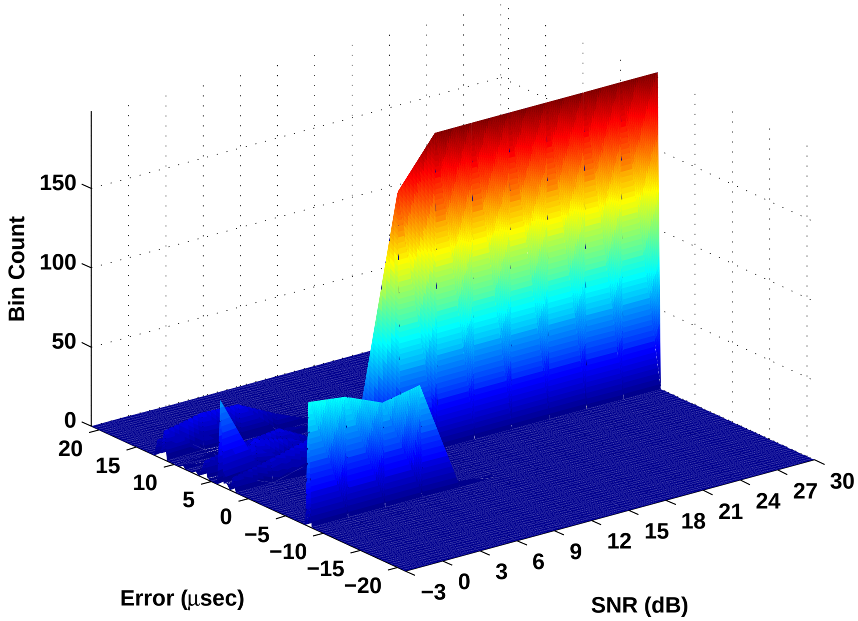
at lower SNRs can be reduced by changing the threshold. However, this also reduces estimation accuracy and precision at higher SNRs.

4.2.2.2 Burst-to-Burst Variability. These results illustrate the effect of burst-to-burst variation using a given AWGN realization that is generated, filtered and scaled to achieve the desired analysis SNRs. Results for 200 collected 802.11A bursts with Fractal-BSCD and Traditional VT estimation are shown in Figure 4.5.

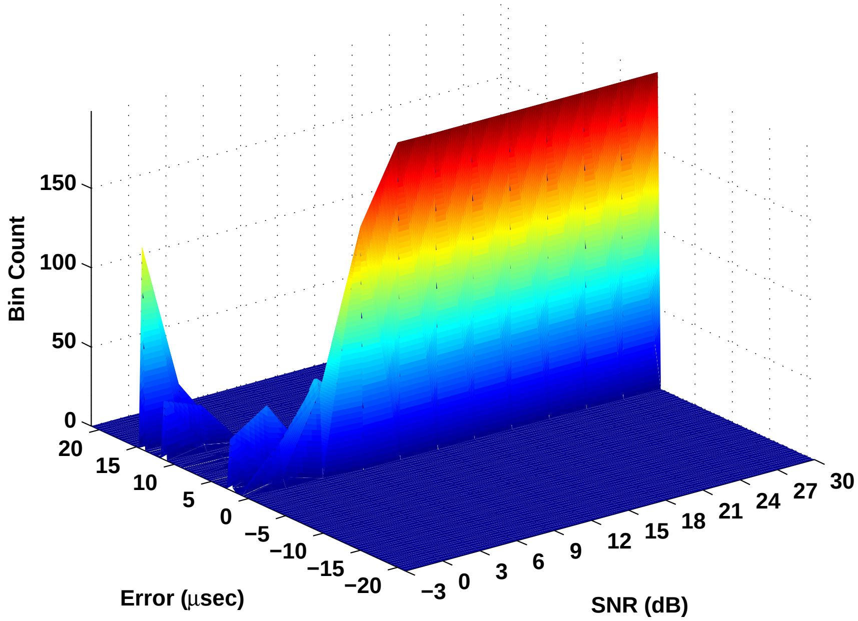
As with the channel noise impact, the two methods perform similarly at higher SNRs. Differences arise at lower SNRs, with the Traditional VT method degrading as before and producing missed detections. The missed detections are shown in Figure 4.5(b) as a peak forming at $t = 16$ μsec. The Fractal-BSCD response degrades differently than before, becoming multi-modal at lower SNRs and producing a significant number of detections in the noise-only portion of the signal. The modes are attributable to anomalous spikes in a specific noise realization. This is consistent with results in [23] and [68] given that BSCD processing is most effective when non-gradual parameter changes occur. At lower SNRs the amplitude change is too gradual in some bursts for the BSCD method to reliably detect them.

4.2.2.3 Combined Noise-Signal Variability. These results illustrate the combined effects of channel noise and burst-to-burst signal variability. In this case, 200 AWGN realizations were generated, filtered and scaled for each SNR and added to each of the 200 collected 802.11A bursts – a total of 40,000 unique AWGN realizations per SNR. Results for Fractal-BSCD and Traditional VT estimation are shown in Figure 4.6.

In this combined channel noise and burst-to-burst variability case, the channel noise effects are dominant. This is evident in that channel noise effect results in Figure 4.4 are nearly identical to the combined effects results Figure 4.6, including the missed detections shown in Figure 4.6(b) as a peak forming at $t = 16$ μsec. At higher SNRs the two methods perform similarly as the noise power varies, with

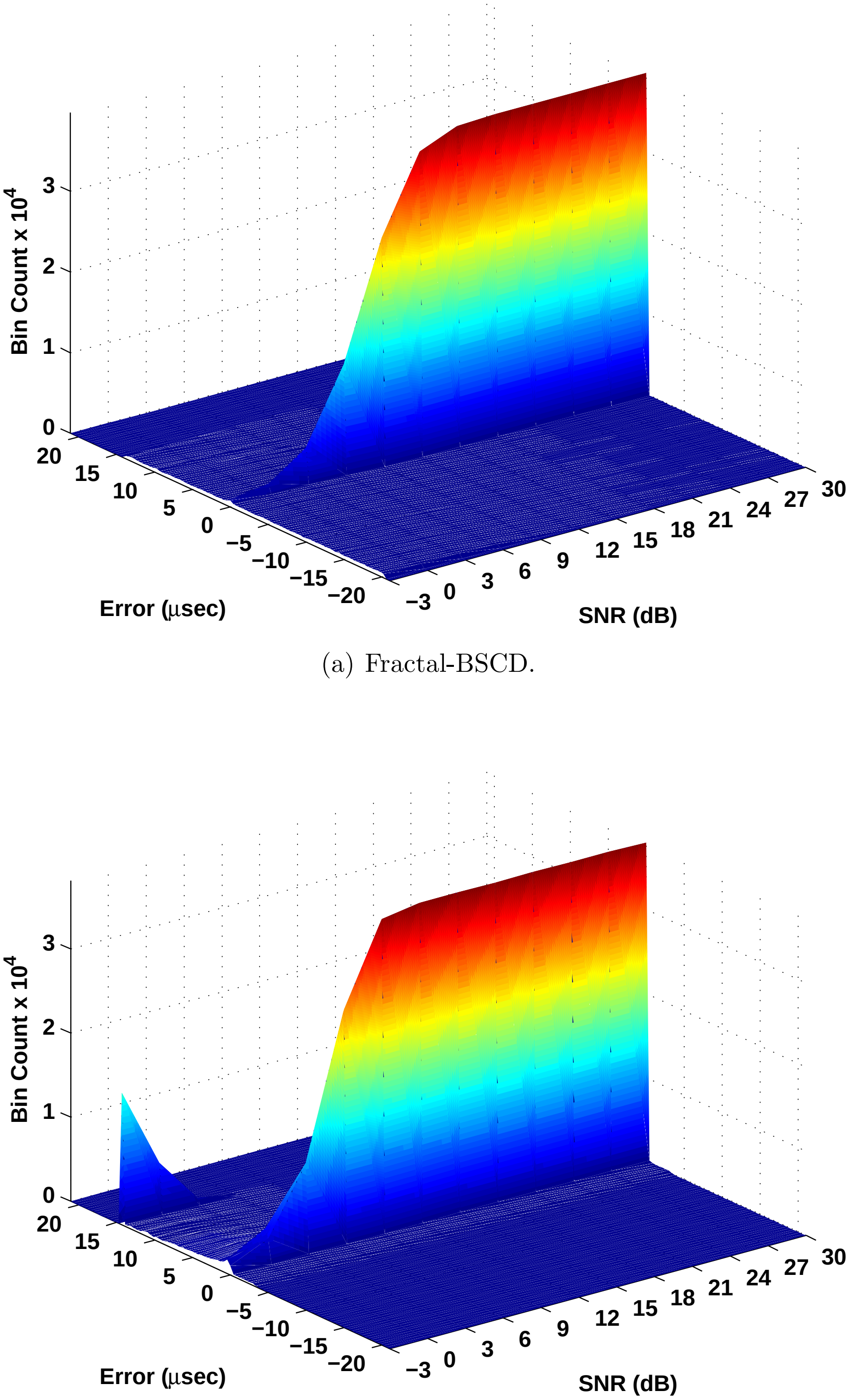


(a) Fractal-BSCD.



(b) Traditional VT.

Figure 4.5: Impact of *RF Burst Variation* on burst location error using (a) Fractal-BSCD and (b) Traditional VT. Histogram for 200 collected 802.11A RF bursts and one generated, filtered and scaled AWGN realization [33].



(a) Fractal-BSCD.

(b) Traditional VT.

Figure 4.6: Impact of *Combined Channel Noise and RF Burst Variation* on burst location error using (a) Fractal-BSCD and (b) Traditional VT. Histogram for 200 independently generated, filtered and scaled AWGN realizations and 200 collected 802.11A bursts [33].

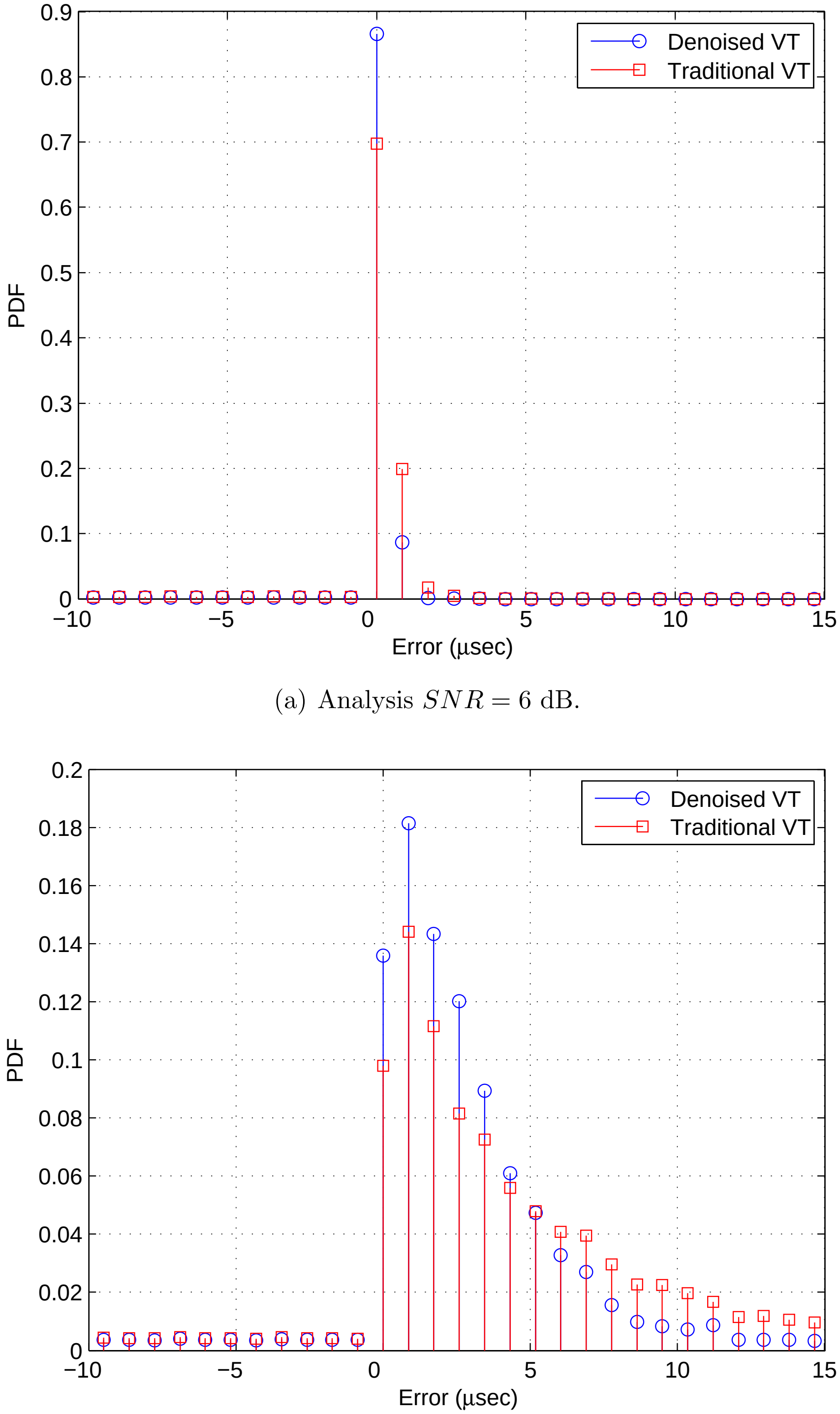
primary differences beginning at $SNR = 9$ dB. Fractal-BSCD degradation is directly attributed to the *a-posteriori* PDF degradation, as calculated per (2.2) and shown in Figure 2.1. The strong peak response in the PDF diminishes and becomes more uniformly distributed as noise power increases. Traditional VT degradation is attributed to, and affected by, threshold selection criterion.

Relative to Fractal-Bayesian Step Change Detector (Fractal-BSCD) technique, burst detection and location performance was best using a Traditional Variance Trajectory (Traditional VT) technique which provided results that were consistent with perfect burst estimation performance at higher SNRs ($10 \leq SNR \leq 30$ dB). However, performance for both techniques diverged at lower SNRs ($-3 \leq SNR \leq 10$ dB) [33]. This shortfall provided an impetus for subsequent burst detection research aimed at improved performance at low SNRs [31].

4.2.2.4 Combined Noise-Signal Variability: Denoised VT. As demonstrated in the previous sections, burst start location error for Fractal-BSCD and Traditional VT becomes symptomatic at $SNR \leq 9$ dB and there is room for improvement for the lower SNR range. In accordance with Section 3.4, the Denoised VT process consists of denoising the bursts with a DT-$\mathbb{C}$WT prior to calculating the Traditional VT.

Unlike results in Section 4.2.2.1 through Section 4.2.2.3 which were presented as 3-dimensional histograms in Figure 4.4 through Figure 4.6, the discernable differences in results of this section were not readily apparent when presented as 3-dimensional histograms. Thus, the results in this section are presented as 2-dimensional histograms for a given subset of SNRs considered. The results in Figure 4.7 show the improvement achieved at $SNR = 6$ dB and $SNR = 0$ dB when denoising is employed.

For the $SNR = 6$ dB results, the Denoised VT technique outperforms the Traditional VT technique by 1) correctly locating 24% more of the burst start locations while 2) experiencing a tighter distribution near the main peak response. Similar improvement is demonstrated for the $SNR = 0$ dB results. While both techniques



(a) Analysis $SNR = 6$ dB.

(b) Analysis $SNR = 0$ dB.

Figure 4.7: Probability Distribution Functions (PDF) for burst start location error using Traditional VT and Denoised VT at (a) $SNR = 6$ dB and (b) $SNR = 0$ dB [31].

experience a main peak that is late, the Denoised VT technique correctly locates 3.7% more of the burst start locations while also exhibiting a tighter distribution near the main peak response.

4.2.3 Error Impact on Device Classification. In an operational implementation, only those bursts causing location convergence according to Section 3.3.2.2 would be used for further processing. Therefore, for comparing classification performance only "dual convergent" bursts per Section 3.1 are used, i.e., only the bursts that result in a converged location solution from *both* techniques being evaluated. All other bursts that resulted in a converged location solution from only one of the techniques are excluded from subsequent classification. This approach was adopted based on early results which showed that singly convergent bursts unduly biased results in favor of the technique yielding the most converged solutions. The distribution differences (and their associated fingerprints) account for the only differences between the two techniques being processed by the classifier. Classification results in this section were generated using a mix of manufactured devices, including two from Cisco (N4U9 as Class A and N4UW as Class B) and one from Dell (BTA4 as Class C). Given the two Cisco devices are very close in serial number their discrimination inherently presents the greatest classification challenge.

4.2.3.1 Fractal-BSCD and Traditional VT Classification. Figure 4.8 shows average MDA-ML classification accuracy with the effects of Perfect, Fractal-BSCD and Traditional VT burst detection error included. In this case, Perfect results are obtained using a start location based on visual inspection of each collected burst. To determine if perfect burst location provides best possible MDA-ML classification accuracy, a uniform randomly distributed error was added to perfect start location estimates and results generated for comparison. As shown, the Perfect with Random Error results are consistent with Perfect results and marginally better/poorer for SNR below/above approximately 14 dB, respectively. With respect to the burst location estimation error impact to classification performance, the Traditional VT technique

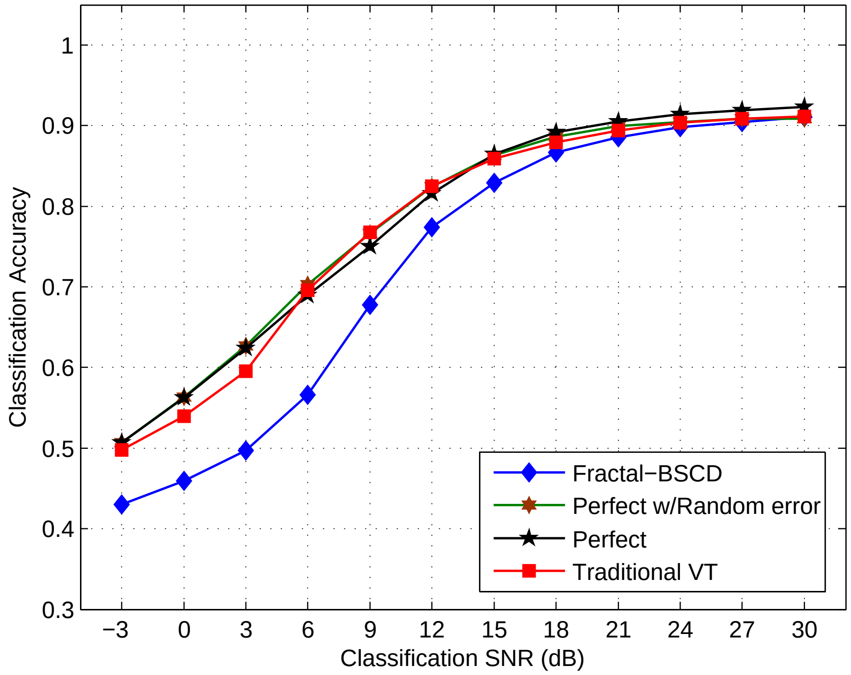


Figure 4.8: Average MDA-ML classification accuracy with Perfect, Fractal-BSCD and Traditional VT burst detection error included. [33].

was consistent with Perfect estimation for $6 \leq SNR \leq 30$ dB but under performed for $-3 \leq SNR \leq 6$ dB. Traditional VT also provided considerable improvement when compared with the Fractal-BSCD technique at lower $SNRs$ ($-3 \leq SNR \leq 18$ dB), i.e., for a given classification accuracy in the range of 50%–80% the required SNR for Traditional VT is 3-6 dB lower than what is required for Fractal-BSCD.

Classification performance is commonly illustrated using a confusion matrix that shows the percentage of time a particular input class is estimated as one of the possible classes, with the diagonal entries representing correct classification. Table 4.1 shows the classification confusion matrix for perfect burst location results in Figure 4.8 at $SNR = 30$ dB. As indicated by off-diagonal entries, the greatest confusion exists in intra-manufacturer classification with Class A and Class B inputs being mostly confused with each other. The Class B input is errantly classified as Class C a small percentage of the time and the Class C input experiences no confusion. Collectively,

Table 4.1: Classification confusion matrix for perfect burst location results in Figure 4.8 at $SNR = 30$ dB.

	Class Estimate		
Input Class	A	B	C
A	**89.5%**	10.5%	0.0%
B	10.0%	**89.5%**	0.5%
C	0.0%	0.0%	**100.0%**

these results illustrate that the most stressing classification challenge is posed for intra-manufacturer discrimination (the two Cisco devices).

4.2.3.2 Traditional VT and Denoised VT Classification. To assess the impact of DT-ℂWT denoising, Denoised VT classification results were generated for comparison. These results are presented in Figure 4.9 which shows average MDA-ML classification accuracy with the effects of Perfect, Traditional VT, and Denoised VT burst detection error included. As before, the Perfect results provide an upper bound on achievable performance. As indicated, Traditional VT and Denoised VT performance is similar for $SNR > 6$ dB and $SNR < -2$ dB. For $-1 < SNR < 5$ dB, the Denoised VT technique outperforms the Traditional VT technique and provides an average improvement in classification accuracy of 1.75%. Relative to results for perfect burst detection and location, the Denoised VT process achieves nearly 34% of the available performance improvement–when used with MDA/ML processing, there is little more to be gained in overall classification performance by improving burst detection and location accuracy.

Confusion matrix results for the $SNR = 3$ dB data points in Figure 4.9 are shown in Table 4.2. Two things are evident when comparing Traditional VT and Denoised VT results, including: 1) minimal difference in Class A and Class B performance, and 2) greatest improvement occurring in correctly classifying Class C which exhibits a 6% increase. These results are consistent with what is expected when considering “What level of improvement is achievable?” Assuming Perfect results represent an upper bound, achievable improvement is determined by comparing di-

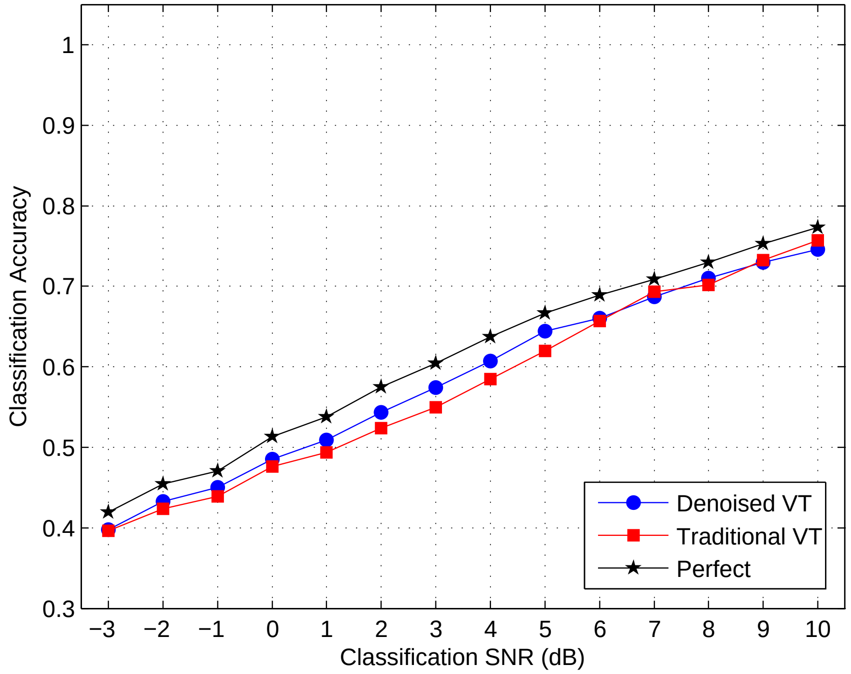


Figure 4.9: Average MDA-ML classification accuracy with Perfect, Traditional VT and Denoised VT burst detection error included. Results obtained for "dual convergent" 802.11A bursts from a mix of Cisco-Cisco-Dell devices [31].

agonal entries in Table 4.2 for Perfect and Traditional VT techniques. For Class A and Class B devices, there is only a 1%-2% margin for improvement in correct classification. However, there is a 12% margin for improvement in Class C classification. Thus, the Denoised VT performance improvement of 6% for Class C represents 50% of the possible improvement.

4.3 MDA/ML Device Classification

As concluded in Section 4.2.3.2 and highlighted by results in Figure 4.9, there is minimal additional improvement that can be made in end-to-end device classification by considering alternative burst location techniques. The reader is reminded here that the focus of this research is on proof-of-concept demonstration without optimization for real-time implementation. Thus, there may be alternate burst detection techniques that are more computationally efficient than those considered here. However, their

Table 4.2: MDA/ML classification confusion matrix for various burst detection methods at $SNR = 3$ dB [31].

Perfect	Class Estimate		
Input Class	A	B	C
A	**68%**	21%	11%
B	31%	**44%**	25%
C	14%	17%	**69%**

Traditional VT	Class Estimate		
Input Class	A	B	C
A	**67%**	22%	11%
B	31%	**42%**	27%
C	22%	21%	**57%**

Denoised VT	Class Estimate		
Input Class	A	B	C
A	**67%**	21%	12%
B	30%	**43%**	27%
C	18%	19%	**63%**

application to the RF fingerprinting process detailed in Figure 3.1 of Section 3.1 is beyond the scope of this research and remains an area of future research.

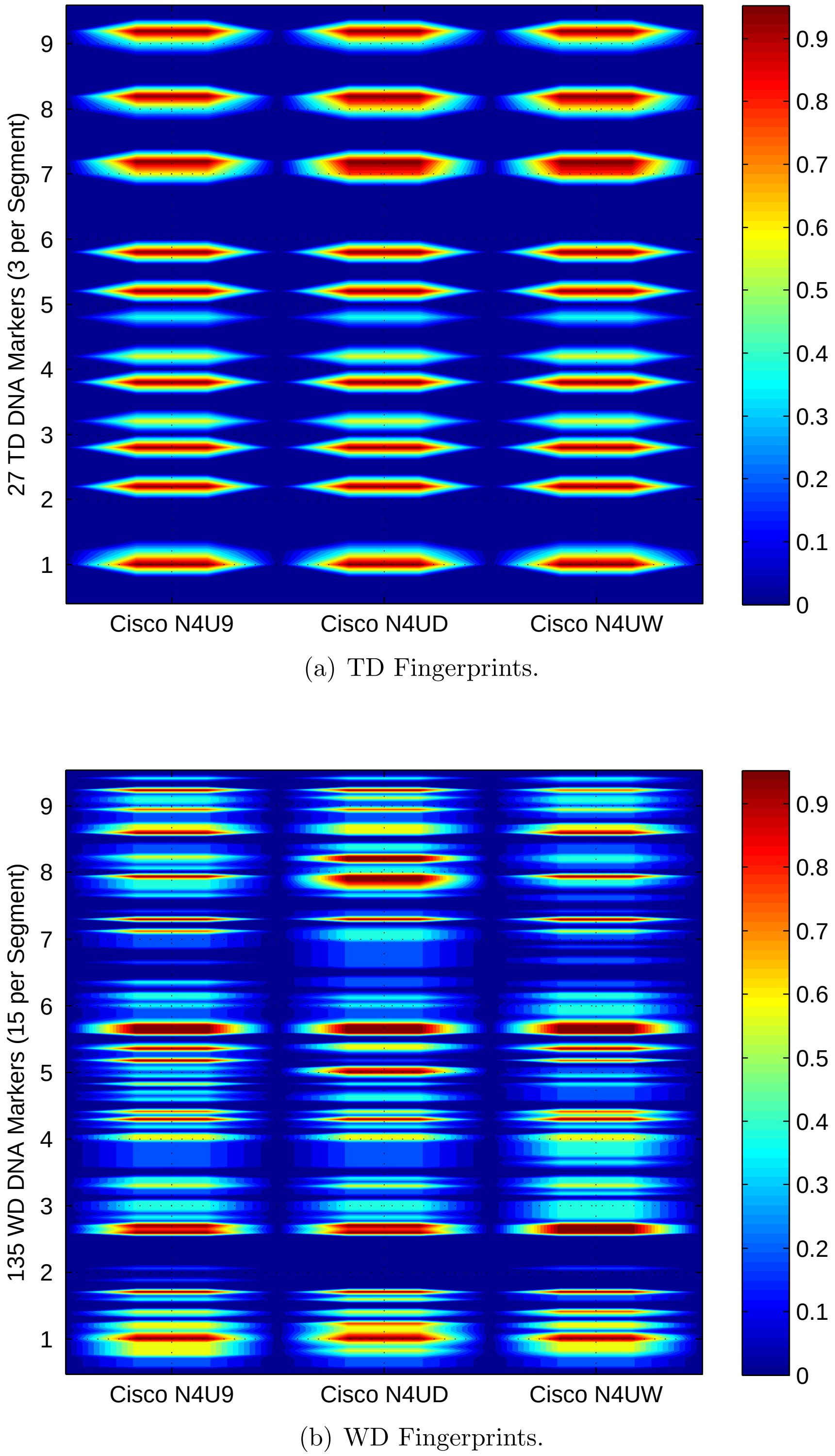
Given the burst detection capability detailed in Section 4.2.3.2, and the inherent robustness of the MDA/ML classification process described in Section 3.3.4, the research emphasis shifted toward improving device classification by considering alternate RF fingerprint features. More specifically, the DT-ℂWT process in Section 2.3, that was used for Denoised VT burst detection, was next used for generating fingerprints according to Section 3.3.3. The incorporation of a DT-ℂWT prior to statistical feature calculation is functionally illustrated in the RF fingerprinting process depicted in Figure 3.3. For comparative assessment and clarity of presentation in this section, results based on DT-ℂWT fingerprints are referred to as Wavelet Domain (WD) results while all other results, including all those presented in previous sections, are referred to as Time Domain (TD) results.

Various combinations of device manufacturers (Cisco, Netgear, Linksys, and AirMagnet) and signals (802.11A and 802.11G) are considered for demonstration with specific stressing cases considered and analyzed. Using three Cisco devices, classification results are generated and analyzed to demonstrate serial number discrimination. This is the most stressing case considered and is denoted throughout as "intra-manufacturer" discrimination. Using a combination of devices from various manufacturers, classification results are generated and analyzed to demonstrate what is denoted as "inter-manufacturer" discrimination. For comparative analysis, results are generated using TD and WD fingerprints generated from *identical* collected signals with *identical* Monte Carlo noise realizations that are appropriately filtered and scaled to achieve desired analysis SNRs. This enables a one-to-one comparison of TD and WD classification results, with a performance "gain" defined as the difference in required SNR, expressed in dB, at a given classification accuracy level. This is analytically expressed as $SNR_{WD} - SNR_{TD}$ at a given classification performance. For tracking performance improvement and/or degradation throughout this section of the document, the performance gain at an 80% classification accuracy level is used per "reasonable" criteria detailed in Section 1.2.1 and is shown in the figures as a circled region.

4.3.1 Statistical Fingerprint Features. The ability to visualize fingerprint features can be insightful for both feature selection and performance analysis. Two important properties that fingerprints should posses to increase overall classification performance are uniqueness and temporal/spectral stability. Greater fingerprint uniqueness across devices provides greater separability and improved classification performance. Temporal and spectral stability of fingerprint features is also important, especially for the MDA/ML training and classification process. Ideally, the statistical fingerprint features used for MDA/ML training and classification do not differ significantly. Given the signal collection conditions used for this research, the temporal and spectral stability of fingerprint features is nearly the best that can be expected. The 2000 bursts used for all of the results presented here were collected over a relatively short time interval (typically less than 0.5 sec) and in an anechoic chamber void of multipath and channel fading effects. The uniqueness of fingerprint statistical features and degree of temporal stability can be illustrated using what are called "Distinct Native Attributes" (DNA) in RF Fingerprint DNA plots.

The uniqueness of fingerprint statistical features is illustrated in Figure 4.10 and Figure 4.11. These RF DNA plots were generated by randomly selecting 250 collected bursts for each device, scaling them to achieve $SNR = 20$ dB, and averaging the corresponding statistical fingerprints. For visual clarity, the average fingerprint features are normalized within each segment where the y-axis segment numbers correspond to the nine statistical measures defined in (3.1) and (3.3). The number of DNA markers per segment is different for TD and WD fingerprints. For TD fingerprints, the number markers is a function of the number of signal regions used for fingerprint generation as expressed in (3.2). For WD fingerprints, the number of markers is a function of the number of signal regions and DT-$\mathbb{C}$WT levels used for fingerprint generation as expressed in (3.4). The RF fingerprints in Figure 4.10 are from one manufacturer (Cisco) and typical of what is used for *intra-manufacturer* discrimination. The RF fingerprints in Figure 4.11 are from three different manufacturers (Cisco, Linksys and Netgear) and are typical of what is used for *inter-manufacturer* discrimination. Two conclusions are readily apparent by analyzing results in Figure 4.10 and Figure 4.11, including: 1) relative to *intra-manufacturer* fingerprint features, the *inter-manufacturer* fingerprint features exhibit greater uniqueness across devices, and 2) relative to TD fingerprints, the WD fingerprint features exhibit greater uniqueness across devices. Subsequent results in this chapter show that greater uniqueness translates to better overall classification performance.

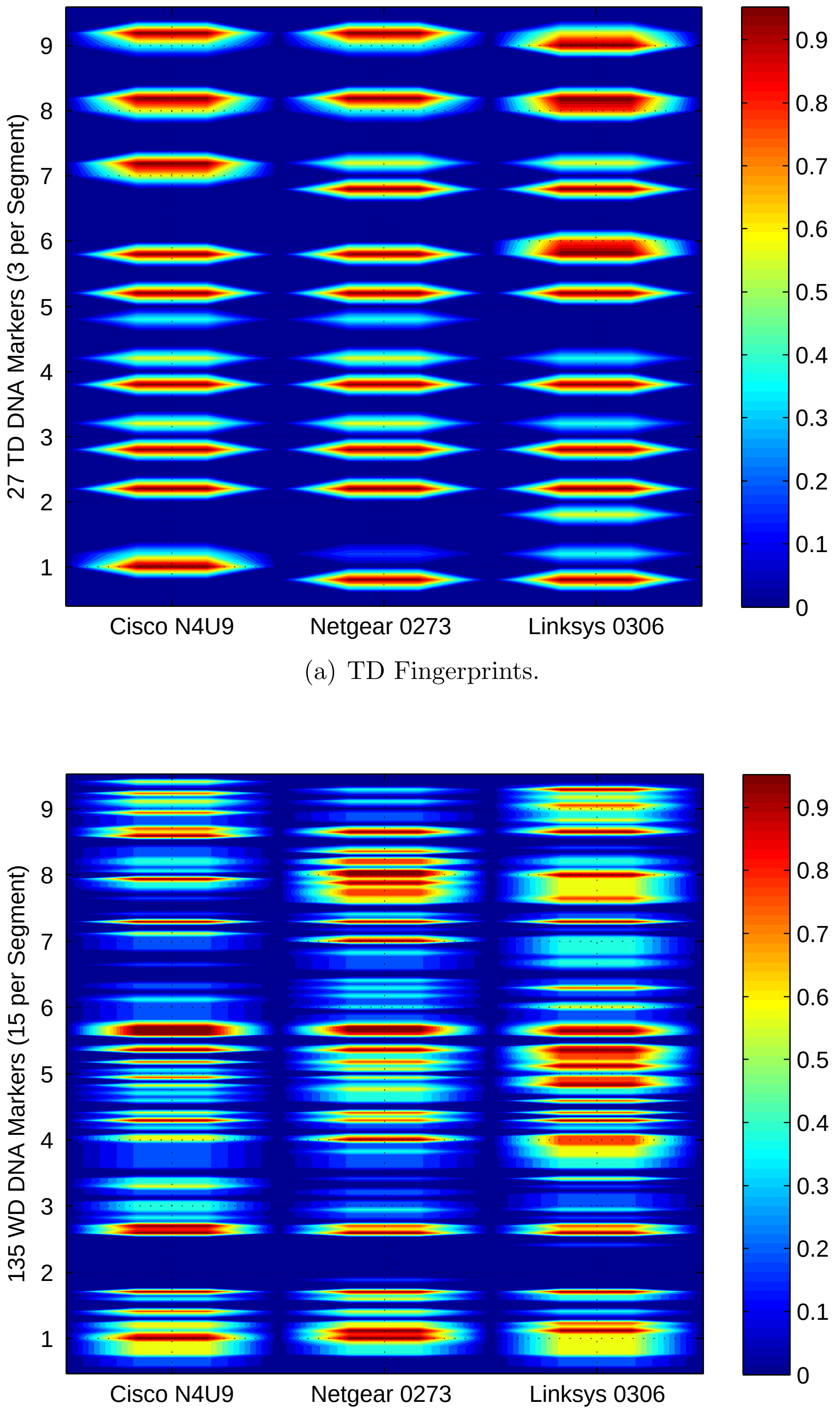
The temporal stability of fingerprint features is demonstrated in Figure 4.12 through Figure 4.14. These RF DNA plots were generated by randomly selecting 25 collected bursts for each device, scaling them to achieve $SNR = 20$ dB, and generating the corresponding fingerprint for each. As before, the fingerprint features



(a) TD Fingerprints.

(b) WD Fingerprints.

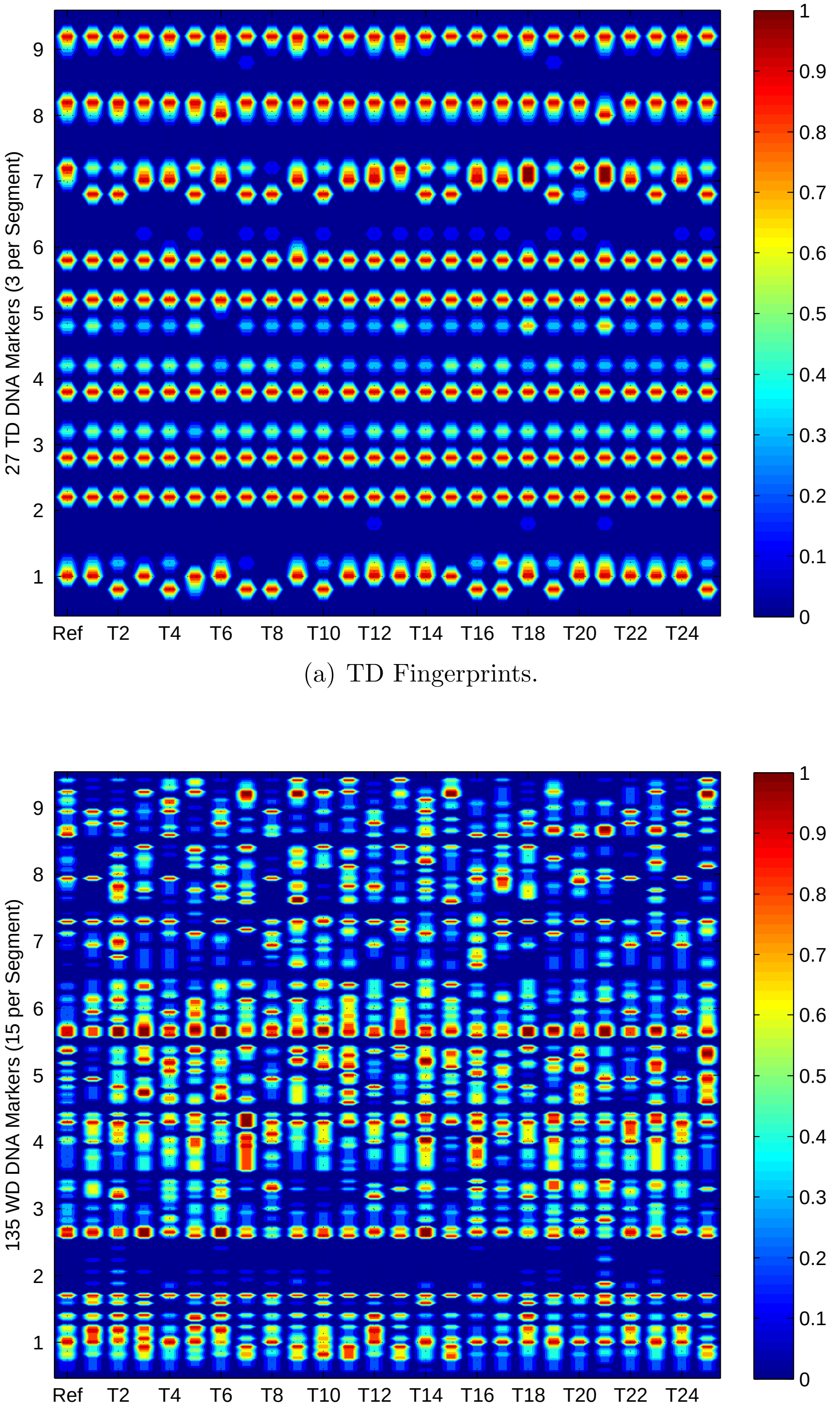
Figure 4.10: Intra-manufacturer average RF fingerprint DNA plots showing (a) TD and (b) WD fingerprints based on 250 randomly selected bursts at $SNR = 20$ dB.



(a) TD Fingerprints.

(b) WD Fingerprints.

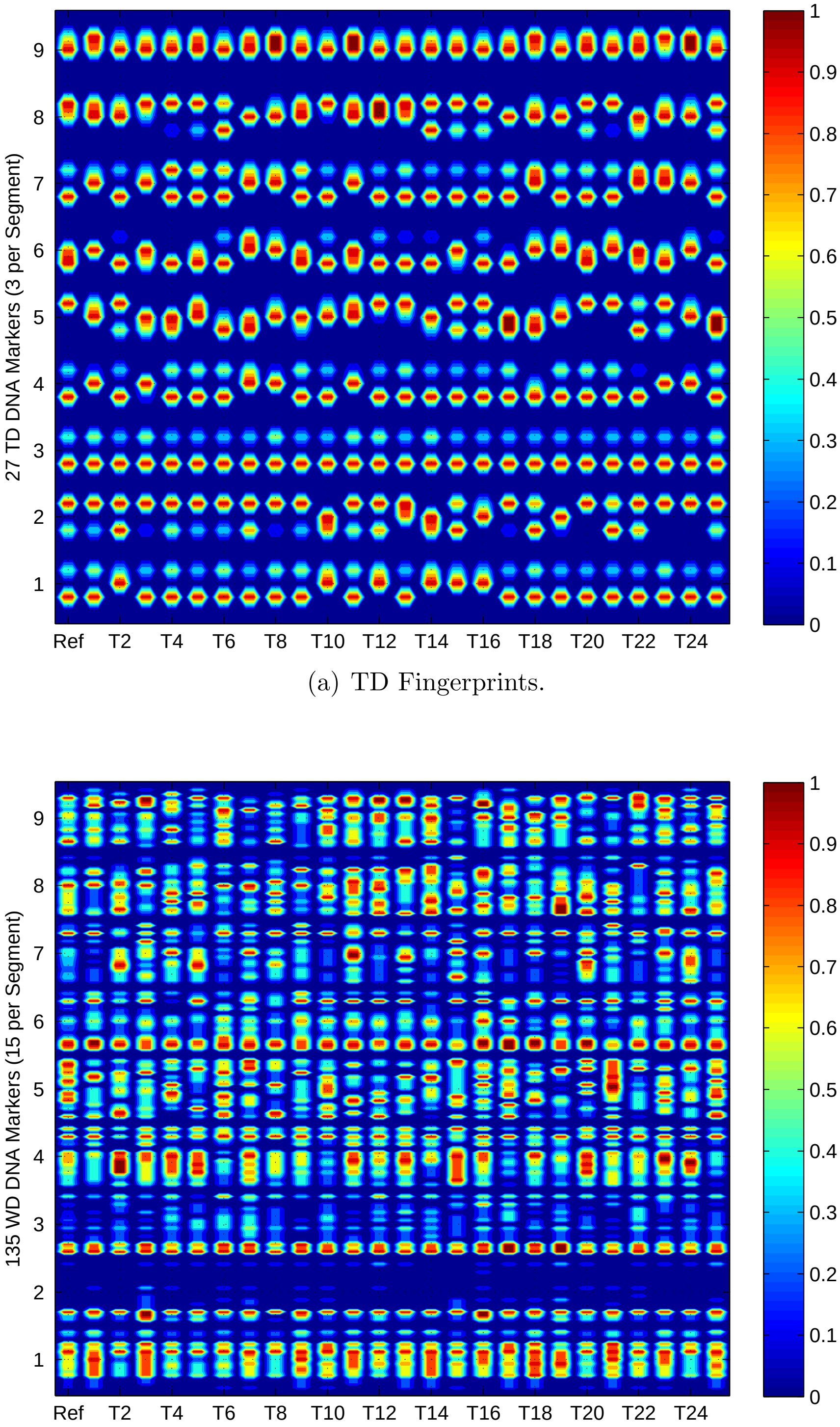
Figure 4.11: Inter-manufacturer average RF fingerprint DNA plots showing (a) TD and (b) WD fingerprints based on 250 randomly selected bursts at $SNR = 20$ dB.



(a) TD Fingerprints.

(b) WD Fingerprints.

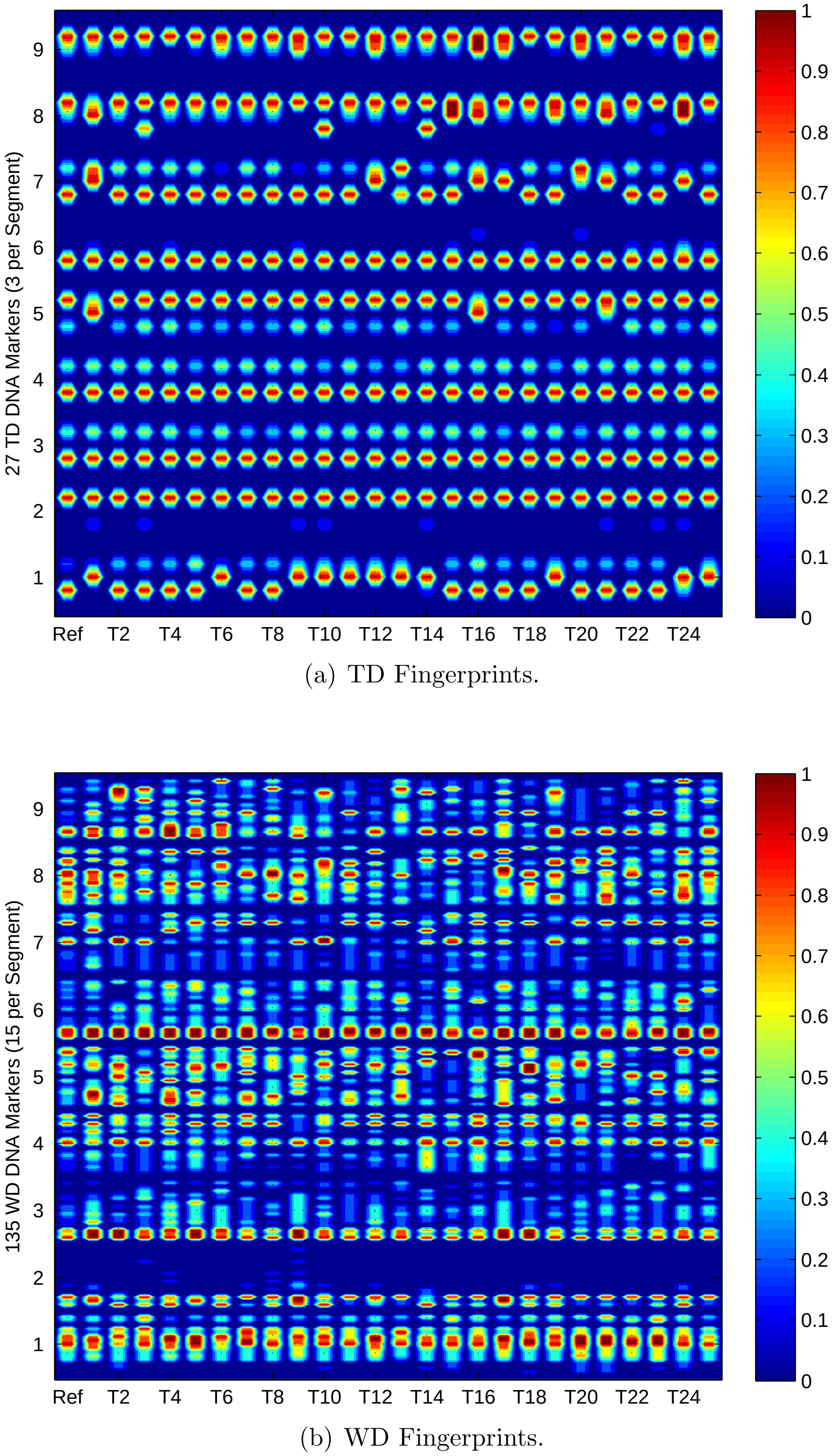
Figure 4.12: Temporal TD Fingerprint Stability: (a) TD and (a) WD Fingerprints for 25 randomly selected bursts from Cisco N4U9 device at $SNR = 20$ dB.



(a) TD Fingerprints.

(b) WD Fingerprints.

Figure 4.13: Temporal TD Fingerprint Stability: (a) TD and (b) WD Fingerprints for 25 randomly selected bursts from Linksys 0306 device at $SNR = 20$ dB.



(a) TD Fingerprints.

(b) WD Fingerprints.

Figure 4.14: Temporal TD Fingerprint Stability: (a) TD and (b) WD Fingerprints for 25 randomly selected bursts from Netgear 0273 device at $SNR = 20$ dB.

are normalized within each segment for visual clarity. The left-most “Ref” fingerprints are the corresponding average reference fingerprints taken from Figure 4.10 and Figure 4.11 as appropriate. These are provided for comparison with the randomly selected test “T” fingerprints which are presented sequentially with increasing time order. Note that the 25 randomly selected test bursts are different than the 250 bursts used to generate the average reference fingerprint. The results in Figure 4.12 through Figure 4.14 clearly illustrate a dissimilar degree of stability among the fingerprint features being used. Note that the effects of temporal stability on the overall MDA/ML classification is outside the scope of this work and is reserved for future research.

4.3.2 TD vs. WD Performance: 802.11A Signals. Intra-manufacturer classification is demonstrated using four Cisco devices transmitting an 802.11A signal, with results presented for all permutations of devices as shown in Table 4.3. Subsequent intra-manufacturer discrimination is then demonstrated using Permutation #1 which presents the “most stressing” conditions for classification. As indicated in Table 4.3 the most stressing permutation uses three Cisco devices having serial numbers that differ in only the last digit. Thus, it is assumed that these devices have been manufactured using identical components, from identical lots, with identical processes, under identical environmental conditions. Thus, discriminating between these devices presents the most stressing case for classification.

Sensitivity to serial number variation is illustrated in Figure 4.15 which shows intra-manufacturer classification results for all four permutations. The mean across

Table 4.3: 802.11A Cisco intra-manufacturer permutations.

	Serial Number			
Perm	N4U9	N4UD	N4UW	N4PX
1	×	×	×	
2		×	×	×
3	×		×	×
4	×	×		×

all four permutations is shown by the filled markers. The resuls for both TD and WD techniques show that Permutation #1 and Permutation #3, which both include Cisco devices with serial numbers N4U9 and N4UW, present the most stressing cases and yield the poorest results for nearly all SNR values considered. As with all previous results, Permutation #1 is the most stressing case at 80% classification accuracy.

The mean classification results in Figure 4.15 are presented again in Figure 4.16 for closer inspection. While both techniques perform similarly at $SNR \geq 25$ dB, the WD fingerprinting technique outperforms the TD technique at the lower SNRs. As highlighted in the circled region, the WD fingerprints achieve 80% classification accuracy at $SNR \approx 11$ dB. This represents a gain of approximately 7 dB with respect to equivalent TD fingerprinting performance.

Classification confusion matrices are presented in Table 4.4 for Permutation #1 of the Cisco devices for signals at $SNR = 11$ dB. As indicated in the lower comparison matrix, WD fingerprinting provides improved classification performance across all three classes, with the greatest improvement of 28.1% obtained in correctly classifying Class B. One common result with both fingerprinting techniques is that Class A and Class C devices are more confused with each other and confused less often with Class B. With respect to the device serial numbers, Class A and Class C are closer to each other than either one is to Class B.

Inter-manufacturer classification is demonstrated using two devices each from Cisco, Netgear, and Linksys transmitting an 802.11A signal, with results presented for device permutations shown in Table 4.5. Average classification performance across all device permutations are shown in Figure 4.17 for both TD and WD fingerprinting. While both techniques perform similarly at $SNR \geq 20$ dB, the WD fingerprinting technique outperforms the TD technique at the lower SNRs. As highlighted in the circled region, the WD fingerprints achieve 80% classification accuracy at $SNR \approx 2$ dB. This represents a gain of approximately 5 dB with respect to equivalent TD fingerprinting performance.

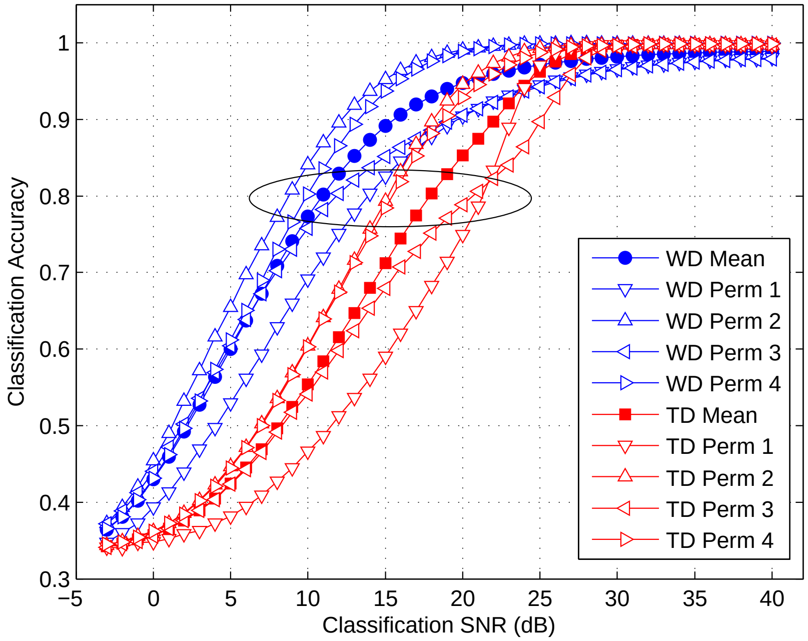


Figure 4.15: Intra-manufacturer MDA/ML classification: Average performance for all four permutations of four Cisco devices transmitting 802.11A signals.

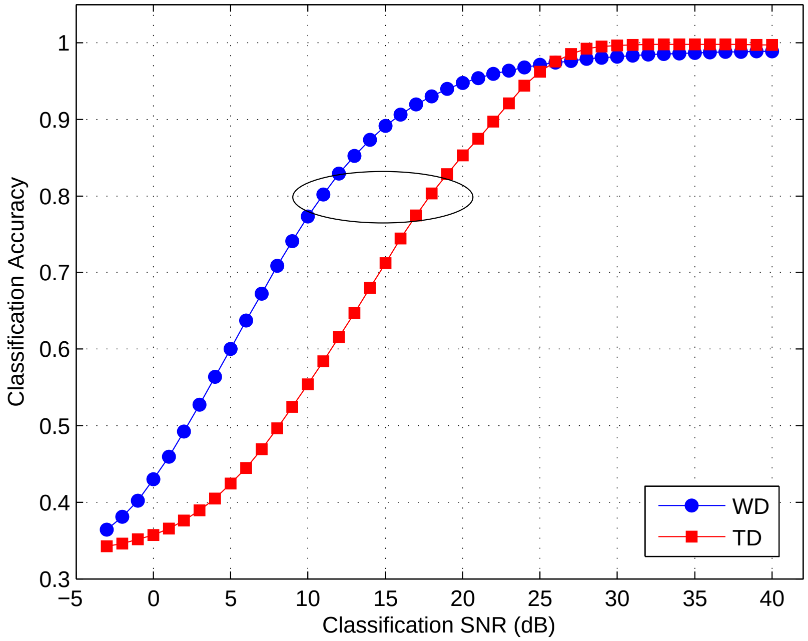


Figure 4.16: Intra-manufacturer MDA/ML classification: Average performance across four permutations of four Cisco devices transmitting 802.11A signals.

Table 4.4: Intra-manufacturer confusion matrices for TD and WD fingerprinting: Permutation #1 from Table 4.3 with 802.11A signals at $SNR = 11$ dB. The difference in performance between the two techniques is provided for comparison.

TD	Class Estimate		
Input Class	A	B	C
A	**49.4%**	17.3%	33.3%
B	18.5%	**65.9%**	15.6%
C	34.2%	12.1%	**53.6%**

WD	Class Estimate		
Input Class	A	B	C
A	**69.5%**	5.9%	24.5%
B	5.3%	**94.0%**	0.7%
C	21.5%	1.3%	**77.2%**

WD – TD	Class Estimate		
Input Class	A	B	C
A	**20.1%**	-11.4%	-8.8%
B	-13.2%	**28.1%**	-14.9%
C	-12.7%	-10.8%	**23.6%**

Classification confusion matrices are presented in Table 4.6 for Permutation #1 of the Cisco, Netgear and Linksys devices at $SNR = 2$ dB. Given similar results were obtained for all permutations considered, only the results for one permutation are presented given the conclusions drawn are generally applicable to the other permutations. While the WD technique increases classification performance for Cisco (Class A) and Netgear (Class B) devices, there is a decrease in Linksys (Class C) classification performance as indicated by the negative diagonal entry in the lower matrix. The greatest improvement of 30.2% is obtained in correctly classifying Class A. Unlike the intra-manufacturer discrimination where the classes are similarly confused regardless of the fingerprint technique, the inter-manufacturer cross-class confusion is different. The TD fingerprints experienced the most confusion between Class A and Class B, while the WD fingerprints showed the greatest confusion between Class B and Class C. This difference accounts for the greater improvement that occurs with Class A.

Table 4.5: 802.11A Inter-manufacturer permutations.

	Cisco		Netgear		Linksys	
Perm	N4U9	N4UD	0273	0217	0306	0307
1	×		×		×	
2		×		×		×

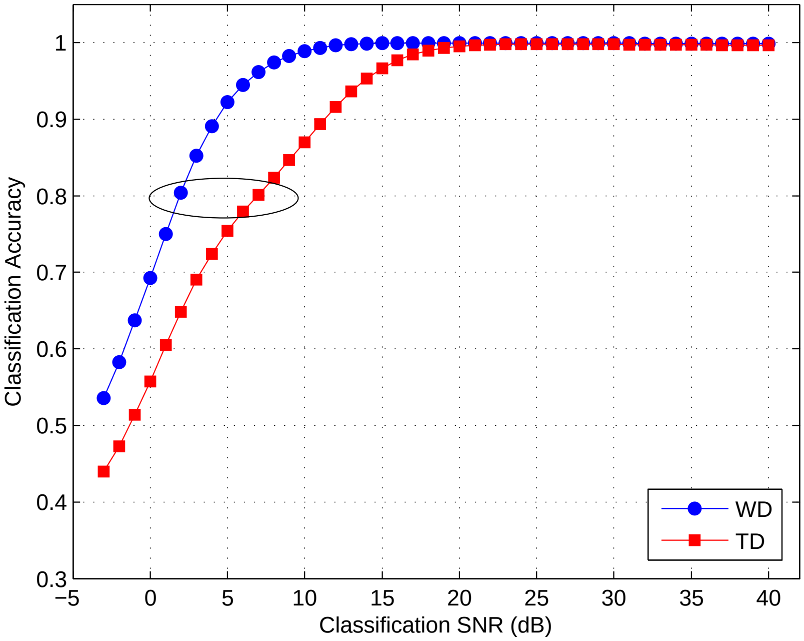


Figure 4.17: Inter-manufacturer MDA/ML classification: Average performance across Cisco, Netgear and Linksys devices transmitting 802.11A signals.

4.3.3 TD vs. WD Performance: 802.11G Signals. To demonstrate that the classification results presented up to this point are not unique to the 802.11A signal, the RF fingerprinting and classification process was applied to an additional OFDM-based signal to demonstrate broader applicability. This was easily accomplished using the same serial-numbered devices as used previously by operating them in an 802.11G signaling mode.

Using the same four Cisco devices (as in Section 4.3.2) transmitting an 802.11G signal, intra-manufacturer discrimination is conducted with the two permutations shown in Table 4.7. Figure 4.18 shows average intra-manufacturer classification performance across the two permutations of Cisco devices for TD and WD fingerprinting.

Table 4.6: Inter-manufacturer confusion matrices for WD and TD fingerprinting: representative permutation of devices with 802.11A signals at $SNR = 2$ dB. The difference in performance between the two techniques is provided for comparison.

TD	Class Estimate		
Input Class	A	B	C
A	**57.5%**	30.9%	11.6%
B	34.7%	**53.5%**	11.7%
C	9.4%	9.2%	**81.3%**

WD	Class Estimate		
Input Class	A	B	C
A	**87.7%**	9.0%	3.3%
B	7.5%	**71.5%**	20.9%
C	3.4%	19.4%	**77.1%**

WD – TD	Class Estimate		
Input Class	A	B	C
A	**30.2%**	-21.9%	-8.3%
B	-27.2%	**18.0%**	9.2%
C	-6.0%	10.2%	**-4.2%**

While both techniques perform similarly at $SNR \geq 20$ dB, the WD fingerprints outperform the TD fingerprints at the lower SNRs. As highlighted in the circled region, the WD fingerprints achieve 80% classification accuracy at $SNR \approx 11$ dB. This represents a gain of approximately 3 dB with respect to equivalent TD fingerprinting performance.

Inter-manufacturer classification is demonstrated using one device each from Cisco, Linksys, and AirMagnet (shown in Table 4.8) transmitting an 802.11G signal Figure 4.19 shows average classification performance for TD and WD fingerprinting. While both techniques perform similarly at $SNR \geq 20$ dB, the WD fingerprinting technique outperforms the TD technique at the lower SNRs. As highlighted in the circled region, the WD fingerprints achieve 80% classification accuracy at $SNR \approx 2$ dB. This represents a gain of approximately 2 dB with respect to equivalent TD fingerprinting performance.

Table 4.7: 802.11G Cisco intra-manufacturer permutations.

	Serial Number			
Perm	N4U9	N4UD	N4UW	N4PX
1	×	×	×	
2		×	×	×

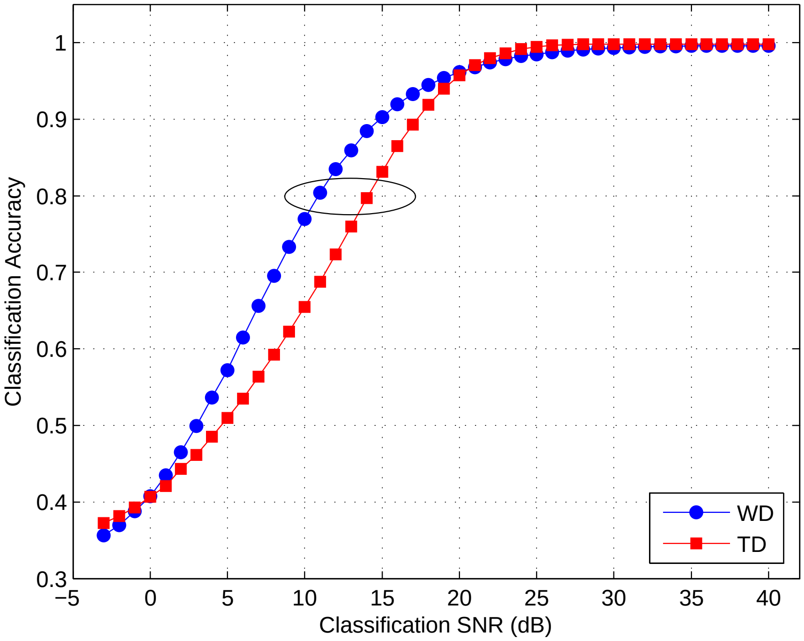


Figure 4.18: Intra-manufacturer MDA/ML classification: Average performance across two permutations of four Cisco devices transmitting 802.11G signals.

4.3.4 Equivalent TD and WD Dimensionality. Based on the number of classification features, the WD fingerprints represent an approximate 5-fold increase in dimensionality over TD fingerprints. This may lead one to conclude that the classification improvement with WD fingerprints is solely attributable to using an increased number of features. It is possible that the performance improvement may be the result of more exploitable features being generated from the DT-ℂWT decomposition. Thus, it is reasonable to ask “Is the noted improvement in Section 4.3.2 attributable to increased feature dimensionality, more exploitable features, or both?” To address this question, results were generated using a subset of 27 selected WD features from the larger 135-feature WD fingerprints. The idea was to compare TD

Table 4.8: 802.11G Inter-manufacturer permutations.

	Cisco	Linksys	AirMagnet
Perm	N4U9	0306	2C01
1	×	×	×

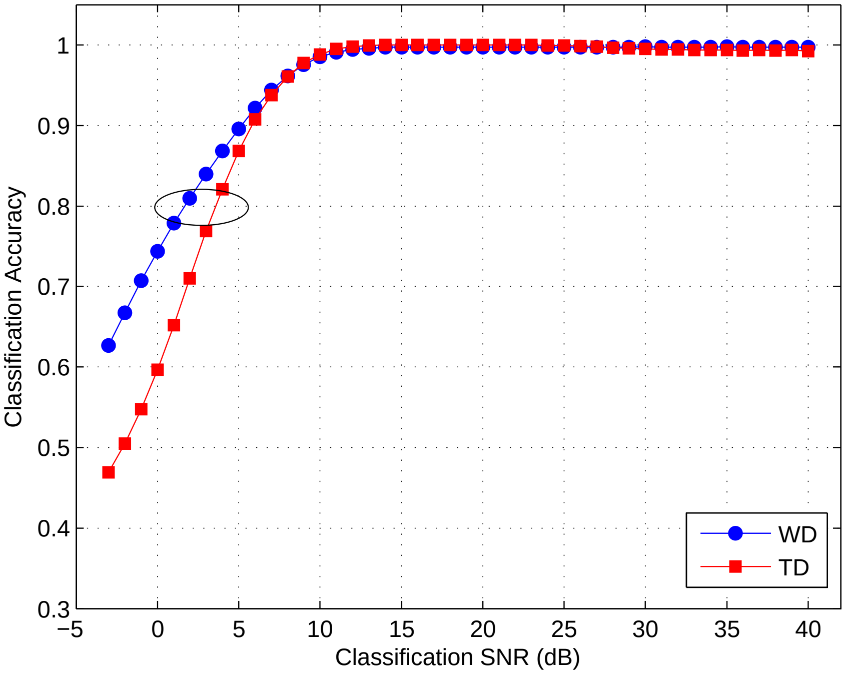


Figure 4.19: Inter-manufacturer MDA/ML classification: Average performance using Cisco, Linksys, and AirMagnet devices transmitting 802.11G signals.

and WD performance using an equivalent number of features. The subset of WD features was selected using the output from a Generalized Relevance Learning Vector Quantization Improved (GRLVQI) classifier [37–39]. The GRLVQI classifier jointly selects features and classifies in order to optimize features for classification. During this process, the algorithm calculates and outputs a relevance rating for each feature considered, indicating feature importance.

Using WD fingerprints from bursts at $SNR = 40$ dB, the GRLVQI classifier was implemented in the Waikato Environment for Knowledge Analysis (WEKA) environment [70] and used to determine relevance factors for all 135 WD features. The features were sorted with respect to their relevance and the 27 most relevant features

Table 4.9: Subset of 27 most relevant WD features from the original 135 features. Relevance ranking (RR) based on GRLVQI classifier output.

RR	**Subregion**	**WD LVL**	**Signal Characteristic**	**Statistic**
1	Entire Preamble	4	Amplitude	Kurtosis
2	Short Symbols	4	Amplitude	Variance
3	Short Symbols	5	Frequency	Variance
4	Entire Preamble	4	Amplitude	Skewness
5	Short Symbols	1	Amplitude	Kurtosis
6	Entire Preamble	5	Frequency	Kurtosis
7	Short Symbols	3	Amplitude	Kurtosis
8	Long Symbols	2	Phase	Kurtosis
9	Entire Preamble	3	Phase	Kurtosis
10	Entire Preamble	3	Phase	Variance
11	Entire Preamble	1	Frequency	Variance
12	Short Symbols	3	Amplitude	Variance
13	Long Symbols	2	Phase	Skewness
14	Entire Preamble	5	Amplitude	Kurtosis
15	Entire Preamble	4	Amplitude	Variance
16	Entire Preamble	3	Amplitude	Kurtosis
17	Entire Preamble	4	Frequency	Kurtosis
18	Short Symbols	1	Frequency	Variance
19	Long Symbols	1	Amplitude	Kurtosis
20	Entire Preamble	5	Phase	Variance
21	Long Symbols	5	Amplitude	Variance
22	Short Symbols	2	Amplitude	Variance
23	Short Symbols	4	Frequency	Kurtosis
24	Entire Preamble	1	Phase	Variance
25	Entire Preamble	3	Phase	Variance
26	Long Symbols	1	Phase	Variance
27	Entire Preamble	1	Phase	Kurtosis

retained for use as alternate WD fingerprints. A rank ordered listing of these features is provided in Table 4.9. The table shows the final relevance ranking (RR), corresponding preamble subregion, WD level (WD LVL), signal characteristic and statistic for each ranked feature. It is interesting to note that a majority of the most relevant features are based on the entire preamble region, followed by the variance statistic and then a tie between the kurtosis statistic and the amplitude characteristic.

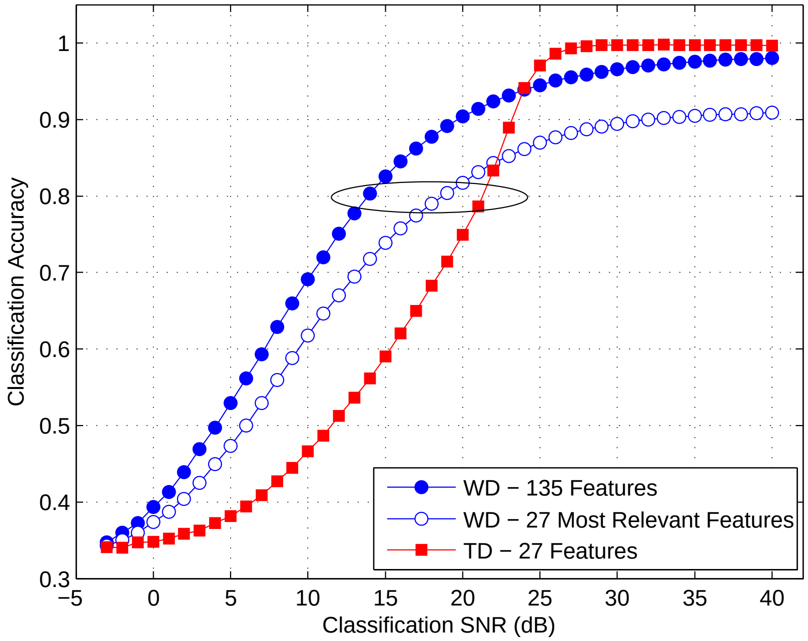


Figure 4.20: Inter-manufacturer MDA/ML classification: Comparison of 27-feature TD and 27-feature WD performance for most stressing case with devices transmitting 802.11A signals.

The 27 most relevant WD features in Table 4.9 were used for WD fingerprinting and performance compared with 27-feature TD fingerprinting performance under the most stressing 802.11A intra-manufacturer discrimination case. Figure 4.20 shows overall classification results. As highlighted in the circled region, the 27-feature WD fingerprints achieve 80% classification accuracy at $SNR \approx 19$ dB. This represents a gain of approximately 2 dB with respect to equivalent 27-feature TD fingerprinting performance. Given equal dimensionality, these results suggest a clear increase in exploitable feature information using the DT-$\mathbb{C}$WT decomposition process.

4.4 Performance Sensitivity Analysis

This section provides results that address classification sensitivity. Overall robustness of the RF fingerprinting and classification process is assessed for three specific cases, including variation in burst location error, variation in MDA/ML training and classification SNRs, and variation in MDA/ML training and classification signal types. Consistent with the overall proof-of-concept research objective, the results here were not generated with a goal toward achieving optimal performance. Rather, they address a few of the most apparent "What if?" type questions that are of interest for operational implementation and provide a basis for the next iteration of research.

4.4.1 Effect of Burst Location Error. The effect of burst location error is demonstrated for TD and WD fingerprinting using random burst location error. This variation addresses the operational situation where equipment used for collecting training data and classification data, equipment which is not necessarily co-located, may be operating in dissimilar environments that are less than ideal. The error considered here is also consistent with what may be induced by laboratory equipment, the fidelity of which can impact collected signal coloration and subsequent burst location accuracy. Two specific random error distributions are considered, including: 1) a four-parameter discrete Beta distribution based on the actual observed error in post-processed collected data, and 2) a uniform distribution having minimum and maximum values that are consistent with the observed error. In both cases, the location error is randomly applied on a burst-by-burst basis to the perfect burst location data. This produces what is referred to here as randomly "jittered" burst location data.

The first series of jittered burst results was generated using statistics from observed location error. The error was determined on a burst-by-burst basis by comparing sample numbers of the -3 dB threshold detected bursts and the corresponding manually detected perfect bursts. This was done during the data collection and post-collection processing detailed in Section 3.2. The resultant histogram for observed error in 9134 collected 802.11A bursts from the four Cisco devices is shown in Figure 4.21. Based on statistics of the observed histogram data (mean, standard deviation, skewness, and kurtosis), a four-parameter discrete Beta distribution generator was created to provide simulated location error similar to what was observed.

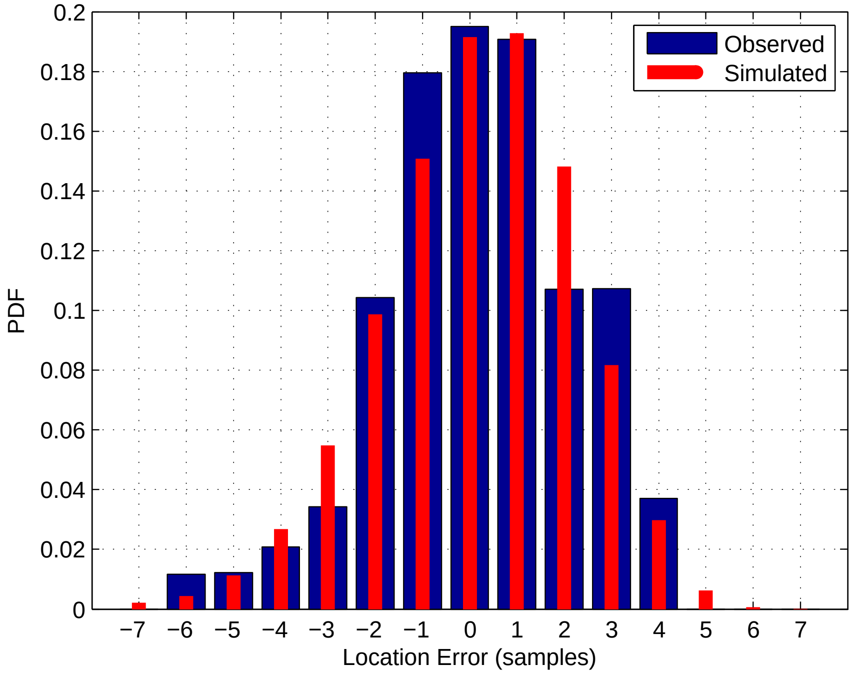


Figure 4.21: Histogram of observed burst location error in 9134 collected 802.11A bursts from four Cisco devices. Simulated error results for the four-parameter discrete Beta distribution are overlayed for comparison.

Simulated error results for the four-parameter discrete Beta distribution are overlayed in Figure 4.21 for comparison.

The random jitter error was applied to perfect burst location data prior to extracting the fingerprints used for both training and classification. This was functionally implemented within Step 2 and described in Section 3.3.4. Intra-manufacturer classification results (for Permutation #2 in Table 4.3) using observed detection error statistics are shown in Figure 4.22 for both WD and TD fingerprinting techniques. For assessing sensitivity to burst location jitter, Figure 4.22 also shows performance for perfect burst location – the WD technique is clearly more robust than the TD technique. Considering the circled region around 80% classification accuracy, two conclusions can be drawn: 1) The WD technique remains superior with 80% classification accuracy achieved at $SNR \approx 9$ dB for both jittered and perfect burst location error. This represents gains of approximately 8 dB (jittered) and 6 dB (perfect) with respect to equivalent TD fingerprinting performance; 2) The WD technique is less sensitive to burst location error. The sensitivities are captured by considering the SNR differences between jittered (SNR_J) and perfect (SNR_P) performance at 80% classification accuracy, where $SNR_\Delta = SNR_J - SNR_P$. These differences are $SNR_\Delta \approx 0$ dB for WD fingerprints and $SNR_\Delta \approx -2.0$ dB for TD fingerprints where the negative sign indicates degradation. The near-zero degradation with WD fingerprints clearly indicates the WD technique is more robust to burst location error.

The second series of jittered burst results was generated using uniformly distributed error of ±6 samples added to the perfect location data. This particular range of values was chosen based on the maximum observed error in Figure 4.21 and presents a more challenging case for classification (higher mean location error relative to the observed statistics case). Results in Figure 4.23 once again demonstrate that WD fingerprints are less sensitive to location error. Comparison of WD results here with those in Figure 4.22 shows minimal additional degradation with uniformly jittered error. However, comparison of TD results here with those in Figure 4.22 shows considerably more degradation with uniformly jittered error.

The increased sensitivity is captured by considering the SNR difference SNR_Δ between jittered and perfect performance at 80% classification accuracy. The difference for WD fingerprints is $SNR_\Delta \approx -1$ dB which is marginally different from the observed jitter case. The difference for TD fingerprints is $SNR_\Delta \approx -12$ dB which is twice the degradation as what occurred in the observed jittered case. These numbers indicate that WD fingerprints are even more robust than previously demonstrated with observed burst location error. This is an important finding for two reasons: 1) it enables subsequent development, demonstration and analysis using a simple uniform error model vice requiring a rigorous statistical model of observed location error, and 2) it paves the way for subsequent trade-off studies and analysis to support burst detector selection (hardware, algorithm, etc.) for system implementation, while at the same time addressing the question "How well does the burst detector need to perform?"

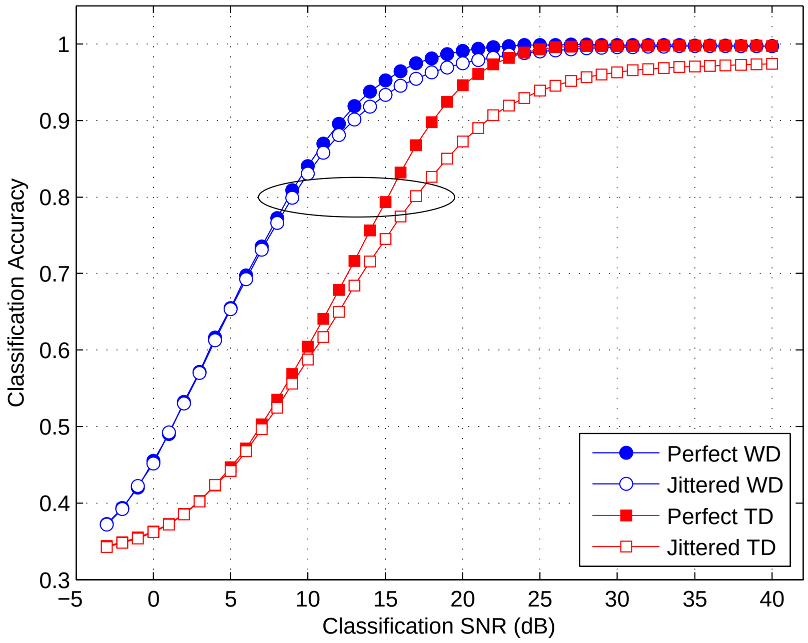


Figure 4.22: Average MDA-ML classification accuracy for 802.11A intra-manufacturer discrimination using *observed* burst location error statistics.

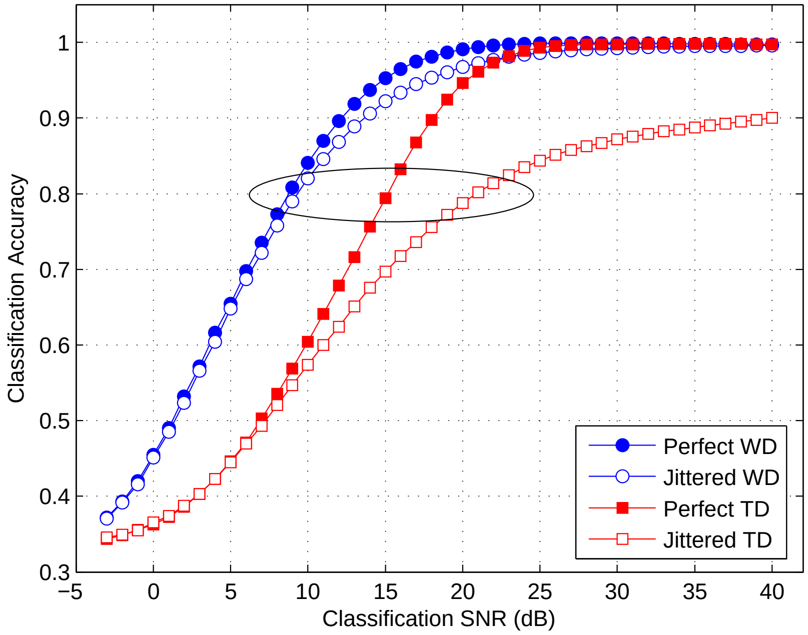


Figure 4.23: Average MDA-ML classification accuracy for 802.11A intra-manufacturer discrimination using *uniform* burst location error statistics.

The final series of results for jittered location error involves the use of dissimilar burst location accuracies for MDA/ML training and classification bursts. The intent is to represent a scenario where higher fidelity data is available for training and lower fidelity data is used for classification. The assumption is that higher fidelity data enables better, more accurate burst location while lower fidelity data yields poorer, less accurate burst location. This situation may occur when bursts for training are collected in a more ideal environment and/or with better equipment, while bursts for classification are collected under poorer environmental conditions and/or with poorer quality equipment. These conditions are simulated here by extracting training fingerprints from bursts with perfect location and extracting classification fingerprints from bursts having randomly jittered location error. In this case, the jittered classification data is generated using the statistical distribution of the observed jitter in Figure 4.22 with a variable mean delay.

Classification accuracy for intra-manufacturer discrimination is shown in Figure 4.24 for a mean delay of 0 to 90 samples (0 to 3.79 μsecs). These results were generated for the most stressing case, Permutation #1 in Table 4.3, for 802.11A signals at $SNR = 40$ dB. Note that performance for 0 mean delay represents an upper bound. As indicated, intra-manufacturer discrimination is highly sensitive to dissimilar burst start location error with performance for both techniques falling below 80% accuracy for all non-zero mean delay values. However, the WD fingerprints remain superior for a majority of the delay values.

Classification accuracy for inter-manufacturer discrimination is shown in Figure 4.25 for a mean delay of 0 to 90 samples (0 to 3.79 μsecs). These results were generated for Permutation #1 in Table 4.5 for 802.11A signals at $SNR = 40$ dB. As indicated, inter-manufacturer discrimination is sensitive to dissimilar burst start location error, just not as sensitive as intra-manufacturer discrimination. In this case, the WD fingerprint performance is relatively stable for mean delays below 14 samples (0.59 μsecs) while the TD fingerprint performance immediately decreases over this same range. Considering the circled region near 80% classification accuracy, the WD

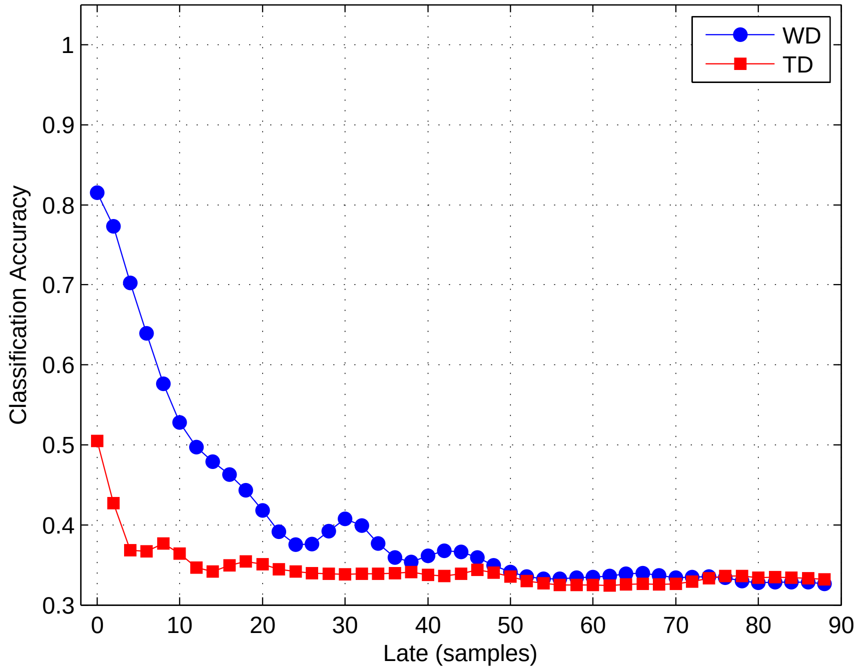


Figure 4.24: Average MDA-ML classification accuracy for 802.11A intra-manufacturer discrimination using dissimilar burst location error.

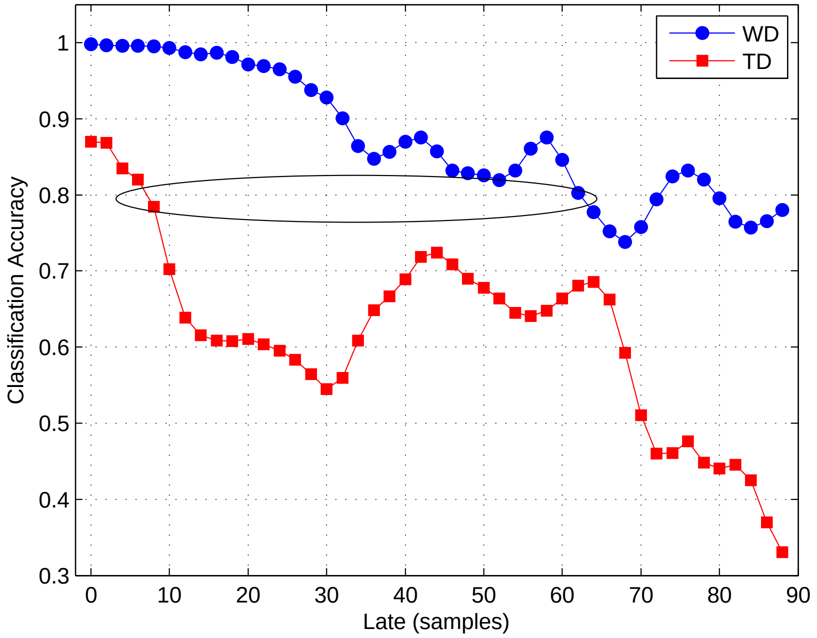


Figure 4.25: Average MDA-ML classification accuracy for 802.11A inter-manufacturer discrimination using dissimilar burst location error.

fingerprints can tolerate up to 55 samples (2.32 μsecs) more of induced mean delay relative to the TD fingerprints.

4.4.2 Effect of Dissimilar Signal SNRs. A comparison is made here between TD and WD fingerprinting performance using dissimilar analysis SNRs for MDA/ML training and classification. Specifically, the training burst SNR is fixed at $SNR =$ 40 dB while the classification burst is varied at $SNR \leq 40$ dB. Fingerprint extraction and classification is conducted using Permutation #1 in Table 4.3.

Results in Figure 4.26 are for intra-manufacturer classification for both WD and TD fingerprints using dissimilar analysis SNRs for training and classification. Relative to performance using identical training and classification SNRs (filled markers), the WD technique experiences a decrease in accuracy for all $SNR < 30$ dB while the TD technique actually performs better at $SNR > 18$ dB and exhibits decreased performance at $SNR \leq 18$ dB. However, comparison of dissimilar SNR results shows that WD performance is more robust for $SNR < 20$ dB. As highlighted in the circled region, WD fingerprints achieve 80% classification accuracy at $SNR \approx 19$ dB. This represents a modest gain of approximately 1 dB with respect to equivalent TD fingerprinting performance. This is approximately 7 dB less gain when compared with performance using identical SNRs for training and classification.

Results in Figure 4.27 are for inter-manufacturer classification for both WD and TD fingerprints using dissimilar analysis SNRs for training and classification. Unlike intra-manufacturer results which exhibited marginal improvement with TD fingerprints over a limited SNR region, there is only degradation in the inter-manufacturer results.

Relative to performance using identical training and classification SNRs (filled markers), the WD technique experiences a decrease in accuracy for all $SNR < 12$ dB while the TD experiences a decrease in accuracy for all $SNR < 20$ dB. Comparison of dissimilar SNR results shows that WD performance is more robust overall and performs better for all SNRs considered. As highlighted in the circled region, WD

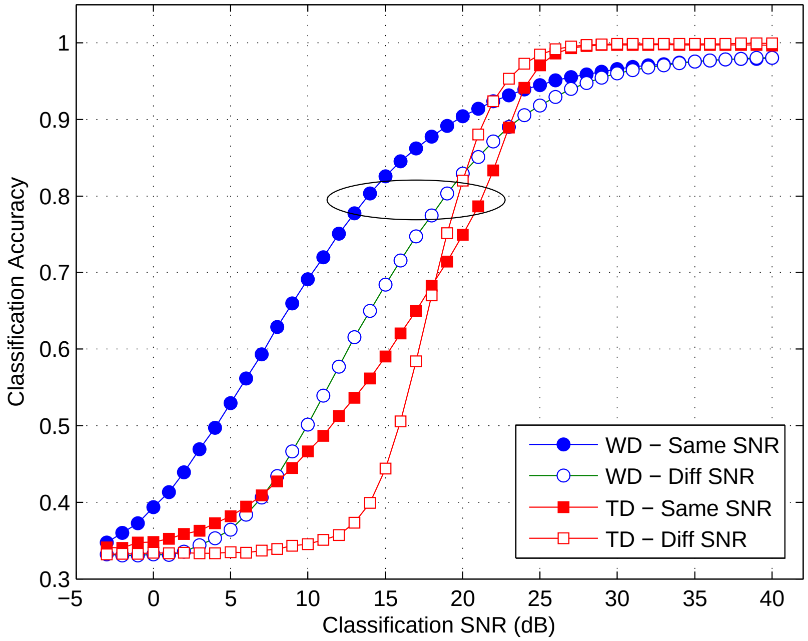


Figure 4.26: Average MDA-ML classification accuracy for 802.11A intra-manufacturer discrimination using dissimilar SNRs.

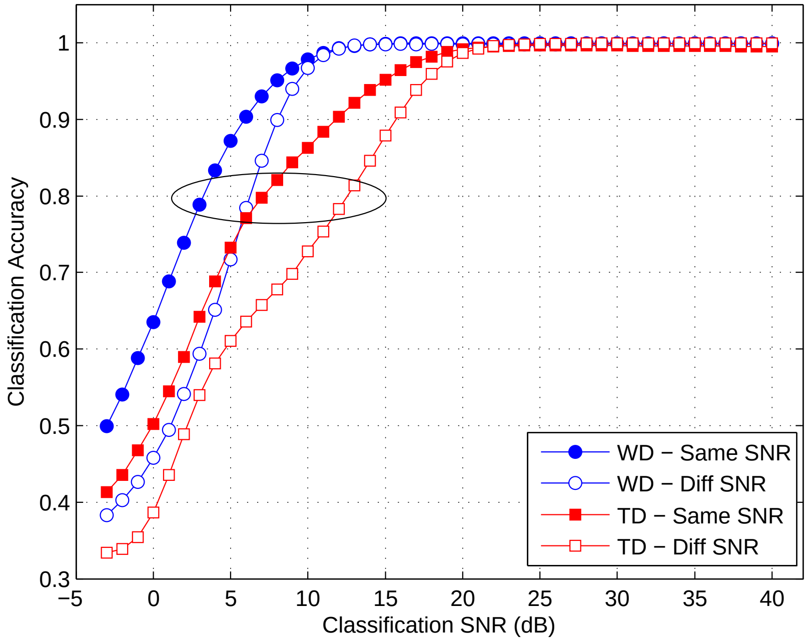


Figure 4.27: Average MDA-ML classification accuracy for 802.11A inter-manufacturer discrimination using dissimilar SNRs.

fingerprints achieve 80% classification accuracy at $SNR \approx 6$ dB. This represents a gain of approximately 6 dB with respect to equivalent TD fingerprinting performance. This is approximately 2 dB more gain when compared with performance using identical SNRs for training and classification.

4.4.3 Effect of Dissimilar Signal Types. The final comparison made between TD and WD fingerprinting performance involves using dissimilar signal types for MDA/ML training and classification fingerprints. Specifically, training fingerprints are generated using 802.11A (802.11G) signals with classification performed using fingerprints generated from 802.11G (802.11A) signals. Recall that the collected 802.11A and 802.11G signals are from the same physical devices operated in two different modes. Thus, the purpose for considering dissimilar signal types is to see if there are inherent signal features that remain unique to a given device as it changes mode. Fingerprint extraction and classification is conducted using Permutation #1 in Table 4.3.

Results in Figure 4.28 are for intra-manufacturer classification for both WD and TD fingerprints using dissimilar signal types for training and classification. For comparison, classification performance is shown for intra-manufacturer discrimination of 802.11A signals using similar signals for training and classification. As indicated by the encircled data points at $SNR = 40$ dB, the intra-manufacturer discrimination capability is very poor (50% or less) using either WD and TD fingerprinting techniques. As consistently demonstrated in previous sections, the WD technique remains more robust and experiences less degradation in accuracy when compared to the TD technique. Given these intra-manufacturer results were so poor, there were no additional results generated for inter-manufacturer discrimination. A detailed investigation into the cause(s) of such poor performance was not within the scope of this research. However, there are two issues that could be considered a good starting point for such an investigation: 1) The same hardware devices were used to produce the 802.11A and 802.11G signals which fundamentally operate at two different carrier

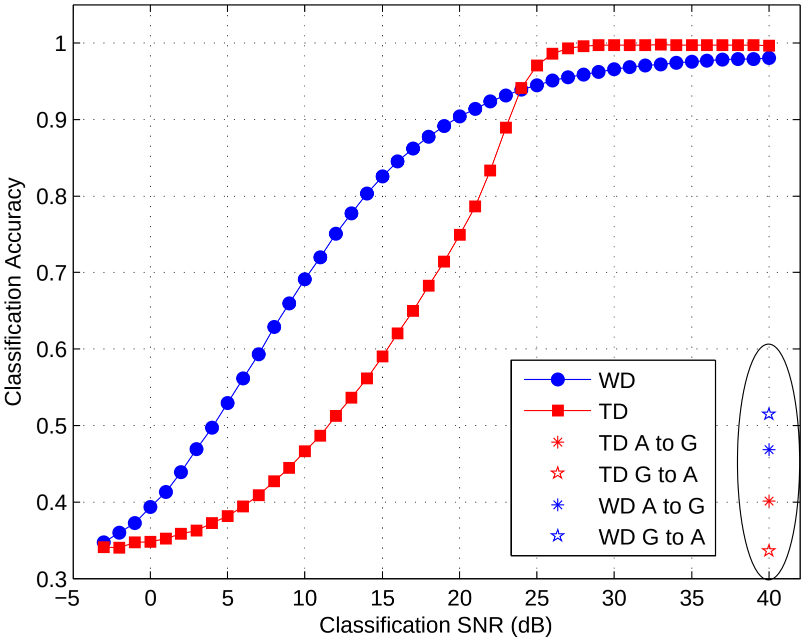


Figure 4.28: Average MDA-ML classification accuracy for intra-manufacturer discrimination using dissimilar signal types (802.11A and 802.11G) for MDA/ML training and classification.

frequencies. Without knowing the exact device details, it can reasonably be assumed that there is at least one component in the RF transmission chain that is either different, or operated differently, between the two modes to place each of the signals at their operating frequencies. Thus, there is perhaps dissimilar coloration that impacts signal features such that they are not the same across the two operating modes; and 2) The same RFSICS was used to collect the two signals. Given the two signals are at different RF carrier frequencies, the internal RFSICS parameters for filtering, downconversion, etc., are necessarily different to ensure collected signal responses reside at baseband. Thus, there is perhaps additional coloration due to RF/IF collection chain differences in the RFSICS that can further impact signal features. Collectively, the RF transmission chain of the devices and the RF/IF collection chain of the RFSICS could be inducing unremovable biases in collected signals.

Summary

This chapter provided modeling, simulation and analysis results that were generated using the processes detailed in Chapter 3. A subset of representative results were presented for Bandwidth Sensitivity, Burst Detection and Location, MDA/ML Classification, and Performance Sensitivity Analysis. Relative to corresponding time-domain (non-wavelet) methods and results, application of the DT-CWT provided improvement for all burst detection and RF fingerprint classification scenarios.

V. Conclusion

This chapter concludes the main document by providing an overall summary of research activities, a summary of key findings, and recommendations for subsequent research. This is followed by an appendix that provides some of the developmental MATLAB® code used to support the research.

5.1 Research Summary

The continued proliferation of affordable Radio Frequency (RF) communication devices has greatly increased wireless user exposure and the need for improved security to protect against spoofing. Historically, research has focused on the detection and mitigation of spoofing using bit-level algorithmic approaches. More recently, there has been a shift toward providing added security within the Physical (PHY) layer of the Open Systems Interconnection (OSI) reference model by exploiting RF features that are 1) inherently unique to a specific device, and 2) are difficult to replicate by an unintended party. This work addresses the extraction and exploitation of RF "fingerprints" to classify emissions and provide hardware specific, serial number identification–Specific Emitter Identification (SEI). The related SEI concepts that formed the foundation for this research are collectively embodied in previous work on RF fingerprinting, electromagnetic signatures, intrapulse modulation, and unintentional modulation [11, 23, 24, 30, 34, 51, 64, 66, 68],

Radar systems have been identified using SEI techniques that exploit inherent signal features that are unique to a given system [9, 40, 60]. The set of exploitable inherent features may contain unintentional modulation contributions that can be influenced by any number of environmental and/or hardware issues, some of which include poor system design (device incompatibility), improper operation (over/under voltage), and physical device limitations (operating temperature range) [30, 34, 68]. Many of the observed unintentional radar modulation effects are similar to what exist in modern wireless communication systems using burst-like waveforms. This begs the question: "Can existing SEI methods be employed with wireless communication signals to achieve radar-like SEI capability?" Answering this provided the motivation for applying a Dual-Tree ℂomplex Wavelet Transform (DT-ℂWT) to improve burst detection and RF fingerprint classification.

Despite the wealth of previous work that forms the basis for this research [23, 24, 42, 54, 55, 63, 64, 66, 67], the task of automatically detecting, identifying and locating RF communication devices remains a challenging problem. The work here addressed four main aspects of this problem, including: 1) the selection and generation of fundamental signal characteristics (amplitude, phase, and/or frequency), 2) the feasibility and repeatability of detecting and locating the start of a burst using selected waveform feature(s) amidst channel noise, 3) the identification and robust extraction of distinguishable fingerprints–features that uniquely characterize the unintentional modulation of a device, and 4) the performance of signal classification under varying channel conditions and Signal-to-Noise Ratio (SNR). As summarized below, various contributions were derived from the research while addressing each of these aspects.

1. **SNR Sensitivity Analysis** [33]: Except for the two most relevant earlier works [54, 55], prior works lacked a detailed sensitivity analysis of burst detection and fingerprint classification performance under varying channel SNR conditions. To address this deficiency, this work analyzed performance of two burst detection techniques, including the Fractal-Bayesian Step Change Detector (Fractal-BSCD) and Traditional Variance Trajectory (Traditional VT). With respect to burst location estimation, both Fractal-BSCD and Traditional VT techniques provided results that were consistent with perfect burst location at higher SNRs ($10 \leq SNR \leq 30$ dB). However, performance for both techniques diverged at lower SNRs ($-3 \leq SNR \leq 10$ dB). With respect to the burst location estimation error impact to classification performance, the Traditional VT technique was consistent with perfect estimation for ($6 \leq SNR \leq 30$ dB)

but under performed for ($-3 \leq SNR < 6$ dB). Traditional VT also provided considerable improvement relative to the Fractal-BSCD at lower $SNRs$ ($-3 \leq SNR \leq 18$ dB), i.e., for a given classification accuracy in the range of 50%–80% the required SNR for Traditional VT is 3–6 dB lower than what is required for Fractal-BSCD.

2. **Burst Detection at Lower SNR** [31]: To improve burst detection and location capability at lower SNRs, the DT-$\mathbb{C}$WT was used to "denoise" signals prior to applying Traditional VT burst detection. Results for this new Denoised VT technique are more effective and provide more robust burst detection and location at lower SNRs ($-3 \leq SNR \leq 10$ dB). Relative to results for perfect burst detection and location, the Denoised VT process achieves nearly 34% of the available performance improvement–when used with Multiple Discriminant Analysis/Maximum Likelihood (MDA/ML) processing, there is little more to be gained in overall classification performance by improving burst detection and location accuracy.

3. **TD Fingerprint Classification**: Given demonstrated improvements in burst detection and location, the research emphasis shifted to improving upon previous Time Domain (TD) RF fingerprinting performance using the newly developed Wavelet Domain (WD) RF fingerprinting technique. To assess relative TD–WD classification performance, it was necessary to replicate previous TD processing. Given that all previous TD work was based on a fixed post-collection bandwidth $BW_{PC} \approx 9$ MHz, an appropriate value based on sound engineering practice versus best or optimal criteria, a sensitivity analysis for varying BW_{PC} was conducted to determine the best choice. For collected 802.11A signals at $SNR = 40$ dB, this analysis indicated that TD performance was very sensitive and exhibited classification variation of nearly 6% for 5 MHz$< BW_{PC} < 9$ MHz, with best case near 100% accuracy realized for $BW_{PC} = 7.7$ MHz and poorest performance realized for $BW_{PC} = 6.3$ MHz. Thus, $BW_{PC} = 7.7$ MHz was used

for all results obtained here which are generally better than previous TD results in [54, 55] based on $BW_{PC} \approx 9$ MHz.

4. **WD Fingerprint Classification** [31–33]: The newly developed WD RF fingerprinting technique uses coefficients from a DT-$\mathbb{C}$WT decomposition to enhance statistical fingerprint features and improve overall device classification performance. Its performance was demonstrated in four stages:

   (a) A BW_{PC} sensitivity analysis was conducted similar to what was done for TD classification. This analysis revealed that WD performance was nearly insensitive to BW_{PC}, with nearly 98% accuracy achieved for all 5 MHz $<$ $BW_{PC} < 9$ MHz. For comparative TD–WD assessment, $BW_{PC} = 7.7$ MHz was used for both techniques (best case).

   (b) Using $BW_{PC} = 7.7$ MHz (best case for both techniques), improved WD classification performance was demonstrated using perfect burst location for both intra-manufacturer (all devices from the same manufacturer) and inter-manufacturer (a mix of devices from different manufacturers) scenarios with both 802.11A and 802.11G signals. TD and WD classification performance was compared under identical scenarios (devices, signal types, SNRs, etc.) using a "gain" metric defined as the reduction in required SNR for the WD technique to achieve the same classification performance as the TD technique. For 80% correct classification performance, the WD technique provided $2-7$ dB gain at 2 dB $< SNR_{WD} <$ 11 dB. The approximate 2% best case TD advantage at higher SNRs rapidly diminishes at lower SNRs that are more consistent with operational environments [31, 33].

   (c) The previous perfect burst location results were based on 27 TD and 135 WD fingerprint features. To address the question, "Is the noted improvement attributable to increased feature dimensionality, more exploitable features, or both?," results were generated using a subset of 27 selected WD features and compared with 27 feature TD results. For an

80% classification level, the WD fingerprints provided a gain of 2 dB at $SNR_{WD} = 19$ dB, suggesting a clear increase in exploitable feature information in the DT-ℂWT coefficients.

(d) Lastly, WD classification performance sensitivity was assessed for variation in burst location error, variation in MDA/ML training and classification SNRs, and variation in MDA/ML training and classification signal types. For all cases considered, the WD technique proved to be more robust and less sensitive when compared to TD technique.

5.2 Recommendations for Future Research

As noted in Section 1.1.2, the choice of demonstrating WD fingerprinting with OFDM-based signals was motivated by two factors, including 1) consistency with previously published TD work [42, 54, 55, 63, 67] and 2) the continued emergence of OFDM-based signals as envisioned for 4G software defined and cognitive radio (SDR/CR) communications [21, 26, 48, 72]. Relative to earlier TD work, the applicability and benefits of DT-ℂWT fingerprint features has been clearly demonstrated and well-received within the technical community [31–33]. However, there remains additional topics of interest that could be investigated. Some of the most evident include:

1. **Optimization of Processes or Parameters**: As used for demonstrating DT-ℂWT applicability to burst detection and RF fingerprinting, there are numerous processes and parameters that impact performance. Given demonstration versus optimization was the goal for this research, the degree to which any given factor, parameter and/or combination thereof impacts performance was not assessed. As developed, implemented and demonstrated, the RF fingerprinting process is well-suited for more rigorous optimization using a Design of Experiments (DOE) methodology with Analysis of Variance (ANOVA). The optimization process could consider any number of conventional techniques, with

two of the most common being Genetic Algorithms (GA) and Response Surface Methodology (RSM).

2. **Demonstration Using Different Signals**: Demonstration results in this research were based on collected 802.11A and 802.11G OFDM-based signals. There are additional OFDM-based signals that are emerging for 4G applications. For example, the Worldwide Interoperability for Microwave Access (WiMax) signal has emerged and is rapidly becoming popular for establishing "last mile" communication connectivity. In this case, the designated WiMax base station serves a similar role as a GSM cellular base station and controls user activity within a defined geographic region. Additional work could be performed to address the use of RF fingerprinting to provide intra-cellular WiMax security.

3. **Demonstration of Cross-Mode Independence**: There is some potential operational benefit if "cross-mode" device discrimination could be reliably accomplished, i.e., achieving serial number SEI based on fingerprint features that are common across multiple operating modes of a given hardware device. For example, there are IEEE 802.11 compliant devices that support multiple modes (signal types) such as an 802.11A/B/G/N device. While less than favorable, the dissimilar signal type results in Section 4.4.3 using 802.11A and 802.11G suggest that the specific features considered here are not robust enough for cross-mode classification. Thus, additional cross-mode work could be performed to determine if there are exploitable underlying RF features for a given device that are independent of operating mode.

4. **Fused Soft-Decision Classification**: All device classification results presented in this work are based on averaging what may be called "hard decision" burst-by-burst classification decisions, i.e., every burst input to the MDA/ML process is associated with a given device class (Class A, Class B, or Class C) independent of how other input bursts are classified. In many applications it is often possible to improve performance by averaging out undesired background noise effects. This can be accomplished in various system processing stages, e.g.,

RF, IF, pre-detection, post-detection, pre-classification, post-classification, etc. Given the communication signals of interest here are burst-like, with hundreds or thousands of burst generated in relatively short time intervals, it is reasonable to assume that device classification may be improved using what may be called "soft-decision" device classification. Additional work could consider using knowledge gained by analyzing a collection of multiple burst-by-burst classification decisions before making a final device classification.

5.3 Sponsor Acknowledgement

This research was sponsored in part by the Sensors Directorate of the Air Force Research Laboratory (AFRL/RY), Wright-Patterson AFB, Ohio, and supports their vision of providing sensor technology to enable complete freedom of air and space operations for warfighters by leading the discovery, development, and integration of affordable sensor technologies [69]. Additional sponsorship was provided by the Tactical SIGINT Technology (TST) Program Office whose mission is to conduct exploratory and advanced research and development in support of requirements and intelligence needs of combatant commanders [56].

Appendix A. MATLAB® Code

The appendix provides the main MATLAB® files used to functionally implement the processes detailed in Chapter 3 and used for obtaining results presented in Chapter 4. As provided below, the code included is for Burst Detection in Section A.1, Preamble Location in Section A.2, Feature Extraction in Section A.3, Device Classification in Section A.4 and DT-ℂWT Transformation in Section A.5.

A.1 Burst Detection

Listing A.1: Code/Detect/PulseDetectV2.m

```
% =============================================================
%          Pulse Detection via Amplitude Thresholding
% =============================================================
%
%   Performs Threshold Amplitude Detection of Pulses and Ouputs
%   a Matrix Containing One Pulse Per Row. Amplitude Detection
%   is Accomplished Using a Simple Leading Edge Detector Opera-
%   tingon a Smoothed Magnitude Response of the Input Signal
%  %
% function [PlsMat,PlsWdth,PlsDb] = PulseDetectV2(Z,MaxPul,...
%              AddSamp,NumSmth,PlsMin,PlsMax,Thresh,NScr,NPlot)
%
%   Created: 4 Nov 2008
%       By: Dr. Michael A. Temple
%   Modified: 8 May 2009
%       By:  Dr. Michael A. Temple
%
% Inputs%
%   Z = Complex Sampled Input Signal (Column or Row Vector).%
%   MaxPul = Desired Maximum # of Pulses to be Detected. Actual
%            Number in Output May be Less Depending on the
%            the Number of Detected Pulses Satisfying PlsMin
%            and PlsMax Criteria%
%   AddSamp = Additional # of Input Samples Included Before
%             & After Threshold Points at Edges of Pulse.%
%   NumSmth = # Samples Smoothed/Averaged Across%
%   PlsMin = Min # Samples in Desired Output Pulse Width
%   PlsMax = Max # Samples in Desired Output Pulse Width%
%   Thresh = Desired Detection Threshold Value in dB
%            Thresh < 0 REQUIRED !%
%   NScr = Output Waitbar Progress/Status to the Screen?
%          1 = Yes    0 = No%
%   NPlot = Produce Output Plots?
%           1 = Yes    0 = No%
% Guide for Selecting Initial Parameter Values%
%   AddSamp: Some Number <= # Samples Between Two Closest Spaced
%            Bursts Divided by 2.
%   NumSmth: 2%-5% of SHORTEST Pulse Duration.  Note that poorer
%            SNR generally requires a larger NumSmth value.%
%
%  Outputs
%   PlsMat = Output Pulse Matrix with One Detected Pulse Per Row.
%            When Variable Width Pulses are Detected, ALL
%            Non-MaxWidth Pulses are Zero-Padded in last Columns.%
%   PlsWdth = Pulse Width (# Samples) Between Leading & Trailing
%             of Detected Pulse Edges:  EstBetween Leading and
%             Trailing Edges of the Smoothed Response.%
%   PlsDb = ACTUAL Relative Power Level (dB) of Output Pulses at
%           Leading Edge Detection Point of SMOOTHED Magnitude.%

function [PlsMat,PlsWdth,PlsDb] = PulseDetectV2(Z,MaxPul,...
         AddSamp,NumSmth,PlsMin,PlsMax,Thresh,NScr,NPlot)
LengthZ=length(Z);
PlsMat=[];
PlsWdth=[];
PlsDb=[];
% Ensure / Make Z a Row Vector
Dim = size(Z);
```

```
if Dim(2)==1 % Column Vector Input ... Change to Row Vector
    Z = Z.'; % Use Non-Conjugate Transpose
end
%  Pad/Extend length of processed 'TmpZ' by 2*PlsMax to help
%  mitigate pulse detection issues the end of input Signal 'Z'
ExtZ=round(2*PlsMax);
TmpZ = [Z ones(1,ExtZ)*min(Z)];
% Note: Matlab's SMOOTH Function ALWAYS returns a Column Vector.
% A Transpose is Used on Smooth Func to Restore a Row Vector
SmthZmag = 20*log10(smooth(abs(TmpZ),NumSmth)');
TmpZmag = SmthZmag;
LenTmpZmag = length(TmpZmag);
MinZ_Db=min(SmthZmag); % Min Value of Input Signal
MaxZ_Db=max(SmthZmag); % Max Value of Input Signal
if NScr==1
    BurstCons = waitbar(0,'Starting Burst Detection Loop');
end
% Begin Main While Loop
% Initialize Pulse Detection While Loop Variables
PulseMatrix = [];
PulseVec = [];
MaxWidth = 0;
NumDet = 0;
PlsWdth=0;
NPlsDet=0; % Intialize Pulses Detection Counter
WhileMax=2*MaxPul; % Set Max # of "While Loop" Iterations
WhileCnt=0;
while NPlsDet < MaxPul % Maximum # of Pulses to be Detected
    WhileCnt=WhileCnt+1;
    if WhileCnt >= WhileMax
        break
    end
    % Find Smoothed Peak Response
    [MaxVal,MaxLoc] = max((TmpZmag));
    LowDex = MaxLoc;
```

```
for k = 1:2*PlsMax  % Search Left to Leading Edge
    if(LowDex-1) < 1 % First Sample Reached
        break        % Stop Search !
    else             % Continue Searching
        if TmpZmag(LowDex-1) > MaxVal + Thresh;
            LowDex=LowDex-1; % Index # at Threshold
        else
            break;
        end
    end
end
PlsLow = LowDex-AddSamp;
if PlsLow < 1
    PlsLow =1;
end
HghDex=MaxLoc;
for k = 1:2*PlsMax % Search Right to Trailing Edge
    if(HghDex+1) > LenTmpZmag % Last Sample Reached
        break                 % Stop Search !
    else                      % Continue Searching
        if TmpZmag(HghDex+1) > MaxVal + Thresh;
            HghDex=HghDex+1; % Index # at Threshold
        else
            break
        end
    end
end
PlsHgh = HghDex+AddSamp;
if PlsHgh > LengthZ
    PlsHgh = LengthZ;
end
TmpWdth=HghDex-LowDex; % Width Between Pulse Edges
% Check:  PlsMin < Temp Width < PlsMax
%   Not Satisfied -> Do NOT Include Current Pulse
%   Satisfied -> INCLUDE Current Detected Pulse
```

```
    if TmpWdth > PlsMin % Include Current Pulse
        % Decrement While Loop Counter For EVERY Detected Pulse
        WhileCnt=WhileCnt-1;
        if TmpWdth < PlsMax
            NumDet=NumDet+1;
            TmpDb(NumDet)=SmthZmag(LowDex)-MaxVal;
            PlsWdth(NumDet)=TmpWdth;
            PulseLoc(NumDet)=PlsLow; % Store Pls Location Index
            PlsDet=TmpZ(PlsLow:PlsHgh);
            PlsDur(NumDet)=length(PlsDet); % Store Pls Duration
            PulseVec = [PulseVec,PlsDet]; % Unsorted Pulse Vector
            NPlsDet=NPlsDet+1; % Update Detected Pulse Counter
            if NScr==1 % Update Status to Screen ?
                waitbar(NPlsDet/MaxPul,BurstCons,
                ['Burst Number ', num2str(NPlsDet),' of ',...
                   num2str(MaxPul), ' Detected.'])
            end
        end
    end
    TmpZmag(PlsLow:PlsHgh)=MinZ_Db; % Remove Current Det Pls
end % Detection While Loop ... Detect Next Pulse%
% End Main While Loop%
if NScr==1 % Update Waitbar Screen Status?
    close (BurstCons)
    display([' '])
    display(['      A Total of ',num2str(NumDet),...
        ' Pulses Satisfied Pulse Width Constraints.'])
    display([' '])
end
% Put Detected Pulses in Matrix Form with ONE pulse per row.
if NumDet > 0
    TmpVec = PulseVec; % Unsorted Pulse Vector
    MaxWidth = max(PlsDur);
    PulseMatrix=zeros(NumDet,MaxWidth);
    for k=1:NumDet
```

```
            PulseMatrix(k,1:PlsDur(k))=TmpVec(1:PlsDur(k));
            TmpVec(1:PlsDur(k))=[];
        end
        % Reorder Pulses to Original Collection Time Order
        PlsMat=[];
        [SortVal,SortLoc]=sort(PulseLoc);
        PlsMat=zeros(NumDet,MaxWidth);
        TmpWdth=PlsWdth;
        for k=1:NumDet
            PlsMat(k,:)=PulseMatrix(SortLoc(k),:);
            PlsWdth(k)=TmpWdth(SortLoc(k)); % Reorder Pulse Widths
            PlsDb(k)=TmpDb(SortLoc(k)); % Reorder Det Point Db
        end
    else
        if NScr==1 % Update Waitbar Screen Status?
            display([’ ’])
            display([’No Detected Pulses Satisfy Pulse Width ...
               Constraint’])
        end
    end
    %Begin Plotting Code
    if NPlot==1 % Satisfied -> Produce Plots
        figure (1) % Magnitude of Input Signal Plot
        subplot(3,1,1)
        plot(abs(Z))
        grid
        axis tight
        title(’Magnitude of Input Signal Z’)
        ylabel(’|Z|’)
        xlabel(’Sample Number’)
        %
        subplot(3,1,2)
        plot(SmthZmag)
        grid
        axis tight
```

```
title(['UN-NORMALIZED Smoothed |Z| for NumSmth = ',...
       num2str(NumSmth)])
ylabel('|Z| (dB)')
xlabel('Sample Number')
%
subplot(3,1,3)
plot(SmthZmag - MaxZ_Db)
grid
axis tight
title(['NORMALIZED Magnitude of Z for NumSmth = ',...
        num2str(NumSmth)])
ylabel('|Z| (dB)')
xlabel('Sample Number')
if NumDet > 0 % Only Generate Plots If Pulses Are Detected
    % Create Sorted 'VECTOR' of Final Pulses for Plotting
    SortVec=reshape(PlsMat.',1,NumDet*MaxWidth);
    figure(2) % Magnitude of Input Signal Plot
    subplot(3,1,1)
    plot(abs(Z))
    axis tight
    grid
    title('Magnitude of Input Signal Z')
    xlabel('Sample Number')
    ylabel('|Z|')
    %
    subplot(3,1,2)
    plot(abs(PulseVec))
    axis tight
    grid
    title(['ABS [Unsorted Pulses]: ',num2str(NumDet),...
           ' Pulses Detected'])
    xlabel('Sample Number')
    ylabel('|Z|')
    %
    subplot(3,1,3)
```

```
plot(abs(SortVec))
axis tight
grid
title(['ABS [Sorted Pulses]: ',num2str(NumDet),...
       ' Pulses Detected'])
xlabel('Sample Number')
ylabel('|Z|')
%
figure(3) % Relative Pulse Amplitude Plot
subplot(3,1,1)
plot(PlsDb,'*')
if Thresh < 0
    title(['Rel Pulse Amp at AddSamp + 1 = ',...
    num2str(AddSamp+1),' for Input Threshold = ',...
    num2str(Thresh),' dB'])
else
    title('Rel Pulse Amp at Threshold Pt: No Input ...
       Threshold')
end
axis tight
set(gca,'XLim',[.98 1.01*NumDet])
xlabel('Pulse Number')
ylabel('dB')
grid
%
subplot(3,1,2)
hold
for k=1:NumDet
    plot(abs(PlsMat(k,:)));
end
grid
title(['Overlay of ABS [PlsMat]: ',...
    num2str(NumDet),' Pulses, ', ...
    'NumSmth = ',num2str(NumSmth)])
xlabel('Sample Number')
```

```
        ylabel('ABS')
        axis tight
        %
        subplot(3,1,3)
        plot(mean(abs(PlsMat)));
        grid
        axis tight
        title(['Column-Wise Mean of ABS [PlsMat]: ',...
                num2str(NumDet),' Pulses'])
        xlabel('Sample Number')
        ylabel('Mean')
    end
end
% End Pulse Detect Function
```

A.2 Preamble Location

Listing A.2: Code/Locate/LocatePreamble.m

```
function [Index,Preamble] = LocatePreamble(Signal,LocMeth,VtThresh...
   ,Threshold,SNRdb,StateI,StateQ,IndFlag,Index,F_BW)
dir = 'F:\Chamber\';
% Hard code to speed up execution
FilterFreqsHardCode;
[B,S] = size(Signal);
B_range = 1:B;
P = 380;
Preamble = zeros(B,P);
if IndFlag ~= 1
    Index = zeros(B,1);
end
Buffer = 500;
C = S+Buffer;
trans_truth = Buffer+1;
RandDataStart = trans_truth + 475; % After Symbol region
wind = 20;
s = 2;
```

```
win_start = 1 : s : s*floor(( 2^11 -wind)/s);
W = length(win_start);
f_RandDataStart = find(win_start<RandDataStart, 1, 'last' );
[Faf, Fsf] = FSfarras;
[af, sf] = dualfilt1;
% Set Chebyshev type I filter parameters:
N = 6;      % Order of filter -- change to 6
Rp = 0.01;  % Passband ripple in dB
Tsig = XDelta*S;
Nsig = round(Tsig/XDelta);
% Set up more filter parameters
Fs2 = 1/(2*XDelta);
Wn = F_BW/Fs2;
[num den] = cheby1(N,Rp,Wn);    % Filter coefficients
for b = B_range % burst
    Data = Signal(b,:);
    Data = [zeros(1,1000),Data,zeros(1,1000)]; % Zero pad data ...
       prior to filtering
    Data = filtfilt(num,den,Data); % Filter data
    Data = Data(1001:1000+Nsig); % Un-Zero pad data after ...
       filtering
    Data = Data - mean(Data);
    Spow = sum(abs(Data).^2)/S;
    Data = Data./(sqrt(Spow/2)*(1+j));
    Spow = sum(abs(Data).^2)/S;
    nz_rz_I = randn(1,C);
    nz_rz_Q = randn(1,C);
    nz = (nz_rz_I+j*nz_rz_Q);
    % Filter Noise
    Nlen = length(nz);
    Noise = [zeros(1,1000),nz,zeros(1,1000)]; % Zero pad noise ...
       prior to filtering
    Noise = filtfilt(num,den,Noise); % Filter noise
    Noise = Noise(1001:1000+Nlen); % Un-Zero noise data after ...
       filtering
```

```
Noise = Noise-mean(Noise);
NP =  sum(abs(Noise).^2)/C;
Npow = (Spow)/(10^(SNRdb/10));
noise = sqrt(Npow/NP) * Noise;
sig = [zeros(1,C-S),Data] + noise;
sig = sig(1: 2^11 );
if IndFlag ~= 1
    %Feature Extraction
    windowed = zeros(W,wind+1);
    if strcmp(LocMeth,’DenVt’) | strcmp(LocMeth,’Vt’)
        if strcmp(LocMeth,’DenVt’)
            max_level = 4;
            sig = sig-mean(sig);
            y = dualtree(sig,max_level,Faf,af);
            for p = 1:max_level
                aa = abs(y{p}{1});
                bb = abs(y{p}{2});
                cc = sqrt((aa).^2 + (bb).^2);
                Y{p}=y{p};%
                % Zeroes coeffs that don’t represent enough of...
                    value
                [m2,n2] = find(abs(cc)<Threshold(p));
                Y{p}{1}(n2) = 0;
                Y{p}{2}(n2) = 0;
            end
            Y{p+1}{1} = y{p+1}{1};
            Y{p+1}{2} = y{p+1}{2};
            Sig = (idualtree(Y,max_level,Fsf,sf));
        elseif strcmp(LocMeth,’Vt’)
            sig = sig-mean(sig);
            Sig = sig;
        end
        for w = 1:W
            windowed(w,1:wind+1) = Sig(win_start(w):win_start(...
               w)+wind); % windowing the total Signal
```

```
                x = (Sig(win_start(w):win_start(w)+wind-1));
                x = x-mean(x);
                V(w) = var(abs(x));
            end
            VT = abs(V(1:end-1)-V(2:end));
            [f_index_vt]=DetThresh(VT(1:f_RandDataStart),V(1:...
               f_RandDataStart),VtThresh);
            Index(b) = win_start(f_index_vt) + wind - s;
        elseif strcmp(LocMeth,'Fractal')
            sig = sig-mean(sig);
            Sig = sig;
            for w = 1:W
                windowed(w,1:wind+1) = Sig(win_start(w):win_start(...
                   w)+wind); % windowing the total Signal
            end
            k = 1:wind/2; % repeat k from 1 to kmax
            fractal_mag = CalcFractals(abs(windowed),k)';
            % Detect Transient
            [f_index_mag_frac_pdf,prob_mag_frac]=Bscd(fractal_mag...
               ,3);
            Index(b) = win_start(f_index_mag_frac_pdf) + wind/2+s;
        elseif strcmp(LocMeth,'Perf')
            Index(b) = trans_truth;
        elseif strcmp(LocMeth,'PerfJitter')
            Jitter = 6;
            x = round(Jitter + (-Jitter-Jitter) * rand(1));
            Index(b) = trans_truth + x;
        end
    end
    Preamble(b,:) = sig(Index(b):Index(b)+P-1);
end
```

Listing A.3: Code/Locate/FilterFreqsHardCode.m

```
FreqValidMax = 5.189169766750000e+009;
FreqValidMin = 5.170615079250000e+009;
```

```
XDelta = 4.210526315789474e-008;
Fs = 23750000;
```

Listing A.4: Code/Locate/CalcFractals.m

```
function [d] = CalcFractals(windowed,k)
%Calculates the fractal dimension, d
index = 1;
[M,N] = size(windowed);
L = zeros(length(k),length(k),M); % initialize for sum over i
for a = 1:length(k)  % k new time series
    m = 1:k(a);  % m ranges from 1 to k
    for b = 1:length(m)
        if (m(b)+k(a))<= (N)
            L(b,a,:) = sum(abs(windowed(:,m(b)+k(a):k(a):end)-...
               windowed(:,m(b):k(a):end-k(a))),2);
            L(b,a,:) = (L(b,a,:)*(N-1)/(floor((N-m(b))/k(a))* k(a)...
               ))/ k(a);
        end
        if isnan(L(b,a,:))
            temp = 0;
        end
    end
end
L = sum(L,1); % average over m
k = repmat(k,[1,1,M]); % repmat k to polyfit
L=squeeze(L);
k=squeeze(k);
for i = 1:M
    p = polyfit(log(k(:,i)),log(L(:,i)),1); %least square line fit...
        for log-log
    d(i) = -p(1);
end
```

Listing A.5: Code/Locate/Bscd.m

```
function [index,prob_density]=Bscd(fractals,w)
```

```
N = length(fractals);
prob_density = zeros(1,N);
m = ceil(w/2);
N = w;
for a = 1:N-w
    % Piecewise fractals for computer precision sake
    d = fractals(a:a+w);
    p =   1/( sqrt(m*(N-m)))/((sum(d.^2)-(1/m)*sum(d(1:m))^2-(1/N-...
       m)*sum(d(m+1:N))^2)^((N-2)/2));
    [prob_density(a+m)] = p;
end
[nothing, index] = max(prob_density);
```

Listing A.6:   Code/Locate/DetThresh.m

```
function [index]=DetThresh(trajectory,support,threshold)
N=length(trajectory);
w_index = 0;
W=N;
w_index = 0;
n=4.5;
m=200;
trigger = mean(trajectory(1:m))+n*std(trajectory(1:m));
ensure = max(support(1:m));
trigger = threshold(1);
ensure = threshold(2);
for i = 1+5:W
    detect = trajectory(i-5);
    verify = mean(support(i-4:i));
    if detect > trigger && verify > ensure
        w_index=i-4;
        break
    end
end
if w_index == 0
    w_index = W;
```

```
end
index = w_index;
```

A.3 Feature Extraction

Listing A.7: Code/Extract/ExtractFeatures.m

```
function Features = ExtractFeatures(Signal,FtrMeth)
[Faf, Fsf] = FSfarras;
[af, sf] = dualfilt1;
[B,S] = size(Signal);
B_range = 1:B;
switch FtrMeth
    case 'Td'
        Features = zeros(B,3,9);
    case {'Wd'}
        Features = zeros(B,3,5,9);
end
Region = {'pre1','pre2','all'};
trans_length.pre1 = 190;
trans_length.pre2 = 190;
trans_length.all = 380;
for b = B_range % burst
    sig = Signal(b,:);
    for x = 1:length(Region)
        if x == 2 %pre2 - must bypass all of pre1
            index_start = 1+trans_length.pre1;
        else
            index_start = 1;
        end
        index_end = index_start+trans_length.(Region{x}) - 1;
        Sig = sig(index_start:index_end);
        Sig = Sig - mean(Sig);
        switch FtrMeth
            case 'Td'
                Features(b,x,:) = InstFeatures(Sig);
            case {'Wd'}
```

```
                max_level = 4;
                zp = 2.^ceil(log2(length(Sig))) - length(Sig);
                switch FtrMeth
                    case 'Wd'
                        SIG = real(Sig);
                end
                Coeffs = dualtree([SIG, zeros(1,zp)],max_level,Faf...
                   ,af);
                for p = 1:max_level+1
                    if p == max_level+1
                        IndTemp = ceil(length(SIG)/(2^max_level));
                    else
                        IndTemp = ceil(length(SIG)/(2^p));
                    end
                    switch FtrMeth
                        case {'Wd','WdC'}
                            aa = Coeffs{p}{1}(1:IndTemp);
                            bb = Coeffs{p}{2}(1:IndTemp);
                            Features(b,x,p,:) = InstFeatures(aa + ...
                               j*bb);
                    end
                end
        end
    end
end
```

Listing A.8: Code/Extract/InstFeatures.m

```
function [Inst_Features]=InstFeatures(signal)
i = [1:length(signal)]';
Fs = 23.75E6;
Tsamp = 1/Fs;
I          = real(signal);
Q          = imag(signal);
Unwrap_Phase = unwrap(atan2(Q,I)');
Inst_Freq  = gradient(Unwrap_Phase,Tsamp)/(2*pi)';
```

```
Inst_Amp    = abs(signal)';
mu_f = mean(Inst_Freq);
Inst_Phase = Unwrap_Phase - 2*pi*i*mu_f/Fs;
Inst_Freq = Inst_Freq - mu_f;
Inst_Phase = Inst_Phase - mean(Inst_Phase);
Inst_Amp = Inst_Amp - mean(Inst_Amp);
% Calculate variance of sub-segment
Amp_var = var(Inst_Amp);
Pha_var = var(Inst_Phase);
Fre_var = var(Inst_Freq);
% Calculate skewness of sub-segment
Amp_skew = skewness(Inst_Amp);
Pha_skew = skewness(Inst_Phase);
Fre_skew = skewness(Inst_Freq);
% Calculate kurtosis of sub-segment
Amp_kurtosis = kurtosis(Inst_Amp);
Pha_kurtosis = kurtosis(Inst_Phase);
Fre_kurtosis = kurtosis(Inst_Freq);
Inst_Features = [Amp_var, Pha_var, Fre_var, Amp_skew, Pha_skew, ...
   Fre_skew, Amp_kurtosis, Pha_kurtosis, Fre_kurtosis];
```

A.4 Device Classification

Listing A.9: Code/Classify/ClassifyDevices.m

```
function ClassAcc = ClassifyDevices(Dvc1,Dvc2,Dvc3)
[B,dim] = size(Dvc1);
k_fold = 5;
if k_fold == 1
    n_class = 1:B;
    n_train = 1:B;
else
    n_class = 1:ceil(1/k_fold*B);
    n_train = n_class(end)+1:B;
end
N_class = length(n_class);
% Initialize Confusion Matrix Variables within the SNR loop
```

```
AA = 0;
AB = AA;
AC = AA;
Aerr = AA;
BB = AA;
BA = AA;
BC = AA;
Berr = AA;
CC = AA;
CA = AA;
CB = AA;
Cerr = AA;

class1data = reshape(Dvc1,[B,dim]);
class2data = reshape(Dvc2,[B,dim]);
class3data = reshape(Dvc3,[B,dim]);
% % Relevant Features
% f27=[102,10,43,57,91,135,97,110,114,24,33,7,65,105,12,99,132,...
%31,92,30,14,4,130,18,39,17,108];
% class1data = class1data(:,f27);
% class2data = class2data(:,f27);
% class3data = class3data(:,f27);
class1train = class1data;
class2train = class2data;
class3train = class3data;
% Scramble data
scramble = randperm(B);
class1data = class1data(scramble,:);
class2data = class2data(scramble,:);
class3data = class3data(scramble,:);
class1train = class1train(scramble,:);
class2train = class2train(scramble,:);
class3train = class3train(scramble,:);
ave_k = zeros(1,k_fold);
for k = 1:k_fold
```

```
i = n_class;
class1data_k = class1data(i,:);
class2data_k = class2data(i,:);
class3data_k = class3data(i,:);
i = n_train;
class1train_k = class1train(i,:);
class2train_k = class2train(i,:);
class3train_k = class3train(i,:);
class1data = circshift(class1data,[N_class,0]);
class2data = circshift(class2data,[N_class,0]);
class3data = circshift(class3data,[N_class,0]);
class1train = circshift(class1train,[N_class,0]);
class2train = circshift(class2train,[N_class,0]);
class3train = circshift(class3train,[N_class,0]);
% Training the MDA Projection Matrix and ML paramters
[X1,X2,x1,x2,R1,R2,R3,data_mean,data_std,W,Fishclass1,...
   Fishclass2,Fishclass3]=Train_class2(class1train_k,...
   class2train_k,class3train_k,class1data_k,class2data_k,...
   class3data_k);
% Classifying
[Tot_err,N_tot,A_in_A,A_in_B,A_in_C,A_err,B_in_B,B_in_A,B_in_C...
   ,B_err,C_in_C,C_in_A,C_in_B,C_err, Fishclass1, Fishclass2, ...
   Fishclass3]...
    =Classify_class(class1data_k,class2data_k,class3data_k,X1,...
       X2,x1,x2,R1,R2,R3,data_mean,data_std,W);
ave_k(k) = (N_tot-Tot_err)./N_tot      ;
%     % Confusion Matrix
%     AA = AA+A_in_A;
%     AB = AB+A_in_B;
%     AC = AC+A_in_C;
%     Aerr = Aerr+A_err;
%     BB = BB+B_in_B;
%     BA = BA+B_in_A;
%     BC = BC+B_in_C;
%     Berr = Berr+B_err;
```

```
    %      CC = CC+C_in_C;
    %      CA = CA+C_in_A;
    %      CB = CB+C_in_B;
    %      Cerr = Cerr+C_err;
end
ClassAcc = ave_k;
```

Listing A.10: Code/Classify/Train.m

```
function [X1 X2 x1 x2 R1 R2 R3 data_mean data_std, W, Fishclass1, ...
    Fishclass2, Fishclass3]=Train_class2(class1data,class2data,...
    class3data,grid1,grid2,grid3)
Limit      = 3;
Res        = .01;
ResPts = 400;
Dis        = 2;
N_rec      = length(class1data(:,1)) ;
N_tot      = N_rec*3;
N_recg      = length(grid1(:,1)) ;
N_totg      = N_recg*3;
%%%%%%%%%%
% PART I %
%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Globally normalize emission records
training_data      = [class1data;class2data;class3data];
data_mean          = ones(N_tot,1)*mean(training_data);
data_std           = ones(N_tot,1)*std(training_data,1);
training_data_norm = (training_data-data_mean)./data_std;
class1data         = training_data_norm(N_rec*0+1:N_rec*1,:);
class2data         = training_data_norm(N_rec*1+1:N_rec*2,:);
class3data         = training_data_norm(N_rec*2+1:N_rec*3,:);
%%%%%%%%%%%
% PART II %
%%%%%%%%%%%
```

```
% Compute the within class covariance matrixes S_k, where k = ...
   1,2,3
S1 = cov(class1data);
S2 = cov(class2data);
S3 = cov(class3data);
% Compute the SW matrix by summing the S_k matrixes
SW = S1 + S2 + S3;
% Compute the within class and total means
m1 = mean(class1data)';
m2 = mean(class2data)';
m3 = mean(class3data)';
mtot = (1/N_tot)*(N_rec*m1 + N_rec*m2 + N_rec*m3);
% Compute the SB matrix
SB = N_rec*(m1 - mtot)*((m1 - mtot)') +...
    N_rec*(m2 - mtot)*((m2 - mtot)') +...
    N_rec*(m3 - mtot)*((m3 - mtot)');
% Solve for x
x = SW\SB;
% Find Fisher plane matrix W (eig vectors corresponding to the two...
    largest eig values)
[V,D]        = eig(x);
lamda        = sum(D);
max1         = find(lamda == max(lamda));
lamda(max1) = NaN;
max2         = find(lamda == max(lamda));
W            = [V(:,max1)';V(:,max2)'];
% Normalize each emission record using the parameters calculated ...
   during training
training_datag      = [grid1;grid2;grid3];
data_meang          = ones(N_totg,1)*data_mean(1,:);
data_stdg           = ones(N_totg,1)*data_std(1,:);
training_data_normg = (training_datag-data_meang)./data_stdg;
class1datag         = training_data_normg(N_recg*0+1:N_recg*1,:);
class2datag         = training_data_normg(N_recg*1+1:N_recg*2,:);
class3datag         = training_data_normg(N_recg*2+1:N_recg*3,:);
```

```
Fishclass1g = W * class1datag';
Fishclass2g = W * class2datag';
Fishclass3g = W * class3datag';
%%%%%%%%%%%
% PART II %
%%%%%%%%%%%
% Project each class using the Fisher plane calculated during ...
    training
Fishclass1 = W * class1data';
Fishclass2 = W * class2data';
Fishclass3 = W * class3data';
LoLimit1 = min([Fishclass1g(1,:),Fishclass2g(1,:),Fishclass3g(1,:)...
    ,Fishclass1(1,:),Fishclass2(1,:),Fishclass3(1,:)]);
LoLimit2 = min([Fishclass1g(2,:),Fishclass2g(2,:),Fishclass3g(2,:)...
    ,Fishclass1(2,:),Fishclass2(2,:),Fishclass3(2,:)]);
HiLimit1 = max([Fishclass1g(1,:),Fishclass2g(1,:),Fishclass3g(1,:)...
    ,Fishclass1(1,:),Fishclass2(1,:),Fishclass3(1,:)]);
HiLimit2 = max([Fishclass1g(2,:),Fishclass2g(2,:),Fishclass3g(2,:)...
    ,Fishclass1(2,:),Fishclass2(2,:),Fishclass3(2,:)]);
ResPts = 400;
x1       = linspace(LoLimit1,HiLimit1,ResPts);
x2       = linspace(LoLimit2,HiLimit2,ResPts);
%%%%%%%%%%%%
% PART III %
%%%%%%%%%%%%
% Find the mean vector for each class
mean1 = mean(Fishclass1');
mean2 = mean(Fishclass2');
mean3 = mean(Fishclass3');
% Find the covariance matrix for each class
K1 = cov(Fishclass1');
K2 = cov(Fishclass2');
K3 = cov(Fishclass3');
% Find the inverted covariance matrix for each class
Q1 = inv(K1);
```

```
Q2 = inv(K2);
Q3 = inv(K3);
[X1 X2] = meshgrid(x1,x2);
K1det = (1/(2*pi*sqrt(det(K1))));
K2det = (1/(2*pi*sqrt(det(K2))));
K3det = (1/(2*pi*sqrt(det(K3))));
px1 = zeros(length(x2),length(x1));
px2 = px1;
px3 = px1;
for i = 1:length(x1)
    for k = 1:length(x2)
                px1(k,i) = [(x1(i)-mean1(1)),(x2(k)-mean1(2))]*Q1...
                   *[(x1(i)-mean1(1));(x2(k)-mean1(2))];
                px2(k,i) = [(x1(i)-mean2(1)),(x2(k)-mean2(2))]*Q2...
                   *[(x1(i)-mean2(1));(x2(k)-mean2(2))];
                px3(k,i) = [(x1(i)-mean3(1)),(x2(k)-mean3(2))]*Q3...
                   *[(x1(i)-mean3(1));(x2(k)-mean3(2))];
    end
end
px1 = K1det * exp(-.5*(px1));
px2 = K2det * exp(-.5*(px2));
px3 = K3det * exp(-.5*(px3));
% Initialize the regions
R1 = zeros(size(px1));
R2 = R1;
R3 = R1;
% Define the Bayesian decision regions
R1(px1>=px2  & px1>px3) = 1;
R2(px2>px1 & px2>=px3) = 1;
R3(px3>=px1 & px3>px2) = 1;
[DD,LL] = bwdist(R1+R2+R3);
R1(find(R1(LL))) = 1;
R2(find(R2(LL))) = 1;
R3(find(R3(LL))) = 1;
```

```
% figure
% hold on
% colormap([1 0 0;0 1 0;0 .5 1])
% plot(-10,-10,'s','MarkerFaceColor','r','MarkerEdgeColor','r')
% plot(-10,-10,'s','MarkerFaceColor','g','MarkerEdgeColor','g')
% plot(-10,-10,'s','MarkerFaceColor','b','MarkerEdgeColor','b')
% plot(-10,-10,'.','MarkerFaceColor','k','MarkerEdgeColor','k')
% plot(-10,-10,'o','MarkerFaceColor','k','MarkerEdgeColor','k')
% Floor_level = -(max([max(max(px1)) max(max(px2)) max(max(px3))])...
   );
% %Combine and plot 3D Gaussians with mean and covariance equal to...
    the class mean and covariance
% px_sum   = px1.*R1 + px2.*R2 + px3.*R3;
% Gousians = surf(X1,X2,px_sum,'LineStyle','none');
% for i = 1:(min(size(X1))-1)/24:min(size(X1))-1
%     i = round(i);
%     plot3(X1(:,i),X2(:,i),px_sum(:,i),'k');
%     plot3(X1(i,:),X2(i,:),px_sum(i,:),'k');
% end
% set(Gousians,'Cdatamapping','direct')
% set(Gousians,'Cdata',1*R1+2*R2+3*R3)
% %Plot the projected points from each class
% scatter3(Fishclass1(1,1:Dis:end),Fishclass1(2,1:Dis:end),...
   Floor_level*ones(1,ceil(size(Fishclass1,2)/Dis)),'r.')
% scatter3(Fishclass2(1,1:Dis:end),Fishclass2(2,1:Dis:end),...
   Floor_level*ones(1,ceil(size(Fishclass2,2)/Dis)),'g.')
% scatter3(Fishclass3(1,1:Dis:end),Fishclass3(2,1:Dis:end),...
   Floor_level*ones(1,ceil(size(Fishclass3,2)/Dis)),'b.')
% %Plot the means of the projected points from each class
% scatter3(mean1(1),mean1(2),Floor_level,'ko','filled')
% scatter3(mean2(1),mean2(2),Floor_level,'ko','filled')
% scatter3(mean3(1),mean3(2),Floor_level,'ko','filled')
% %Plot the Bayesian decision regions for the classes
% contour3(X1,X2,(R1 + Floor_level-.5),1,'r')
% contour3(X1,X2,(R2 + Floor_level-.5),1,'g')
% contour3(X1,X2,(R3 + Floor_level-.5),1,'b')
% %Set axis parameters
% xlabel('Y1')
% ylabel('Y2')
% legend('Class A','Class B','Class C','Class Point','Class Mean')
% axis([min(x1) max(x1) min(x2) max(x2) Floor_level -Floor_level])
% view(45,22.5)
% lightangle(45,22.5)
% light('Style','infinite');
% material shiny
% lighting phong
% grid on
% hold off
```

Listing A.11: Code/Classify/Classify.m

```
function [Tot_err,N_tot,A_in_A,A_in_B,A_in_C,A_err,B_in_B,B_in_A,...
    B_in_C,B_err,C_in_C,C_in_A,C_in_B,C_err, Fishclass1, Fishclass2...
    , Fishclass3]=...
     Classify_class(class1data,class2data,class3data,X1,X2,x1,x2,R1...
        ,R2,R3,data_mean,data_std,W)
Dis          = 1;
N_rec        = length(class1data(:,1))      ;
N_tot        = N_rec*3;
%%%%%%%%%%
% PART I %
%%%%%%%%%%
% Normalize each emission record using the parameters calculated ...
    during training
training_data      = [class1data;class2data;class3data];
data_mean          = ones(N_tot,1)*data_mean(1,:);
data_std           = ones(N_tot,1)*data_std(1,:);
training_data_norm = (training_data-data_mean)./data_std;
class1data         = training_data_norm(N_rec*0+1:N_rec*1,:);
class2data         = training_data_norm(N_rec*1+1:N_rec*2,:);
class3data         = training_data_norm(N_rec*2+1:N_rec*3,:);
```

```
%%%%%%%%%%%
% PART II %
%%%%%%%%%%%
% Project each class using the Fisher plane calculated during ...
    training
Fishclass1 = W * class1data';
Fishclass2 = W * class2data';
Fishclass3 = W * class3data';
% %%%%%%%%%%%%
% % PART III %
% %%%%%%%%%%%%
% Round and rescale data for confusion matrix calculations
scale1   = size(R1,2)-1;
scale2   = size(R1,1)-1;
X1_shift = min(min(X1));
X2_shift = min(min(X2));
X1_scale = max(max(X1)) - X1_shift;
X2_scale = max(max(X2)) - X2_shift;
X1_1 = 1 + round(((Fishclass1(1,:) - X1_shift) / X1_scale ) * ...
    scale1   );
X2_1 = 1 + round(((Fishclass1(2,:) - X2_shift) / X2_scale ) * ...
    scale2   );
X1_2 = 1 + round(((Fishclass2(1,:) - X1_shift) / X1_scale ) * ...
    scale1   );
X2_2 = 1 + round(((Fishclass2(2,:) - X2_shift) / X2_scale ) * ...
    scale2   );
X1_3 = 1 + round(((Fishclass3(1,:) - X1_shift) / X1_scale ) * ...
    scale1   );
X2_3 = 1 + round(((Fishclass3(2,:) - X2_shift) / X2_scale ) * ...
    scale2   );
% Initialize the individual classification terms
A_in_A = 0;
A_in_B = 0;
A_in_C = 0;
A_err  = 0;
```

```
B_in_A = 0;
B_in_B = 0;
B_in_C = 0;
B_err  = 0;
C_in_A = 0;
C_in_B = 0;
C_in_C = 0;
C_err  = 0;
n=find(X2_1 > 0 & X1_1 > 0 & X2_1 < scale2 & X1_1 < scale1);
A_in_A = sum(diag(R1(X2_1(n),X1_1(n))));
A_in_B = sum(diag(R2(X2_1(n),X1_1(n))));
A_in_C = sum(diag(R3(X2_1(n),X1_1(n))));
A_err = N_rec-length(n);
n=find(X2_2 > 0 & X1_2 > 0 & X2_2 < scale2 & X1_2 < scale1);
B_in_A = sum(diag(R1(X2_2(n),X1_2(n))));
B_in_B = sum(diag(R2(X2_2(n),X1_2(n))));
B_in_C = sum(diag(R3(X2_2(n),X1_2(n))));
B_err = N_rec-length(n);
n=find(X2_3 > 0 & X1_3 > 0 & X2_3 < scale2 & X1_3 < scale1);
C_in_A = sum(diag(R1(X2_3(n),X1_3(n))));
C_in_B = sum(diag(R2(X2_3(n),X1_3(n))));
C_in_C = sum(diag(R3(X2_3(n),X1_3(n))));
C_err = N_rec-length(n);
% Count total misclassifications
Tot_err = A_in_B + A_in_C + A_err + B_in_A + B_in_C + B_err + ...
    C_in_A + C_in_B + C_err;
% figure
% X1   = 1 + ((X1 - X1_shift) / X1_scale ) * scale1 ;
% X2   = 1 + ((X2 - X2_shift) / X2_scale ) * scale2 ;
% hold on
% plot(-10,-10,’x’,’MarkerFaceColor’,’r’,’MarkerEdgeColor’,’r’)
% plot(-10,-10,’+’,’MarkerFaceColor’,’g’,’MarkerEdgeColor’,’g’)
% plot(-10,-10,’*’,’MarkerFaceColor’,’b’,’MarkerEdgeColor’,’b’)
% % Plot the Baysian decision regions
% contour(X1,X2,R1,1,’r’)
% contour(X1,X2,R2,1,'g')
% contour(X1,X2,R3,1,'b')
% % Plot the test points
% scatter(X1_1(1:Dis:end),X2_1(1:Dis:end),'x','r')
% scatter(X1_2(1:Dis:end),X2_2(1:Dis:end),'+','g')
% scatter(X1_3(1:Dis:end),X2_3(1:Dis:end),'*','b')
% % Set axis paramiters
% xlabel('Y1')
% ylabel('Y2')
% legend('Class A Point','Class B Point','Class C Point')
% axis([1 scale1  1  scale2])
% set(gca,'XTick',(1:(scale1-1)/4:scale1))
% set(gca,'YTick',(1:(scale2-1)/4:scale2))
% set(gca,'XTickLabel',{min(x1),((min(x1))+(min(x1)+max(x1))/2 )...
   /2,...
%    (min(x1)+max(x1))/2,(((min(x1)+max(x1))/2)+max(x1))/2, max(x1...
   )})
% set(gca,'YTickLabel',{min(x2),((min(x2))+(min(x2)+max(x2))/2 )...
   /2,...
%    (min(x2)+max(x2))/2,(((min(x2)+max(x2))/2)+max(x2))/2, max(x2...
   )})
% axis square
% grid on
% hold off
```

A.5 DT-ℂWT Transformation

Listing A.12: Code/DualTree/dualtree.m

```
function w = dualtree(x, J, Faf, af)

% Dual-tree Complex Discrete Wavelet Transform
% USAGE:
%    w = dualtree(x, J, Faf, af)
% INPUT:
%   x - N-point vector
%       1) N is divisible by 2^J
%       2) N >= 2^(J-1)*length(af)
%   J - number of stages
%   Faf - filters for the first stage
%   af - filters for the remaining stages
% OUTPUT:
%   w - DWT coefficients
%      w{j}{1}, j = 1..J - real part
%      w{j}{2}, j = 1..J - imaginary part
%      w{J+1}{d} - lowpass coefficients, d = 1,2
% EXAMPLE:
%    x = rand(1, 512);
%    J = 4;
%    [Faf, Fsf] = FSfarras;
%    [af, sf] = dualfilt1;
%    w = dualtree(x, J, Faf, af);
%    y = idualtree(w, J, Fsf, sf);
%    err = x - y;
%    max(abs(err))
% WAVELET SOFTWARE AT POLYTECHNIC UNIVERSITY, BROOKLYN, NY
% http://taco.poly.edu/WaveletSoftware/
% normalization
x = x/sqrt(2);
% Tree 1
[x1 w{1}{1}] = afbDT(x, Faf{1});
for j = 2:J
    [x1 w{j}{1}] = afbDT(x1, af{1});
end
w{J+1}{1} = x1;
% Tree 2
[x2 w{1}{2}] = afbDT(x, Faf{2});
for j = 2:J
    [x2 w{j}{2}] = afbDT(x2, af{2});
end
w{J+1}{2} = x2;
```

Listing A.13: Code/DualTree/afbDT.m

```
function [lo, hi] = afbDT(x, af)

% Analysis filter bank
% USAGE:
%    [lo, hi] = afb(x, af)
% INPUT:
%    x - N-point vector, where
%             1) N is even
%             2) N >= length(af)
%    af - analysis filters
%    af(:, 1) - lowpass filter (even length)
%    af(:, 2) - highpass filter (even length)
% OUTPUT:
%    lo - Low frequecy output
%    hi - High frequency output
% EXAMPLE:
%    [af, sf] = farras;
%    x = rand(1,64);
%    [lo, hi] = afb(x, af);
%    y = sfb(lo, hi, sf);
%    err = x - y;
%    max(abs(err))
% WAVELET SOFTWARE AT POLYTECHNIC UNIVERSITY, BROOKLYN, NY
% http://taco.poly.edu/WaveletSoftware/

N = length(x);
L = length(af)/2;
x = cshift(x,-L);
% lowpass filter
lo = upfirdn(x, af(:,1), 1, 2);
lo(1:L) = lo(N/2+[1:L]) + lo(1:L);
lo = lo(1:N/2);
% highpass filter
hi = upfirdn(x, af(:,2), 1, 2);
```

```
hi(1:L) = hi(N/2+[1:L]) + hi(1:L);
hi = hi(1:N/2);
```

Listing A.14: Code/DualTree/idualtree.m

```
function y = idualtree(w, J, Fsf, sf)

% Inverse Dual-tree Complex DWT
% USAGE:
%    y = idualtree(w, J, Fsf, sf)
% INPUT:
%    w - DWT coefficients
%    J - number of stages
%    Fsf - synthesis filters for the last stage
%    sf - synthesis filters for preceeding stages
% OUTUT:
%    y - output signal
% See dualtree
% WAVELET SOFTWARE AT POLYTECHNIC UNIVERSITY, BROOKLYN, NY
% http://taco.poly.edu/WaveletSoftware/

% Tree 1
y1 = w{J+1}{1};
for j = J:-1:2
   y1 = sfbDT(y1, w{j}{1}, sf{1});
end
y1 = sfbDT(y1, w{1}{1}, Fsf{1});
% Tree 2
y2 = w{J+1}{2};
for j = J:-1:2
   y2 = sfbDT(y2, w{j}{2}, sf{2});
end
y2 = sfbDT(y2, w{1}{2}, Fsf{2});
% normalization
y = (y1 + y2)/sqrt(2);
```

Listing A.15: Code/DualTree/sfbDT.m

```
function y = sfbDT(lo, hi, sf)

% Synthesis filter bank
% USAGE:
%    y = sfb(lo, hi, sf)
% INPUT:
%    lo - low frqeuency input
%    hi - high frequency input
%    sf - synthesis filters
%    sf(:, 1) - lowpass filter (even length)
%    sf(:, 2) - highpass filter (even length)
% OUTPUT:
%    y - output signal
% See also afb
% WAVELET SOFTWARE AT POLYTECHNIC UNIVERSITY, BROOKLYN, NY
% http://taco.poly.edu/WaveletSoftware/

N = 2*length(lo);
L = length(sf);
lo = upfirdn(lo, sf(:,1), 2, 1);
hi = upfirdn(hi, sf(:,2), 2, 1);
y = lo + hi;
y(1:L-2) = y(1:L-2) + y(N+[1:L-2]);
y = y(1:N);
y = cshift(y, 1-L/2);
```

Listing A.16: Code/DualTree/FSfarras.m

```
function [af, sf] = FSfarras

% Farras filters organized for the dual-tree
% complex DWT.
% USAGE:
%    [af, sf] = FSfarras
% OUTPUT:
```

```
%    af{i}, i = 1,2 - analysis filters for tree i
%    sf{i}, i = 1,2 - synthesis filters for tree i
% See farras, dualtree, dualfilt1.
% WAVELET SOFTWARE AT POLYTECHNIC UNIVERSITY, BROOKLYN, NY
% http://taco.poly.edu/WaveletSoftware/

af{1} = [
                  0                  0
  -0.08838834764832  -0.01122679215254
   0.08838834764832   0.01122679215254
   0.69587998903400   0.08838834764832
   0.69587998903400   0.08838834764832
   0.08838834764832  -0.69587998903400
  -0.08838834764832   0.69587998903400
   0.01122679215254  -0.08838834764832
   0.01122679215254  -0.08838834764832
                  0                  0
 ];

sf{1} = af{1}(end:-1:1, :);

af{2} = [
   0.01122679215254                  0
   0.01122679215254                  0
  -0.08838834764832  -0.08838834764832
   0.08838834764832  -0.08838834764832
   0.69587998903400   0.69587998903400
   0.69587998903400  -0.69587998903400
   0.08838834764832   0.08838834764832
  -0.08838834764832   0.08838834764832
                  0   0.01122679215254
                  0  -0.01122679215254
];

sf{2} = af{2}(end:-1:1, :);
```

Listing A.17: Code/DualTree/dualfilt1.m

```
function [af, sf] = dualfilt1

% Kingsbury Q-filters for the dual-tree complex DWT
% USAGE:
%    [af, sf] = dualfilt1
% OUTPUT:
%    af{i}, i = 1,2 - analysis filters for tree i
%    sf{i}, i = 1,2 - synthesis filters for tree i
%    note: af{2} is the reverse of af{1}
% REFERENCE:
%    N. G. Kingsbury,  "A dual-tree complex wavelet
%    transform with improved orthogonality and symmetry
%    properties", Proceedings of the IEEE Int. Conf. on
%    Image Proc. (ICIP), 2000
% See dualtree
% WAVELET SOFTWARE AT POLYTECHNIC UNIVERSITY, BROOKLYN, NY
% http://taco.poly.edu/WaveletSoftware/
% These cofficients are rounded to 8 decimal places.

af{1} = [
   0.03516384000000                   0
                  0                   0
  -0.08832942000000  -0.11430184000000
   0.23389032000000                   0
   0.76027237000000   0.58751830000000
   0.58751830000000  -0.76027237000000
                  0   0.23389032000000
  -0.11430184000000   0.08832942000000
                  0                   0
                  0  -0.03516384000000
 ];

af{2} = [
                  0  -0.03516384000000
                  0                   0
  -0.11430184000000    0.08832942000000
                  0    0.23389032000000
   0.58751830000000   -0.76027237000000
   0.76027237000000    0.58751830000000
   0.23389032000000                   0
  -0.08832942000000   -0.11430184000000
                  0                   0
   0.03516384000000                   0
];

sf{1} = af{1}(end:-1:1, :);
sf{2} = af{2}(end:-1:1, :);
```

Bibliography

1. Agilent Technologies Inc., USA. *Agilent E3238 Signal Intercept and Collection Solutions: Family Overview, Publication 5989-1274EN*, Jul 2004.

2. Bayram, I. and I. W. Selesnick. "On the Dual-Tree Complex Wavelet Packet and M-band Transforms". *IEEE Trans on Signal Processing*, 56(6):2298–2310, Jun 2008. ISSN 1053-587X.

3. Birchenough, D.R., M.A. Temple, and M.J. Mendenhall. "Classifying Emissions from GSM Communication Devices Using Entropy-Based RF Fingerprinting". *2009 Military Communications Conference (MILCOM 2009)*. Oct 2009.

4. Carmona, R.A. "Wavelet Identification of Transients in Noisy Time Series". *Proceedings of SPIE: Wavelet Applications in Signal and Image Processing*, volume 2034, 392–400. Nov 1993.

5. Chen, Y., W. Trappe, and R. Martin. "Detecting and Localizing Wireless Spoofing Attacks". *IEEE Conference on Sensor, Mesh and AdHoc Comm and Nets (SECON)*, 193–202. Jun 2007.

6. Defence R&D Canada - Ottawa. "Interferometric Intrapulse Radar Receiver for Specific Emitter Identification and Direction-Finding". *Fact Sheet REW 224*, Jun 2007.

7. Del Marco, S.P. and J.E. Weiss. "M-band Wavepacket-Based Transient Signal Detector Using a Translation-Invariant Wavelet Transform". *Optical Engineering*, 33:2175–2182, Jul 1994.

8. Del Marco, S.P. and J.E. Weiss. "Improved Transient Signal Detection Using a Wavepacket-Based Detector with an Extended Translation-Invariant Wavelet Transform". *IEEE Trans on Signal Processing*, 45(4):841–850, Apr 1997. ISSN 1053-587X.

9. DeYoung D., et al. "Fulfilling Roosevelts' Vision for American Naval Power (1923-2005)". *NRL MR/1001–06-8951*, Jun 2006.

10. Duda, R.O., P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, Inc., New York, 2nd edition, 2001.

11. Dudczyk, J., J. Matuszewski, and M. Wnuk. "Applying the Radiated Emission to Specific Emitter Identification". *International Conference on Microwaves, Radar and Wireless Comm*, 431–434. May 2004.

12. Dudczyk, J. and M. Wnuk. "The Utilization of Unintentional Radiation for Identification of the Radiation Sources". *34th European Microwave Conference*, volume 2, 777–780. Oct 2004.

13. Durak, L. and O. Arikan. "Short-Time Fourier Transform: Two Fundamental Properties and an Optimal Implementation". *IEEE Transactions on Signal Processing*, 51(5):1231–1242, May 2003. ISSN 1053-587X.

14. Dutta, A. and G.V. Anand. "Detection of Transient Signals by Wavelet Packet Transform and Stochastic Resonance". *2004 IEEE Region 10 Conference*, volume 1, 251–254. Nov 2004.

15. Fabbroni, L., M. Vannucci, E. Cuoco, G. Losurdo, M. Mazzoni, and R. Stanga. "Wavelet Tests for the Detection of Transients in the VIRGO Interferometric Gravitational Wave Detector". *IEEE Trans on Instrumentation and Measurement*, 54(1):151–162, Feb 2005.

16. Fisher, R.A. "The Use of Multiple Measurements in Taxonomic Problems". *Annals of Eugenics*, 7:179–188, 1936.

17. Frisch, M. and H. Messer. "The Use of the Wavelet Transform in the Detection of an Unknown Transient Signal". *IEEE Trans on Information Theory*, 38(2), Mar 1992.

18. Frisch, M. and H. Messer. "Detection of a Transient Signal of Unknown Scaling and Arrival Time Using the Discrete Wavelet Transform". *1991 International Conference on Acoustics, Speech, and Signal Processing (ICASSP-91)*, volume 2, 1313–1316. Apr 1991. ISSN 1520-6149.

19. Frisch, M. and H. Messer. "Detection of a Known Transient Signal of Unknown Scaling and Arrival Time". *IEEE Trans on Signal Processing*, 42(7):1859–1863, Jul 1994. ISSN 1053-587X.

20. Gimelshteyn, M. *Classifying Commercial Receiver Emissions Using Fisher Discriminant Analysis*. Master's thesis, Air Force Institute of Technology, 2950 Hobson Way, WPAFB, OH, Mar 2006.

21. Grant, P. "Complex Communications". *IET Knowledge Network*, (Publication IETKN-0911), Jun 2009.

22. Hall, J. *Detection of Rogue Devices in Wireless Networks*. Ph.D. thesis, School of Computer Science, Carleton University, Aug 2006.

23. Hall, J., M. Barbeau, and E. Kranakis. "Detection of Transient in Radio Frequency Fingerprinting Using Signal Phase". *IASTED International Conference on Wireless and Optical Communications*. May 2003.

24. Hall, J., M. Barbeau, and E. Kranakis. "Using Transceiverprints for Anomaly Based Intrusion Detection". *3rd IASTED International Conference on Comm, Internet and Info Technology (CIIT)*. Nov 2004.

25. Hastie, T., R. Tibshirani, and J. Friedman. *Data Mining, Inference, and Prediction*. Springer, 2001.

26. Haykin, S. "Cognitive Radio: Brain-Empowered Wireless Communications". *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, Feb 2005.

27. Higuchi, T. "Approach to an Irregular Time Series on the Basis of the Fractal Theory". *Phys. D*, 31(2):277–283, 1988. ISSN 0167-2789.

28. Hotelling, H. "The Generalization of Student's Ratio". *Annals of Mathematical Statistics*, 1931.

29. IEEE Computer Society. *IEEE Std 802.11-2007*, Jun 2007.

30. Kawalec, A. "Mixed Method Based on Intrapulse Data and Radiated Emission to Emitter Sources Recognition". *15th International Conference on Microwaves, Radar and Wireless Comm.* May 2004.

31. Klein, R.W., M.A. Temple, and M.J. Mendenhall. "Application of Wavelet Denoising to Improve OFDM-Based Signal Detection and Classification". *Journal on Security and Communication Networks, Special Issue: Security in Next Generation Wireless Networks*, 2(6), 2009.

32. Klein, R.W., M.A. Temple, and M.J. Mendenhall. "Application of Wavelet-Based RF Fingerprinting to Enhance Wireless Network Security". *Journal on Communication Networks, Special Issue: Secure Wireless Networking*, Under Review, Jul 2009.

33. Klein, R.W., M.A. Temple, M.J. Mendenhall, and D.R. Reising. "Sensitivity Analysis of Burst Detection and RF Fingerprinting Classification Performance". *Proceedings of the 2009 IEEE International Conference on Communications*. Jun 2009.

34. Langley, L.E. "Specific Emitter Identification (SEI) and Classical Parameter Fusion Technology". *IEEE Western Electronics Show and Conference (WESCON)*, 277–381. Sep 1993.

35. Li, Q. and W. Trappe. "Detecting Spoofing and Anomalous Traffic in Wireless Networks via Forge-Rresistant Relationships". *IEEE Trans on Information Forensics and Security*, 2(4):793–808, Dec 2007.

36. MATLAB®. *Statistics Toolbox*. The Mathworks, Inc., r2007a edition.

37. Mendenhall, M.J. and E. Merenyi. "Generalized Relevance Learning Vector Quantization for Classification-Driven Feature Extraction from Hyperspectral Data". *American Society for Photogrammetry and Remote Sensing*. May 2006.

38. Mendenhall, M.J. and E. Merenyi. "Relevance-Based Feature Extraction from Hyperspectral Images in the Complex Wavelet Domain". *2006 IEEE Mountain Workshop on Adaptive and Learning Systems*, 24–29. Jul 2006.

39. Mendenhall, M.J. and E. Merenyi. "Relevance-Based Feature Extraction for Hyperspectral Images". *IEEE Transactions on Neural Networks*, 19(4):658–672, Apr 2008. ISSN 1045-9227.

40. MilitaryPeriscope.com. “SEI – SPECIFIC EMITTER IDENTIFICATION”. URL http://www.periscope.ucg.com/terms/t0000271.html.

41. O’Ruanaidh, J.J.K. and W.J. Fitzgerald. *Numerical Bayesian Methods Applied to Signal Processing*. New York.

42. Pacheco, R.A., O. Ureten, D. Hatzinakos, and N. Serinken. “Bayesian Frame Synchronization Using Periodic Preamble for OFDM-Based WLANs”. *IEEE Signal Processing Letters*, 12(7):524–527, Jul 2005. ISSN 1070-9908.

43. Percival, D.B. and A.T. Walden. *Wavelet Methods for Time Series Analysis*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Jul 2000.

44. Pitta, J.A., R.D. Hippenstiel, and M.P. Fargues. *Transient Detection Using Wavelets*. Master’s thesis, Naval Post Graduate School, MONTEREY CA, Mar 1995.

45. Prochazka, A. and M. Storek. “Wavelet Transform Use for Signal Classification by Self-Organizing Neural Networks”. *Artificial Neural Networks, 1995., Fourth International Conference on*, 295–299. Jun 1995.

46. Remley, K.A., C.A. Grosvenor, R.T. Johnk, D.R. Novotny, P.D. Hale, M.D. McKinley, A. Karygiannis, and E. Antonakakis. “Electromagnetic Signatures of WLAN Cards and Network Security”. *IEEE International Sym on Signal Processing and Info Tech*, 484–488. Dec 2005.

47. Rioul, O. and M. Vetterli. “Wavelets and Signal Processing”. *IEEE Signal Processing Magazine*, Oct 1991.

48. Saha, D., D. Grunwald, and D. Sicker. “Wireless Innovation Through Software Radios”. *ACM SIGCOMM Computer Communication Review*, 39(1), Jan 2009.

49. Selesnick, I.W. “Wavelet software at Polytechnic University, Brooklyn, NY”. http://taco.poly.edu/WaveletSoftware/.

50. Selesnick, I.W., R.G. Baraniuk, and N.C. Kingsbury. “The Dual-Tree Complex Wavelet Transform”. *IEEE Signal Processing Magazine*, 22(6):123–151, Nov 2005. ISSN 1053-5888.

51. Serinken, N. and O. Ureten. “Generalised Dimension Characterization of Radio Transmitter Turn-On Transients”. *IEE Electronics Letters*, 36(12):1064–1064, Jun 2000.

52. Shaw, D. and W. Kinsner. “Multifractal Modelling of Radio Transmitter Transients for Classification”. *IEEE WESCANEX 97: Communications, Power and Computing. Conference Proceedings.*, 306–312. May 1997.

53. Sheng, Y., K. Tan, G. Chen, D. Kotz, and A. Campbell. “Detecting 802.11 MAC Layer Spoofing Using Received Signal Strength”. *IEEE 27th Annual Conference on Computer Comm.* Apr 2008.

54. Suski, W.C., M.A. Temple, M.J. Mendenhall, and R.F. Mills. "Radio Frequency Fingerprinting Commercial Communication Devices to Enhance Electronic Security". *International Journal Electronic Security and Digital Forensics*, 1(3):301–322, 2008.

55. Suski, W.C., M.A. Temple, M.J. Mendenhall, and R.F. Mills. "Using Spectral Fingerprints to Improve Wireless Network Security". *Proceedings of the 2008 IEEE Global Communications Conference*. Nov 2008.

56. Tactical SIGINT Technology (TST) Program Management Office. "Using DTIC as a Program Management Tool". *2009 DTIC Conference on Defense Scientific and Technical Information*. Apr 2009.

57. Tekbas, O.H. and N. Serinken. "Transmitter Fingerprinting from Turn-on Transients". *NATO SET Panel Symposium on Passive and LPI Radio Frequency Sensors*. Warsaw, Poland, April 2001.

58. Tekbas, O.H., N. Serinken, and O. Ureten. "An Experimental Performance Evaluation of a Novel Radio-Transmitter Identification System Under Diverse Environmental Conditions". *Canadian Journal of Electrical and Computer Engineering*, 29(3):203–209, Jul 2004. ISSN 0840-8688.

59. Tekbas, O.H., O. Ureten, and N. Serinken. "Improvement of Transmitter Identification System for Low SNR Transients". *Electronics Letters*, 40(3):182–183, Feb 2004. ISSN 0013-5194.

60. Terry, I. "Networked SEI in Fleet Battle Experiment Juliet (FBE-J)", Fall 2002.

61. Thiruvengadam, S.J., P. Chinnadurai, M.T. Thirumalai Kumar, and V. Abhaikumar. "Wavelet Based Signal Processing Algorithms for Early Target Detection". *2003 Conference on Convergent Technologies for Asia-Pacific Region*, volume 3, 1175–1179. Oct 2003.

62. Toonstra, J. and W. Kinsner. "Transient Analysis and Genetic Algorithms for Classification". *IEEE WESCANEX 95. Communications, Power, and Computing. Conference Proceedings.*, volume 2, 432–437. May 1995.

63. Ureten, O., R.A. Pacheco, N. Serinken, and D. Hatzinakos. "Bayesian Frame Synchronization for 802.11a WLANs: Experimental Results". *Canadian Conference on Electrical and Computer Engineering, 2005.*, 884–887. May 2005. ISSN 0840-7789.

64. Ureten, O. and N. Serinken. "Detection of Radio Transmitter Turn-On Transients". *IEE Electronics Letters*, 35(23):1996–1997, Nov 1999.

65. Ureten, O. and N. Serinken. *Detection, Characterisation and Classification of Radio Transmitter Turn-On Transients*, chapter Multisensor Fusion, 611616. Kluwer Academic Publishers, Dordrecht, Netherlands, 2000.

66. Ureten, O. and N. Serinken. "Bayesian Detection of WiFi Transmitter RF Fingerprints". *IEE Electronics Letters*, 41(6):373–374, Mar 2005.

67. Ureten, O. and N. Serinken. "Improved Coarse Timing for Burst Mode OFDM". *IEEE Global Telecommunications Conference, 2007. GLOBECOM '07.*, 2841–2846. Nov. 2007.

68. Ureten, O. and N. Serinken. "Wireless Security Through RF Fingerprinting". *Canadian Journal of Electrical and Computer Engineering*, 32(1):27–33, Winter 2007.

69. US Air Force Sensors Directorate (AFRL/SY), Wright-Patterson AFB, OH, 45433. "Vision Statement". http://www.wpafb.af.mil/afrl/sn/.

70. Witten, Ian H. and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

71. Woelfle, M., M.A. Temple, M. Mullins, and M.J. Mendenhall. "Detecting, Identifying and Locating Bluetooth Devices Using RF Fingerprints". *2009 Military Communications Conference (MILCOM 2009)*. Oct 2009.

72. Youell, N.T. and M. Niknam. "4G: Fact or Fiction?" *Communications Technical Journal*, 6(1), Sep 2008.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*
03–09–2009

2. REPORT TYPE
Doctoral Dissertation

3. DATES COVERED *(From — To)*
September 2006-September 2009

4. TITLE AND SUBTITLE
Application of Dual-Tree ℂomplex Wavelet Transforms to Burst Detection and RF Fingerprint Classification

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER
ENG 09-300

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)
Klein, Randall W., Major, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765 DSN: 785-3636

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/DEE/ENG/09-12

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Attn: AFRL/RYRE (Dr. Vasu Chakravarthy)
2241 Avionics Circle, Bldg 620
WPAFB OH 45433-7734
(937)255-5579
Vasu.Chakravarthy@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RYRE

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This work addresses various Open Systems Interconnection (OSI) Physical (PHY) layer mechanisms to extract and exploit RF waveform features (”fingerprints”) that are inherently unique to specific devices and that may be used to provide hardware specific identification (manufacturer, model, and/or serial number). This is addressed by applying a Dual-Tree ℂomplex Wavelet Transform (DT-ℂWT) to improve burst detection and RF fingerprint classification. A ”Denoised VT” technique is introduced to improve performance at lower SNRs, with denoising implemented using a DT-ℂWT decomposition prior to Traditional VT processing. A newly developed Wavelet Domain (WD) fingerprinting technique is presented using statistical WD fingerprints with Multiple Discriminant Analysis/Maximum Likelihood (MDA/ML) classification. The statistical fingerprint features are extracted from coefficients of a DT-ℂWT decomposition. Relative to previous Time Domain (TD) results, the enhanced WD statistical features provide improved device classification performance. Additional performance sensitivity results are presented to demonstrate WD fingerprinting robustness for variation in burst location error, MDA/ML training and classification SNRs, and MDA/ML training and classification signal types. For all cases considered, the WD technique proved to be more robust and exhibited less sensitivity when compared with the TD Technique.

15. SUBJECT TERMS
RF Fingerprinting, OFDM, Dual-Tree Complex Wavelet Transform

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Michael A. Temple (ENG)
					19b. TELEPHONE NUMBER *(include area code)*
U	U	U	UU	150	(937)255-3636x4279; email:michael.temple@afit.edu