



**RADIO FREQUENCY-BASED
MICROCONTROLLER ANOMALY
DETECTION**

THESIS

Justin P. Wylie, Capt, USAF
AFIT-ENG-MS-16-M-053

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENG-MS-16-M-053

RADIO FREQUENCY-BASED MICROCONTROLLER
ANOMALY DETECTION

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
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 Master of Science in Electrical Engineering

Justin P. Wylie, B.S.E.E.

Capt, USAF

March 24 2016

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENG-MS-16-M-053

RADIO FREQUENCY-BASED MICROCONTROLLER
ANOMALY DETECTION
THESIS

Justin P. Wylie, B.S.E.E.
Capt, USAF

Committee Membership:

Major Samuel J. Stone, PhD
Chair

Dr. Michael A. Temple
Member

Dr. Barry E. Mullins
Member

Abstract

The research presented here focuses on applying the Correlation-Based Anomaly Detection (CBAD) process to a Microcontroller Unit (MCU) representative of the microcontrollers found in embedded systems as a means of either detecting an anomalous behavior, or verifying that the device is behaving normally. This research utilizes a previously developed method to collect Unintentional Radio Frequency (RF) Emissions (UREs) from $N_D = 5$ Texas Instruments MSP430F5529 MCUs from two different manufacturing lots. Once captured, the time domain signals are processed and used as one of two inputs into the CBAD algorithm. The second input is a SNR_A dependent reference signal generated based on the collection of $N_{trng} = 500$ signals from two of the devices, one from each manufacturing lot. The output of the CBAD process is a correlation statistic z_v , which is compared with a detection threshold t_v that varies with SNR_A to determine whether the input signal is running the *Anomalous* or *Normal* Advanced Encryption Standard (AES) program.

Performance for the CBAD process was evaluated using Receiver Operating Characteristic (ROC) curves against an arbitrary benchmark of Equal Error Rate (EER) of $\text{EER} \leq 10\%$, or equivalently a False Anomaly Detection Rate (FADR) of $\text{FADR} \leq 10\%$ while maintaining a False Normal Verification Rate (FNVR) of $\text{FNVR} \leq 10\%$, consistent with previous Air Force Institute of Technology (AFIT) research. When comparing an unknown signal from a particular manufacturing lot to a reference signal from the same lot, the benchmarks were met in all cases when the analysis Signal-to-Noise Ratio $\text{SNR}_A \geq 0\text{dB}$. Conversely, when comparing an unknown signal from a particular manufacturing lot to a reference signal from a different lot, the benchmarks were not met for any investigated SNR_A .

*To my wife, for her love, patience and understanding of all the time away while
doing homework or performing research.*

*To my parents, for their unending support of all my endeavors and for instilling in
me a desire to learn.*

Acknowledgements

I would like to thank my academic advisor, Major Samuel Stone, for supporting me throughout this research effort. His guidance and knowledge were indispensable and very much appreciated. I would also like to thank my research committee members, Dr. Mullins and Dr. Temple for their support.

Justin P. Wylie

Table of Contents

	Page
Abstract	iv
Acknowledgements	vi
List of Figures	ix
List of Tables	xiv
List of Abbreviations	xv
I. Introduction	1
1.1 Motivation	1
1.1.1 Software-based Vulnerabilities	3
1.1.2 Hardware-based Vulnerabilities	4
1.1.3 Encryption as Vulnerable Process	4
1.2 Approach	5
1.2.1 Emission Collection	7
1.2.2 Correlation	7
1.2.3 Operating Condition Verification	8
1.3 Research Contributions	8
1.4 Document Organization	9
II. Background	11
2.1 Microcontrollers	11
2.2 Vulnerabilities	12
2.3 Current Malware Prevention/Detection Methods	13
2.4 Advanced Encryption Standard Algorithm	14
2.4.1 Motivation To Use AES	14
2.4.2 AES Primer	14
2.5 RF Signal Collection	16
2.5.1 Correlation	17
2.5.2 Operation Verification	18
III. Methodology	20
3.1 Microcontroller Device Description	21
3.2 Microcontroller Operating Conditions	23
3.2.1 <i>Normal</i> Operation	24
3.2.2 <i>Anomalous</i> Operation	24
3.3 Radio Frequency Emission Collection	24
3.3.1 Microcontroller Collection Configuration	25

	Page
3.3.2 RF Probe Placement	25
3.3.3 Sampling and Triggering	28
3.4 Post-Collection Processing	29
3.4.1 Filtering	30
3.4.2 Signal-to-Noise Ratio Scaling	31
3.4.3 Down-conversion	32
3.4.4 Hilbert Transform	33
3.4.5 Jitter Removal	34
3.4.6 Non-Uniform Decimation	35
3.5 Correlation-Based Anomaly Detection Algorithm	36
3.5.1 Statistic Generation	36
3.5.2 Reference Signal Generation	38
3.5.3 Threshold Selection	38
3.5.4 Declaration	39
IV. Results	41
4.1 Performance Evaluation Criteria	41
4.2 Like-Device Performance	43
4.3 Cross-Device Performance	47
4.3.1 Cross-Lot	47
4.3.2 Like-Lot	49
4.4 Lot Discrimination	57
4.5 Consolidated Results	62
V. Conclusion	64
5.1 Research Summary	64
5.1.1 Purpose	64
5.1.2 CBAD Results	65
5.2 Future Work	66
Appendix A. Complete Result Set	68
Appendix B. AES C Code	98
Bibliography	109

List of Figures

Figure	Page
1	Examples of Embedded Systems 2
2	Open Systems Interconnect (OSI) Model 6
3	Advanced Encryption Standard (AES) Input Bytes 15
4	General Form of AES Encryption/Decryption 16
5	Overall Process of Detecting Operating Condition 20
6	Device Under Test (DUT) 22
7	Custom-made Mount 25
8	In-line Filter Impulse Response 26
9	Collected <i>Normal</i> Signal 27
10	Power Spectral Density 28
11	Trigger waveform 29
12	Post-collection Processing Block Diagram 30
13	Processing Filter Frequency Response 31
14	Single Filtered <i>Normal</i> Signal 31
15	Low Pass Filter (LPF) Filter Frequency Response 33
16	Single Down-Converted <i>Normal</i> Signal 33
17	Single Hilbert-transformed <i>Normal</i> Signal 34
18	Single Fully-transformed <i>Normal</i> Signal 36
19	Correlation-Based Anomaly Detection (CBAD) Process 37
20	Training Statistic Probability Mass Function (PMF) 39
21	Single Device Performance, DUT3 45
22	Single Device Performance, DUT11 46

Figure		Page
23	Blue Lot Reference Against Red Lot Device DUT6 Receiver Operating Characteristic (ROC) Curve	48
24	Red Lot Reference Against Blue Lot Device DUT4 ROC Curve	49
25	Blue Cross-Device Results, No Threshold Reset	51
26	Red Cross-device Results, No Threshold Reset	52
27	PMFs of DUT4 statistics	54
28	DUT4 True Anomaly Detection Rate (TADR) After Threshold Reset	55
29	PMFs of DUT5 statistics	56
30	DUT5 TADR After Threshold Reset	57
31	Lot Discrimination ROC Curve, Blue Reference	58
32	PMFs Lot Discrimination, Blue Reference	59
33	Lot Discrimination SNR_A vs Cross-lot Detection, Blue Reference	60
34	Lot Discrimination ROC Curve, Red Reference	60
35	PMFs Lot Discrimination, Red Reference	61
36	Lot Discrimination SNR_A vs Cross-lot Detection, Red Reference	62
37	Blue Reference vs DUT3 ROC Curve	68
38	SNR_A vs. TADR Blue Reference vs DUT3	69
39	Statistic PMFs Blue Reference vs DUT3	69
40	Blue Reference vs DUT4 ROC Curve	70
41	SNR_A vs. TADR Blue Reference vs DUT4, pre-regeneration	70
42	SNR_A vs. TADR Blue Reference vs DUT4 post-regeneration	71

Figure	Page
43	Statistic PMFs Blue Reference vs DUT4 pre-regeneration 71
44	Statistic PMFs Blue Reference vs DUT4 post-regeneration 72
45	Blue Reference vs DUT5 ROC Curve 72
46	SNR_A vs. TADR Blue Reference vs DUT5 pre-regeneration 73
47	SNR_A vs. TADR Blue Reference vs DUT5 post-regeneration 73
48	SNR_A vs. TNVR Blue Reference vs DUT5 pre-regeneration 74
49	SNR_A vs. TNVR Blue Reference vs DUT5 post-regeneration 74
50	Statistic PMFs Blue Reference vs DUT5 pre-regeneration 75
51	Statistic PMFs Blue Reference vs DUT5 post-regeneration 75
52	Blue Reference vs DUT6 ROC Curve 76
53	SNR_A vs. TADR Blue Reference vs DUT6 pre-regeneration 76
54	SNR_A vs. TADR Blue Reference vs DUT6 post-regenerat 77
55	SNR_A vs. TNVR Blue Reference vs DUT6 pre-regeneration 77
56	SNR_A vs. TNVR Blue Reference vs DUT6 post-regeneration 78
57	Statistic PMFs Blue Reference vs DUT6 pre-regeneration 78
58	Statistic PMFs Blue Reference vs DUT6 post-regeneration 79
59	Blue Reference vs DUT11 ROC Curve 79
60	SNR_A vs. TADR Blue Reference vs DUT11 pre-regeneration 80

Figure		Page
61	SNR _A vs. TADR Blue Reference vs DUT4 post-regeneration	80
62	SNR _A vs. TNVR Blue Reference vs DUT11 pre-regeneration	81
63	SNR _A vs. TNVR Blue Reference vs DUT11 post-regeneration	81
64	Statistic PMFs Blue Reference vs DUT11 pre-regeneration	82
65	Statistic PMFs Blue Reference vs DUT11 post-regeneration	82
66	Red Reference vs DUT3 ROC Curve	83
67	SNR _A vs. TADR Red Reference vs DUT3 pre-regeneration	83
68	SNR _A vs. TADR Red Reference vs DUT3 post-regeneration	84
69	SNR _A vs. TNVR Red Reference vs DUT3 pre-regeneration	84
70	SNR _A vs. TNVR Red Reference vs DUT3 post-regeneration	85
71	Statistic PMFs Red Reference vs DUT3 pre-regeneration.....	85
72	Statistic PMFs Red Reference vs DUT3 post-regeneration	86
73	Red Reference vs DUT4 ROC Curve	86
74	SNR _A vs. TADR Red Reference vs DUT4 pre-regeneration	87
75	SNR _A vs. TADR Red Reference vs DUT4 post-regeneration	87
76	SNR _A vs. TNVR Red Reference vs DUT4 pre-regeneration	88

Figure		Page
77	SNR _A vs. TNVR Red Reference vs DUT4 post-regeneration	88
78	Statistic PMFs Red Reference vs DUT4 pre-regeneration	89
79	Statistic PMFs Red Reference vs DUT4 post-regeneration	89
80	Red Reference vs DUT5 ROC Curve	90
81	SNR _A vs. TADR Red Reference vs DUT5 pre-regeneration	90
82	SNR _A vs. TADR Red Reference vs DUT5 post-regeneration	91
83	Statistic PMFs Red Reference vs DUT5 pre-regeneration	91
84	Statistic PMFs Red Reference vs DUT5 post-regeneration	92
85	Red Reference vs DUT6 ROC Curve	92
86	SNR _A vs. TADR Red Reference vs DUT6 pre-regeneration	93
87	SNR _A vs. TADR Red Reference vs DUT6 post-regeneration	93
88	SNR _A vs. TNVR Red Reference vs DUT6 pre-regeneration	94
89	SNR _A vs. TNVR Red Reference vs DUT6 post-regeneration	94
90	Statistic PMFs Red Reference vs DUT6 pre-regeneration	95
91	Statistic PMFs Red Reference vs DUT6 post-regeneration	95
92	Red Reference vs DUT11 ROC Curve	96
93	SNR _A vs. TADR Red Reference vs DUT11	96
94	Statistic PMFs Red Reference vs DUT11	97

List of Tables

Table		Page
1	Research Contributions	10
2	Devices Under Test (DUTs) and Lot Numbers	22
3	Table of Possible Outcomes	42
4	Overall Research Results	63

List of Abbreviations

Abbreviation	Page
Microcontroller Unit (MCU)	1
Correlation-Based Anomaly Detection (CBAD)	1
Software (SW)	1
Supervisory Control and Data Acquisition (SCADA)	1
Industrial Control Systems (ICSs)	2
Central Processing Units (CPUs)	2
Random Access Memory (RAM)	2
Read-Only Memory (ROM)	2
Hardware (HW)	4
Integrated Circuit (IC)	4
Defense Advanced Research Projects Agency (DARPA)	4
Supply Chain Hardware Integrity for Electronics Defense (SHIELD)	4
National Institute of Standards and Technology (NIST)	5
United States (US)	5
Advanced Encryption Standard (AES)	5
Device Under Test (DUT)	5
Radio Frequency (RF)	5
Open Systems Interconnect (OSI)	6
Differential Power Analysis (DPA)	6
Radio Frequency Distinct Native Attributes (RF-DNA)	6
Programmable Logic Controller (PLC)	7
Intentional Radio Frequency Emission (IRE)	7

Abbreviation	Page
Unintentional Radio Frequency Emission (URE)	7
Air Force Institute of Technology (AFIT)	7
Region of Interest (ROI)	7
Multiple Discriminant Analysis/Maximum Likelihood (MDA/ML)	8
Generalized Relevance Learning Vector Quantized-Improved (GRLVQI)	8
Identification (ID)	8
Radio Frequency Intelligence (RFINT)	8
Dimensional Reduction Analysis (DRA)	10
Input/Output (I/O)	11
Intrusion Detection and Prevention Systems (IDPSs)	13
Human-Machine Interface (HMI)	13
De-Militarized Zone (DMZ)	13
Federal Information Processing Standards Publication 197 (FIPS-197)	14
Side Channel Analysis (SCA)	14
Data Encryption Standard (DES)	14
Triple Data Encryption Standard (3DES)	14
Receiver Operating Characteristic (ROC)	18
Commercial Off The Shelf (COTS)	21
Texas Instruments (TI)	21
Light-Emitting Diode (LED)	23
Discrete Fourier Transform (DFT)	27
Power Spectral Density (PSD)	28
Low Pass Filter (LPF)	28

Abbreviation	Page
Band-Pass Filter (BPF)	30
Signal-to-Noise Ratio (SNR)	31
Additive White Gaussian Noise (AWGN)	32
Intermediate Frequency (IF)	32

RADIO FREQUENCY-BASED MICROCONTROLLER ANOMALY DETECTION

I. Introduction

This chapter introduces the research topic by providing the motivation for performing the research and describing the approach taken to achieve the research goals. Section 1.1 discusses the prevalence and vulnerabilities of Microcontroller Unit (MCU) dependent systems in both military and civilian environments. Section 1.2 proposes using the Correlation-Based Anomaly Detection (CBAD) process developed by Stone to detect malicious Software (SW) running on a MCU [81]. Section 1.3 outlines the research contributions of this work, and how it differs from previous and current research. Section 1.4 outlines the remaining chapters of this document.

1.1 Motivation

Most people in today's society own a personal computer, but few people realize that personal computers are a minority in the overall number of computers in the world. The majority of computing platforms fall into the classification of *embedded systems*, which are responsible for running the systems in your car, home appliances, television remote, and many more everyday items [48]. Figure 1 shows several more examples of how embedded systems control the devices in our everyday lives, but is far from all-inclusive.

In addition to the everyday items listed above, embedded systems also control vitally important systems, such as military smart card and datalink encryption, avionics systems in airplanes, and Supervisory Control and Data Acquisition (SCADA) net-

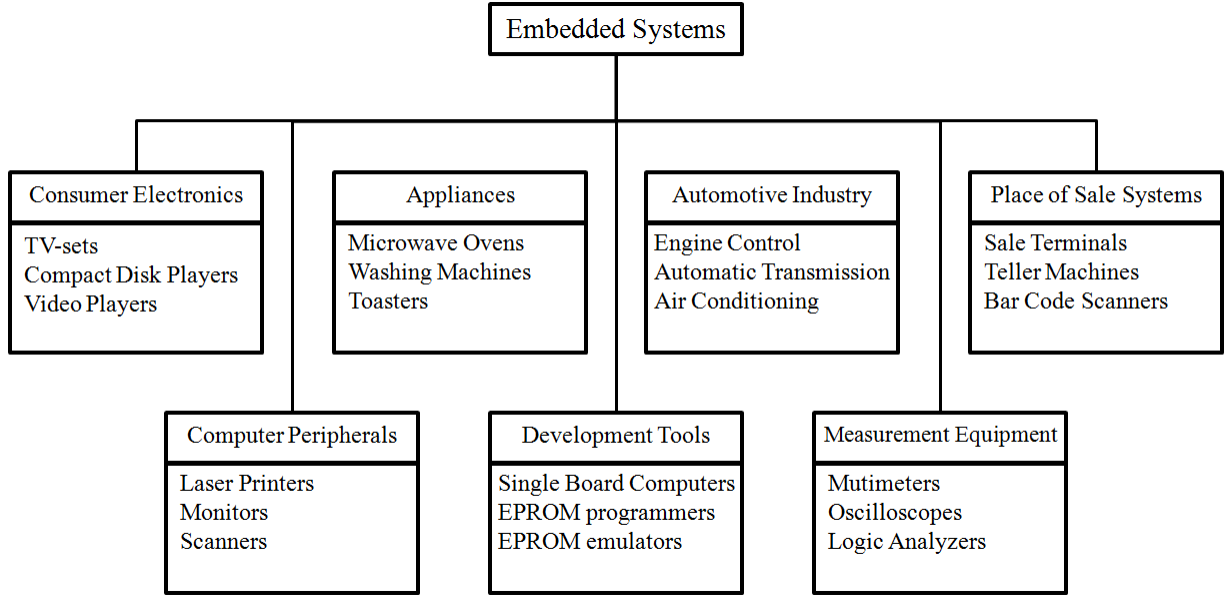


Figure 1. Examples of embedded systems in the civilian sector. [48]

works in Industrial Control Systems (ICSs) that are used in power plants, water plants and other crucial services. If any of these systems were to fail or become compromised, it could result in substantial damage costs, military mission failure, or even death.

Modern embedded systems are often based on *microcontrollers*, or MCUs, which by some accounts make up approximately half of all Central Processing Units (CPUs) sales in the world [54]. Microcontrollers differ from general microprocessors in that they come with peripherals, such as on-board Random Access Memory (RAM) and Read-Only Memory (ROM) and are used to perform single specific tasks, whereas microprocessors do not come packaged with any peripherals and are used in applications with multiple unspecific tasks. With the increased use of embedded systems and MCUs comes an increased risk of cyber attack due to software and hardware vulnerabilities. These attacks are especially concerning when they impact critical infrastructure or when targeting key functions such as data encryption.

1.1.1 Software-based Vulnerabilities.

Experts at McAfee, a well-known cyber defense organization, believe that as our critical infrastructure becomes more Internet-facing, cyber attacks against ICSs become more likely [58]. McAfee also notes that some of these systems use no more than the default login credentials for protection. Vulnerabilities such as these make it much easier for potential adversaries to gain access to critical infrastructure systems.

There are several examples of high-profile attacks against ICSs in recent history, such as STUXNET in 2010 [1], an attack against a German steel mill [5], and the most recent publicized attack in 2015 which caused a power outage in Ukraine [32]. A survey was performed by the Aspen Institute polled Information Technology (IT) and security professionals with an average 12 years of security experience, representing 625 critical infrastructure organizations and found that 48% of respondents believe a cyber attack will take down critical infrastructure with potential loss of life [2].

In 2003 and 2013, a Presidential Directive [10] and Executive Order [66], respectively, were issued to prioritize defense of these critical assets. However, despite efforts to protect ICSs they remain vulnerable. In 2015, the ICS Cyber Emergency Response Team (ICS-CERT) responded to 295 reported incidents involving critical infrastructure in the United States (US), with about half of the attacks occurring against the critical manufacturing and energy categories [45]. The number of attacks is likely higher, since operators may choose not to disclose incidents to ICS-CERT.

MCUs and embedded systems responsible for monitoring or operating numerous peripherals within an ICS facility may introduce cyber attack risks. It has been shown that an attacker can mimic a firmware update, and remotely program a microcontroller with malicious code [92]. Because microcontrollers do not typically run Anti-Virus (AV) SW or Host-Based Intrusion Detection System (HIDS), it is imperative to determine a non-destructive method that can detect attacks or anomalies

in real time. Additionally, network security hardware such as firewalls and Network Intrusion Detection System (NIDS) are likely not configured to focus on SCADA/ICS specific attacks. Early detection of a zero-day attack may help minimize damage to the system and prevent a catastrophic failure.

1.1.2 Hardware-based Vulnerabilities.

Microcontrollers can also be susceptible to Hardware (HW)-based attacks, which include counterfeit parts and HW trojans. Since the 1960s, more Integrated Circuit (IC) device manufacturers have been outsourcing the fabrication of semiconductor devices to Taiwan, China, and other countries where fabrication efforts are less expensive [1]. Outsourcing the manufacturing of semiconductor devices introduces the possibility of these nations embedding malicious HW trojans into the IC or selling counterfeit devices which can fail prematurely or not meet design specifications. In fact, it is estimated that 10.0% of electronic parts in use are counterfeit substitutions, resulting in immediate market losses of about \$200B USD [63].

The defense agency, Defense Advanced Research Projects Agency (DARPA), has been investigating methods to combat counterfeit devices with the Supply Chain Hardware Integrity for Electronics Defense (SHIELD) and TRUST in ICs programs. The SHIELD program implements passive, un-powered sensors capable of detecting attempts to image, de-solder or de-lid an IC [7]. The DARPA TRUST in ICs program attempts to develop technologies which will ensure the trust of ICs used in military systems that are fabricated in untrusted conditions [8].

1.1.3 Encryption as Vulnerable Process.

Encryption is used to secure communication channels by obfuscating the original message in such a way that if intercepted, an attacker will be unable to decipher the

message without the secret key. Sound encryption processes are key in everything from securing national secrets to protecting citizens from theft. It is critical that programs and devices processing sensitive information employ effective means of data encryption.

In 2002, National Institute of Standards and Technology (NIST) established the Rijndael cipher as the official encryption standard of the US government [9]. With this announcement, the Rijndael algorithm became the Advanced Encryption Standard (AES). As the official standard, the AES algorithm is also a common target for research into Side Channel Analysis (SCA), Differential Power Analysis (DPA), and other cyber attack methods. For this reason, and the ease of access to a readily available implementation on the Device Under Test (DUT), AES is the encryption algorithm selected for this research.

1.2 Approach

Current implementations of Intrusion Detection and Prevention Systems (IDPSs) rely on detecting bit-level changes and anomalies at the upper layers (Layers 2-7) of the Open Systems Interconnect (OSI) 7-layer network model seen in Figure 2. However, attackers are constantly finding new methods of bypassing these security systems [76]. One possible alternative is to focus on detecting anomalies at the Physical layer (Layer 1) of the OSI model instead of the upper layers, since it is much more difficult to purposely alter the Radio Frequency (RF) emissions of a device to replicate authorized emissions than it is to manipulate bits and operations in the upper layers (Data Link through Application layers). The proposed method institutes a *white list* approach, which is a list approved entities (i.e., operations, devices, people, etc.) for a given system. It can be difficult to know what method an adversary will use in order to attack a system and therefore a *black list* of unapproved operations can

grow too large to be feasible. The benefit of a white list as opposed to a black list is that training on a “known good” configuration and rejecting any other configuration can be much more resource effective, and can prevent zero-day attacks that are not on current black lists.

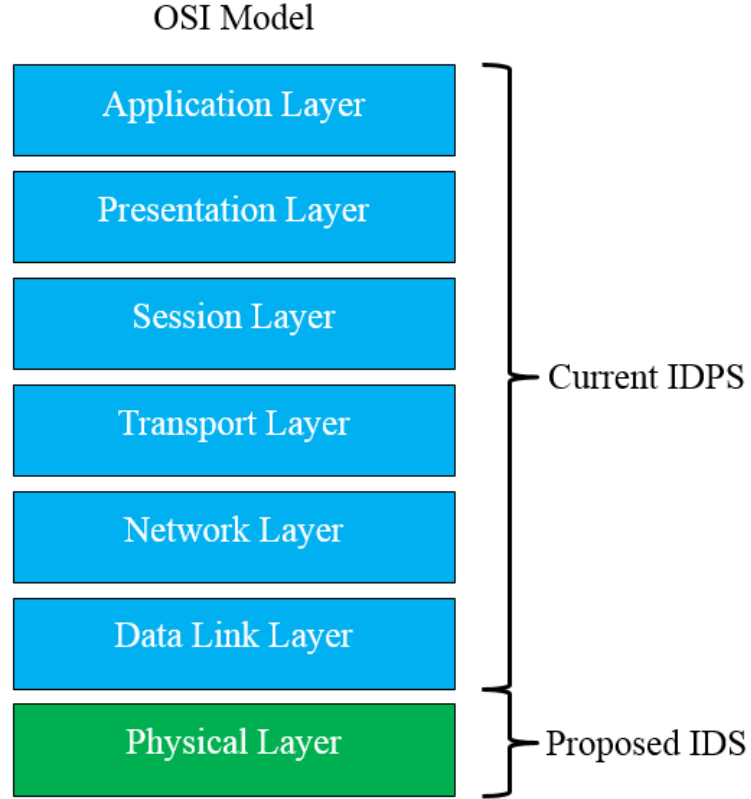


Figure 2. OSI 7-layer network model

Classification of devices and operations based on attributes of the Physical Layer have been demonstrated in a large body of research [13, 15, 18–20, 22, 23, 25–27, 33, 39, 42, 47, 72–74, 81, 95]. Most of the research outside of Air Force Institute of Technology (AFIT) focuses on the use of DPA. AFIT’s focus has primarily been Radio Frequency Distinct Native Attributes (RF-DNA), which are generated based on the statistics of the amplitude, phase, and frequency of collected emissions. While using RF-DNA has proven effective, it can also be very resource-intensive when determining which statistic to use when classifying a rogue device or anomalous operation. However, the

research performed by Stone [81] utilized one statistic, namely the correlation statistic, to accurately detect anomalous operations and discriminate between Programmable Logic Controller (PLC) devices. A PLC is a special purpose computer used in ICS applications. This research will build upon Stone’s research and apply the principle to MCUs, which are more versatile and ubiquitous than PLCs and used in a broader range of applications.

1.2.1 Emission Collection.

Most of the research on detecting anomalies and classifying devices based on Physical Layer attributes is based on devices that are designed to broadcast information. The signals collected from such devices are called Intentional Radio Frequency Emission (IRE). The MCU emissions considered for this research do not fall into this category of device, and so this research utilizes the collection of Unintentional Radio Frequency Emission (URE). Previous AFIT work provided the framework for collecting these emissions and is used in this research [13, 14, 81, 96].

There are several distinct differences between IREs and UREs. The average power of UREs is substantially lower than that of IRE, so UREs are collected using a highly sensitive RF probe with a relatively large bandwidth instead of an antenna tuned to a specific frequency and narrower bandwidth. In addition to a low average power, UREs do not have an engineered structure, and therefore require a different technique when determining the Region of Interest (ROI) and processing of the captured signal after collection.

1.2.2 Correlation.

As mentioned before, utilizing RF-DNA to classify devices or device operations can be resource intensive. An alternative method developed by Stone [81] uses cor-

relation to generate a statistic, and is used in this research. Correlation provides an effective anomaly detection method, while maintaining predictable computational complexity when compared to classification algorithms such as Multiple Discriminant Analysis/Maximum Likelihood (MDA/ML) and Generalized Relevance Learning Vector Quantized-Improved (GRLVQI).

1.2.3 Operating Condition Verification.

Previous AFIT work focuses on device classification and verification [14, 15, 79, 80, 96]. Device classification is a one-to-many comparison that assigns an unknown device to one of several possible classes within an authorized device pool. Regardless of whether the unknown device belongs to any of the authorized classes, it will be classified (correctly or incorrectly) as one of the classes and is analogous to a black list approach. The verification process is a one-to-one comparison which evaluates “how much like” an unknown entity resembles a pre-established reference and is analogous to a white list approach. This research uses the same principles as device Identification (ID) verification found in several bodies of work [15, 46, 71, 81].

1.3 Research Contributions

This research focuses on expanding the use of the CBAD process developed by Stone from PLCs to MCUs to discriminate between operating conditions, which can in turn detect potentially malicious SW or firmware. Successful research would provide a resource-efficient, non-destructive method of detecting potentially catastrophic zero-day attacks against embedded systems, and address a specific vulnerability affecting AES encryption. Table 1 summarizes the AFIT research contributions to the Radio Frequency Intelligence (RFINT) field, and what this research will add to the current body of knowledge.

1.4 Document Organization

The remaining chapters are organized as follows. Chapter II provides background information on the uses and vulnerabilities of microcontrollers, RF signal collection, the AES algorithm, post-collection processing, the correlation operation, and operation environment verification. Chapter III details the methodology used for this research, including the collection and processing of UREs from a MCU, the CBAD process, the modifications to the AES algorithm, and the method of determining the operating environment of a particular DUT. Chapter IV presents the results of the process described in Chapter III, and the metrics used to evaluate CBAD process performance. Chapter V provides a summary of the research efforts and potential future research efforts.

Table 1. Relational mapping between RFINT Technical Areas in Previous related work and Current AFIT research contributions. The $\times$ symbol denotes specific areas addressed.

Technical Area	Previous Work		Current Research	
	Addressed	Ref #	Addressed	Ref #
TD Features	$\times$	[71, 72, 79–85, 87, 88, 94–96]	$\times$	[79, 80, 97, 98]
SD Features	$\times$	[14, 15, 74, 95]		
CD Features	$\times$	[81–85, 87, 88]	$\times$	[97, 98]

Emission Type				
Intentional (IRE)	$\times$	[25, 49, 50, 71, 74, 87, 88, 94, 95]		
Unintentional (URE)	$\times$	[13–15, 81–85, 96]	$\times$	[79, 80, 97, 98]
Burst	$\times$	[25, 49, 50, 71, 74, 87, 88, 94, 95]		
Continuous	$\times$	[13–15, 81–85, 96]	$\times$	[97, 98]
High SNR	$\times$	[25, 49, 50, 71, 74, 87, 88, 94, 95]		
Low SNR	$\times$	[13–15, 81–85, 96]	$\times$	[79, 80, 97, 98]

Classification/Verification Processes				
MDA/ML	$\times$	[13–15, 49, 50, 72, 74, 87, 88, 94–96]		
GRLVQI	$\times$	[49, 50, 72, 74, 81–85, 96]		
LFS	$\times$	[39–42]		
CBAD	$\times$	[81]	$\times$	[97, 98]

Dimensional Reduction Analysis (DRA)				
MDA/ML	$\times$	[39, 49, 50, 72, 74, 96]		
GRLVQI	$\times$	[72, 74, 81–85, 96]		
LFS	$\times$	[39–42]		

Verification				
Electronic Components	$\times$	[13–15, 85]		
Authorized Wireless Devices	$\times$	[25, 72, 74]		
Rogue Wireless Devices	$\times$	[25, 72, 74]		
Device Operations	$\times$	[81–84]	$\times$	[79, 80, 97, 98]

II. Background

This chapter provides the background information on the topics associated with this research. Section 2.1 provides a brief overview of microcontrollers and how they are used in military operations and Supervisory Control and Data Acquisition (SCADA) systems. Section 2.2 provides background on several attack vectors and specifies the attack used when creating the *Anomalous* program used in this research. Section 2.3 details the current methods of microcontroller malware and counterfeit detection and why they are inadequate. Section 2.4 provides an overview of the Advanced Encryption Standard (AES) algorithm, and details the portions relevant to this research. Section 2.5 provides background on the Correlation-Based Anomaly Detection (CBAD) process and how it is used to verify the operating condition of a microcontroller.

2.1 Microcontrollers

A Microcontroller Unit (MCU) is a specialized Integrated Circuit (IC) that is able to complete specific functions very efficiently and with less power than other ICs such as general microprocessors. Microprocessors can perform multiple tasks, but require external peripherals to be utilized. MCUs are made up of a Central Processing Unit (CPU), memory, a system clock, and programmable Input/Output (I/O) peripherals. Each of these components will be present, but can vary between each microcontroller depending on the application. Microcontrollers are versatile and can be programmed to do a number of tasks, from lighting an Light-Emitting Diode (LED) to a full encryption implementation. Programs are typically written in a higher-level language, like the C programming language, unless a particular application is very timing-sensitive [4]. In those cases, Assembly may be used.

2.2 Vulnerabilities

As more systems become interconnected, the threat of network penetration drastically increases. This is especially dangerous for aging and outdated Hardware (HW) and Software (SW) found in many Industrial Control Systems (ICSs). In the past, ICSs were typically self-contained making security a low priority, however modern ICS utilize Internet Protocol (IP) in order to enable remote administration, networked ICS components, and allow business systems access to ICS data [11]. While these systems are connecting to more external networks, they are also failing to implement basic access controls to the systems, such as differentiating between administrator and standard user accounts [93]. Once an attacker is able to gain access to the network, it is possible to send malicious firmware updates to the microcontroller-based systems on the network, as was demonstrated in an attack in 2011 that successfully manipulated the microcontroller running a laptop’s battery, sending false messages about the battery status to the computer [92].

Another attack against an IC was demonstrated in 2015 and is the basis of the *Anomalous* code used in this research [89]. The attack successfully weakened an AES implementation programmed onto a Field Programmable Gate Array (FPGA). By replacing the contents of the *S-box* with the identity function, the output ciphertext of the AES algorithm became much more susceptible to cryptanalysis since the attack turns both the encryption and decryption functions into a linear bijection. This means if an attacker is able to collect one plaintext/modified ciphertext pair, he can compute the weakened key. Once that weakened key is computed, the attacker is then able to decipher any subsequent ciphertext generated by the modified algorithm. To better understand the attack, a brief overview of the AES algorithm is provided in Section 2.4.

2.3 Current Malware Prevention/Detection Methods

The most common method of determining whether a microcontroller is a legitimate device or a counterfeit is functional testing. Functional testing can accurately detect when functionality of a device is lacking or has been removed, but it will likely not detect any functionality added to the HW or SW for a device by an adversary, such as a HW trojan [6].

Modern microcontrollers typically come equipped with several security measures to prevent the modification of memory, thereby preventing an attacker from maliciously tampering with the device [59]. However, older models of MCUs are often found in aging ICSs and often do not have these additional security measures, so they remain vulnerable to the malicious firmware updates described previously. Instead, they rely on traditional Intrusion Detection and Prevention Systems (IDPSs) and firewalls that work at the upper bit-level layers of the Open Systems Interconnect (OSI) model (layers 2-7) to remain secure; however attackers are always finding new sophisticated methods of bypassing these security measures [21]. One such method involves gaining access to the Human-Machine Interface (HMI) used to provide SCADA controls, which are often found outside the ICS De-Militarized Zone (DMZ), any firewalls, IDPS, and other security monitoring devices. In order to grant the HMI control of the SCADA systems, the security features must be configured to allow traffic from the HMI to enter the network, thus creating a vulnerability if certain precautions are not taken at the HMI system [51]. An alternative to bit-level credentials uses Radio Frequency (RF) emissions at the Physical Layer from a device to discriminate between devices, as well as verifying operating conditions. This method has proven successful in a large body of research [14, 15, 18–20, 22, 23, 25, 27, 30, 38, 70–74, 79–81, 95, 97, 98].

2.4 Advanced Encryption Standard Algorithm

This section briefly discusses the aspect of the AES algorithm relevant to this research according to the National Institute of Standards and Technology (NIST) Federal Information Processing Standards Publication 197 (FIPS-197) [64], and why it was selected for this research.

2.4.1 Motivation To Use AES.

In 2001, the Rijndael cipher, now called AES, was established as the official encryption standard of the United States (US) government by NIST [9]. It is used in military systems, such as smart cards and communications, and can be implemented in ICs such as FPGAs and MCUs [34,35]. Since then, several research efforts have been performed exploiting unintentional emissions and attacking the AES algorithm using Side Channel Analysis (SCA) [36, 53, 55, 89]. The prevalence of the AES algorithm as opposed to other less used algorithms such as Data Encryption Standard (DES) and Triple Data Encryption Standard (3DES), and the documented attacks against the algorithm make AES an ideal program candidate for this research.

2.4.2 AES Primer.

AES is a block encryption algorithm, meaning all inputs to each round are 128 bits in length. The algorithm accepts a 128-bit plaintext block and a key block that varies in size (either 128, 192, or 256 bits) as inputs. A key schedule is applied to the input key, where several subkeys are generated, the number of which depends on the size of the input key. The size of the input also dictates the number of AES rounds the message must pass through. Regardless of the input key size, all subkeys generated are 128 bits in length. The AES implementation used in this research is AES-128, meaning the input key length is 128 bits. After the algorithm receives the

input bits, both the plaintext and subkey blocks are arranged in a 4 byte by 4 byte array, as shown in Figure 3.

input bytes

in_0	in_4	in_8	in_{12}
in_1	in_5	in_9	in_{13}
in_2	in_6	in_{10}	in_{14}
in_3	in_7	in_{11}	in_{15}

Figure 3. 128 input bits shown as an array of 16 bytes [64].

The plaintext array is then copied to the State, and all subsequent cipher operations are performed on the State. Once the State has passed through all operations in the AES algorithm, it is copied to the output and is then called *ciphertext*. Figure 4 shows the general form of AES encryption. AES-128 consists of $N_r = 10$ rounds. The first nine rounds are identical and consist of the following four operations:

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

The final round is similar to the previous nine, but the MixColumns operation is omitted. The attack used in this research only involves the SubBytes operation; therefore, the others are not discussed here. The SubBytes round applies a non-linear, byte-wise substitution of the State bytes with values found in a table called the *S-box*. The *S-box* values are calculated using multiplication in the Galois Field ($GF(2^8)$) and are discussed in detail in Section 5.1.1 of the FIPS-197 documentation. The values

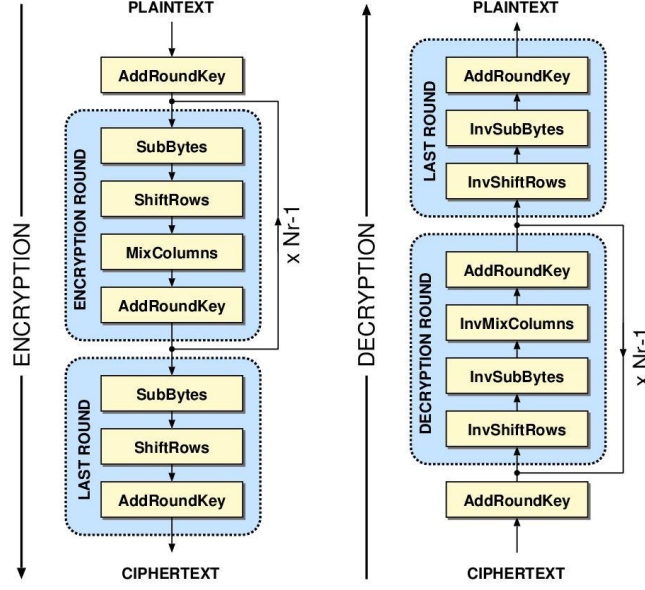


Figure 4. General form of AES encryption (left) and decryption (right) [36].

of the *S-box* are typically pre-calculated and stored in memory on the device running the encryption in order to save computational time and resources.

The attack discussed in Section 2.2 maps the bytes found in the S-box to the identity function (Eq. 1). To the user, the plaintext will still be encrypted but will be far more susceptible to cryptanalysis. For Swierczynski and his team to modify the values of the *S-box*, it was necessary for them to locate the logic unit on the FPGA which contained it and manipulate the bit stream. To implement the same function in the microcontroller used in this research, a simple code modification was all that was required.

$$\text{S-box}_{\text{AES}}^{id}(i) = i, \forall i \in \text{GF}(2^8) \quad (1)$$

2.5 RF Signal Collection

Emission collection for this research closely mimics the methods used in previous Air Force Institute of Technology (AFIT) work [79, 81, 96]. The signals of interest

are Unintentional Radio Frequency Emissions (UREs) which is essentially Electromagnetic (EM) leakage of a device during operation. As such, these emissions are not well-structured and are emitted with very low average powers when compared to Intentional Radio Frequency Emissions (IREs). To collect emissions with such low power, a highly sensitive near-field EM probe is placed in close proximity to the microcontroller, and determining the chip location is largely based on visual cues such as waveform shape, which is further discussed in Section 3.3. Using a near-field EM probe allows for a non-destructive and non-contact method of capturing collections, as opposed to other methods which require a physical connection or additional circuitry added to the Device Under Test (DUT) [52], or “decapping” the device to expose the die as was performed in [86].

2.5.1 Correlation.

Correlation is a key function in several applications such as implementing a matched-filter for estimating digital symbols [78] and identifying signals when noise may be an issue [24]. Correlation is also less resource-intensive than other classification algorithms such as Multiple Discriminant Analysis/Maximum Likelihood (MDA/ML) and Generalized Relevance Learning Vector Quantized-Improved (GRLVQI) due to its well-defined complexity, relative simplicity, and predictable computational complexity.

Correlation is the fundamental operation of the CBAD process, and is used to provide a metric of similarity between two signals in this research, namely the pre-established reference signal and an unknown collected signal. The CBAD process uses the autocorrelation of the reference signal and the cross-correlation of the reference signal with the unknown signal to determine signal similarity. Given an example discrete complex-valued reference sequence $x[n]$ and an example discrete complex-

valued unknown sequence $y[n]$, the autocorrelation ($R_{XX}[k]$) and cross-correlation $R_{XY}[k]$ sequences at lag k are given by

$$R_{XX}[k] = \sum_n x[n]x^*[n - k], \quad (2)$$

$$R_{XY}[k] = \sum_n x[n]y^*[n - k], \quad (3)$$

respectively, where $*$ denotes the complex conjugate.

2.5.2 Operation Verification.

Operation verification is a one-to-one comparison between an unknown operation and a pre-established reference operation. The unknown operation is declared as either *Normal* or *Anomalous* correctly or incorrectly based on a previously determined statistic threshold. The statistic threshold and reference operation are generated using a training model based on a “known good” baseline configuration.

To provide a visual representation of how accurately the CBAD process can detect the current operating condition of the DUT, Receiver Operating Characteristic (ROC) curves are presented since ROC curves are often used for binary classification in fields such as biometric systems [46]. The ROC curves presented in this research are plots of False Anomaly Detection Rate (FADR) versus True Anomaly Detection Rate (TADR) as the threshold is changed. The terms FADR and TADR are specific to this research and can differ in other bodies of research depending on the metrics used. To generate the ROC curves, the lowest-valued statistic from the is set as the lowest possible threshold, t_{low} , and the highest-valued statistic is set as the highest possible threshold, t_{high} . The threshold $\forall t_{low} \leq t \leq t_{high}$ is varied, and the values of FADR versus TADR are plotted for each value of t . In addition to the ROC curves, SNR_A versus TADR plots are presented to show CBAD performance as the SNR_A

varies. In this research SNR_A is varied by combining a collected signal with power-scaled Additive White Gaussian Noise (AWGN). AWGN is often used in digital communication simulations to model channel noise [77] and is therefore used in this research.

III. Methodology

This chapter details the approach taken to perform the research and generate the results captured in Chapter IV. The goal is to declare device operations as *Normal* or *Anomalous* by comparing a collected unknown signal with a pre-established reference signal. The reference signal is generated by training on a collection of signals collected from the device while it is performing *Normal* operations. The process of declaring the current operating condition of a particular microcontroller device as either *Normal* or *Anomalous* operation can be divided into three major steps outlined in Figure 5: Emission collection, post-collection processing, and Correlation-Based Anomaly Detection (CBAD) statistic generation/declaration. The processes used in each of these steps are based on existing Air Force Institute of Technology (AFIT) work on anomaly detection [81].

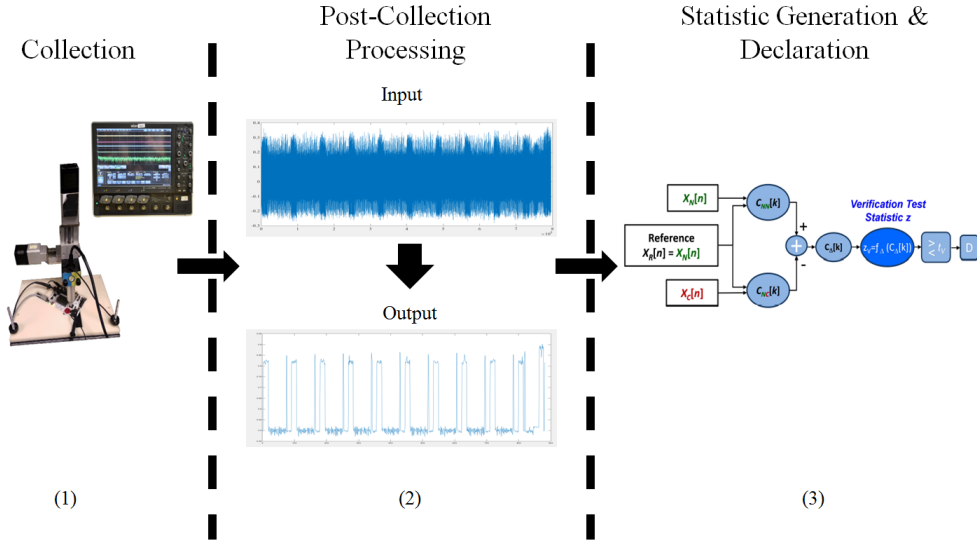


Figure 5. Overall process to determine the operating condition of a microcontroller based on previous research [81]. Three main steps are performed: (1) Emission Collection, (2) Post-collection Processing in MATLAB[®] and (3) CBAD statistic generation and declaration.

Section 3.1 discusses the Device Under Test (DUT) used in this research and the reasons for selecting it. Section 3.2 discusses the operating conditions the DUTs op-

erated in when performing the research. Section 3.3 details the Unintentional Radio Frequency Emission (URE) collection process, including collection setup, probe placement, and sampling/triggering. Section 3.4 describes the post-collection processing stage, where the signal is transformed using filtering, adding independent Additive White Gaussian Noise (AWGN) realizations, down-conversion, applying the Hilbert transform, jitter removal, and non-uniform decimation. Section 3.5 describes the CBAD algorithm, how statistics are generated from the algorithm, and how a declaration is made based off of a comparison between the threshold value, t_v , and the generated statistic z_v .

3.1 Microcontroller Device Description

All emissions were collected from $N_D = 5$ Commercial Off The Shelf (COTS) Texas Instruments (TI) MSP430F5529 Microcontroller Unit (MCU) devices. One of the DUTs is shown affixed to an experimenter’s board and indicated by a red square in Figure 6. Each MCU was selected from one of two manufacturing lots and assigned a color and number to visually differentiate between them. The number and color mappings for each device are shown in Table 2. The intention of collecting emissions from devices with the same make/model but different manufacturing lots was to determine whether a reference signal generated from one device and/or lot could be used to detect anomalies in another device and/or lot, and to ensure the process applies across the two model lots considered.

Table 2. DUT assigned ID values and designated lot numbers/colors

DUT ID	Lot Designation	Color Designation
DUT3	Lot 1	Blue
DUT4	Lot 1	Blue
DUT5	Lot 2	Red
DUT6	Lot 2	Red
DUT11	Lot 2	Red

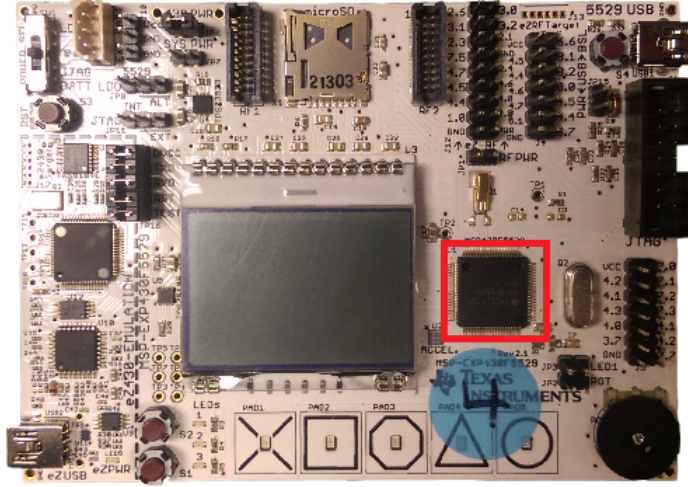


Figure 6. TI MSP430F5529 (indicated by red box) affixed to an experimenter board. The identification label in this case indicates DUT4 from the blue lot

The MSP430 series MCU was selected due to its popularity having sold hundreds of millions of units and its versatility in a multitude of applications, making it representative of MCUs implemented in embedded designs [31, 90]. The MSP430 also represents a 16-bit architecture, as opposed to the 8-bit devices in other related studies [13, 61, 62].

Each DUT is programmed with an Advanced Encryption Standard (AES) implementation provided by TI using the C programming language [91]. Two variations of this code were used when programming the MCUs to model *Normal* and *Anomalous* programs, and are further described in 3.2.

3.2 Microcontroller Operating Conditions

As stated in Section 3.1, two variations of the AES C code implementation provided by TI were used when programming the MCU devices: *Normal* operation and *Anomalous* operation. The program was modified to add a trigger to the code, which is discussed in Section 3.3.3.

In addition to the trigger, code was added to illuminate a Light-Emitting Diode (LED) if the ciphertext output of the MCU deviated from the expected ciphertext based on proper AES implementation, indicating *Anomalous* behavior. The final modification that both programs received was code added to illuminate a second LED that would toggle every $N_L = 500$ AES loops to allow a means of confirming the program was continuously running on the MCU. This provided a visual cue to easily ensure the device is operating while collections are performed.

When extracting the Region of Interest (ROI) from a collected signal, only samples within the rising and falling trigger edges are kept. The trigger is toggled immediately before and immediately after the AES algorithm. Therefore, the Radio Frequency (RF) waveforms emitted by the code implementing the LED functions are removed during this process since the LEDs activate after the AES loop completes. All code used when programming the DUTs can be found in Appendix B.

3.2.1 *Normal* Operation.

The *Normal* program implements the AES algorithm as provided by TI and is described in detail in Section 2.4. The only modifications to the program provided by TI are the addition of the trigger and LED code mentioned in Section 3.2

3.2.2 *Anomalous* Operation.

The *Anomalous* operation is based on a verified attack vector demonstrated by Swierczynski [89]. If an attacker is able to modify the contents of the S-box described in Section 2.4 to the identity mapping (Eq 1), the ciphertext becomes extremely vulnerable to cryptanalysis attacks, and the secret key of the modified AES algorithm can be determined. Even with the modification, the algorithm encrypts the plaintext, and the resulting ciphertext can be decrypted with the new key created by the modification. An adversary would be capable of calculating the key with one plaintext/modified ciphertext pair.

3.3 Radio Frequency Emission Collection

This section details the collection process used in this research. For each DUT, $N_{tst} = 1000$ testing signals were collected for both the *Normal* and *Anomalous* operating conditions. To create a reference signal, an additional $N_{trng} = 500$ training signals were collected from both red lot device DUT11 and blue lot device DUT3 to provide a reference signal generated from both manufacturing lots. Training signals are used to create a reference signal that is subsequently used on the remaining testing signals (not previously been seen by the CBAD process) to verify the performance of both the reference signal selection process, and the CBAD process. No training signal is used when generating final results and similarly, no testing signal is used to generate a reference signals or thresholds.

3.3.1 Microcontroller Collection Configuration.

Each DUT is placed inside a custom-made mount shown in Figure 7, which restricts movement of the MCU during the collection processes in an effort to mitigate effects movement may have on collections. The mount is affixed to an automated XYZ table, which holds a Riscure near-field RF probe with baseband bandwidth $W_{BB} = 500$ MHz. Placement of the probe using the XYZ table is controlled by a MATLAB[®] script. The near-field probe and probes attached to the trigger and clock Input/Output (I/O) pins are connected to a Lecroy Wavepro 760Zi-A Oscilloscope. The oscilloscope samples the incoming collections at a rate of $f_s = 250$ MSps. In order to prevent aliasing of frequency content of $f \geq 125$ MHz, a Low Pass Filter (LPF) with -3 dB cutoff frequency of $f_{CO} = 96.83$ MHz is placed in-line with the near-field RF probe and the oscilloscope. The impulse response for the anti-aliasing filter is shown in Figure 8.

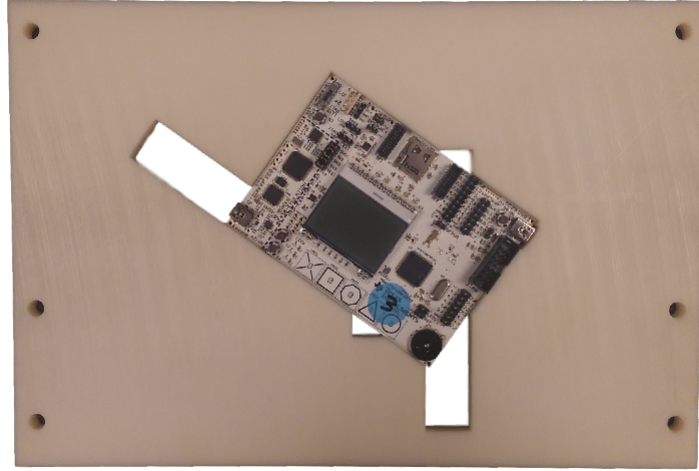


Figure 7. Custom-made mount that houses the DUT during the collection process

3.3.2 RF Probe Placement.

This section describes the process of selecting the chip location of the near-field RF probe. The UREs emitted from a MCU can vary substantially with minute

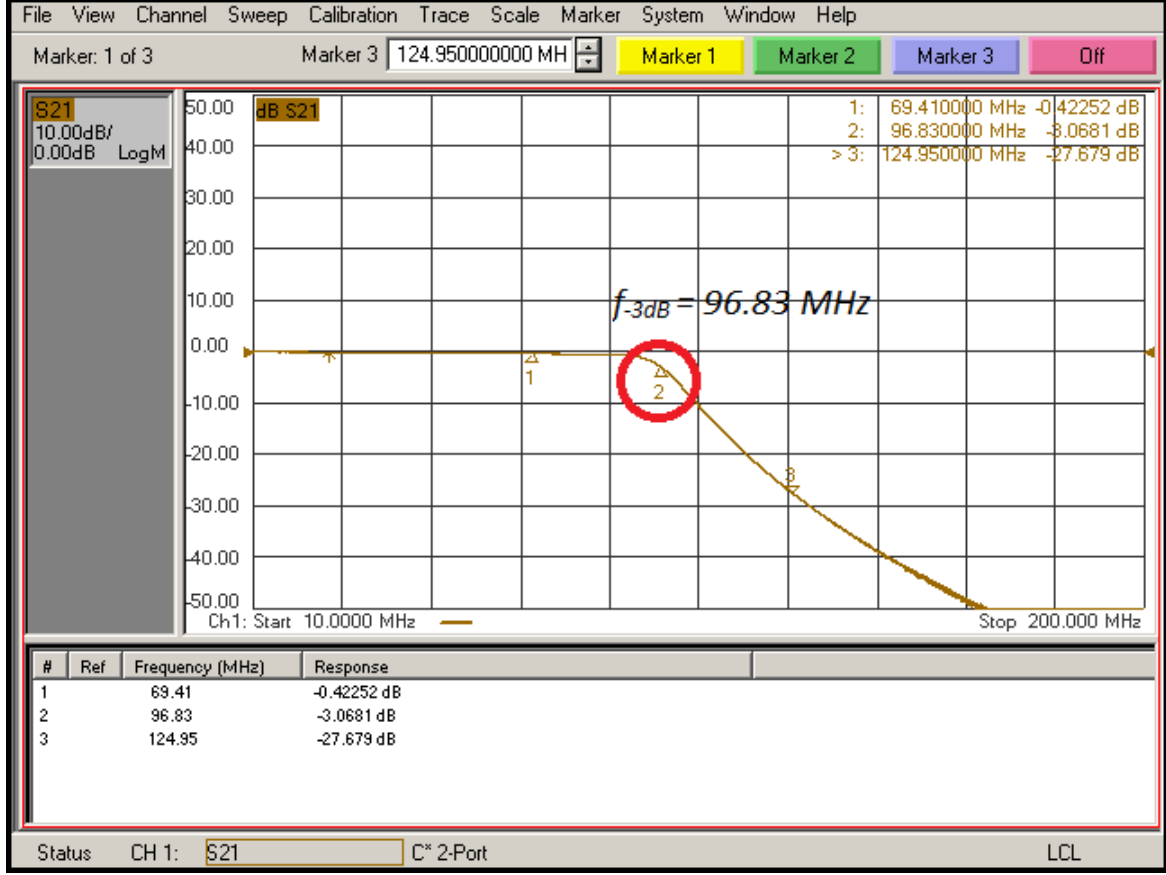


Figure 8. Impulse response of the $f_{CO} = 96.83$ MHz LPF. Frequencies $f \geq 124.95$ MHz are shown to be attenuated by -27.879 dB.

movements of the RF near-field probe. Therefore, it was necessary to determine a collection location that provided a relatively high power at the frequency of interest, and to develop a method to repeatedly place the probe in the same location on each device after determining the probe location to be used.

When scanning over the surface of the DUT with the near-field probe, a set of $N_p = 10$ peaks are observed at multiple locations on the surface of the device, as shown in Figure 9. These peaks correspond to the number of AES rounds described in Section 2.4, as well as the LED programming. The X and Y coordinates of the locations where the peaks were clearly visible were recorded. Once all locations where the peaks were clearly visible were recorded, a Time Domain (TD) signal $x[n]$ with a

total of N_x samples was collected at each.

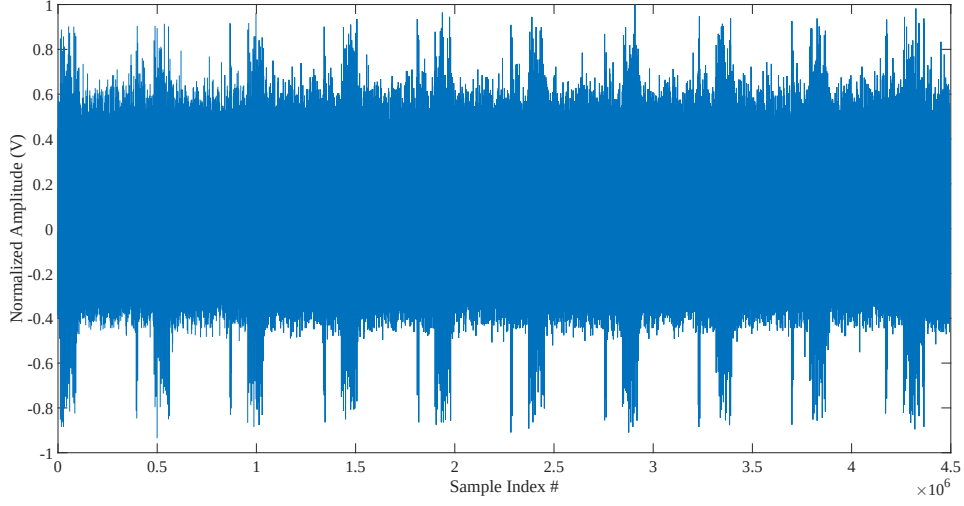


Figure 9. Single TD *Normal* signal. $N_p = 10$ bursts are observed, indicating the number of rounds in the AES algorithm.

The Power Spectral Density (PSD) at each location was calculated from the Discrete Fourier Transform (DFT) using

$$\text{PSD}(x[n]) = |X[n]|^2 \quad (4)$$

where $X[n]$ is the DFT of $x[n]$ and is calculated by

$$X[n] = \frac{1}{N_x} \sum_{k=1}^{N_x} x[n] e^{-j\Phi(N_x, k, m)} : 1 \leq m \leq N_x, \quad (5)$$

where

$$\Phi(N_x, k, m) = \left(\frac{2\pi}{N_x} \right) (k-1)(m-1) : 1 \leq m \leq N_x \quad (6)$$

The location that provided the highest peak power at $f_h = 57.28$ MHz was selected as the position used when performing the remaining collections for every DUT. The frequency $f_h = 57.28$ MHz was chosen as the frequency component of interest when

determining the near-field probe location on the surface of the chip after investigating the mean normalized PSD of $N_{PSD} = 50$ collection PSDs shown in Figure 10. The top ten peak frequencies were used as center frequencies f_c with several W_{BP} values at each f_c of the filter to be used in the post-collection processing described in Section 3.4.1. The subsequent filtered signal was examined after using each of the ten peak frequencies as f_c , and $f_c = 57.28$ MHz was the peak frequency examined that best maintained the original signal's general shape.

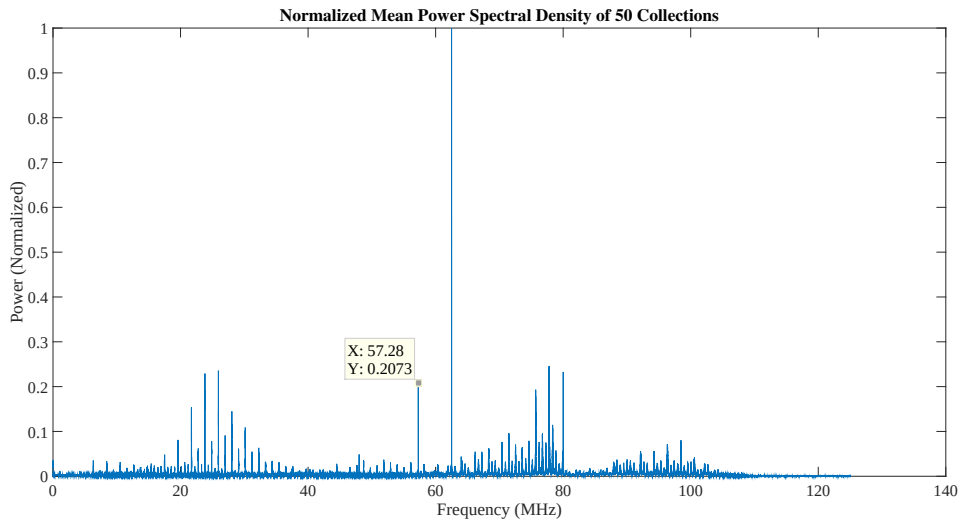


Figure 10. Normalized PSD of the signal of interest

3.3.3 Sampling and Triggering.

Signals were collected with the LeCroy oscilloscope specified in Section 3.3, which samples the collected signals, $x_c[n]$, at a rate of $f_s = 250$ MSps with a bit depth of 8 bits. Each collection captures a signal that is $t_c = 20$ milliseconds, making each signal $N_{CS} = f_s \cdot t_c = 5$ million samples in length immediately after collection. To minimize aliasing caused by frequencies exceeding the Nyquist limit of $f \geq 125$ MHz, a LPF with a cutoff frequency $f_{co} = 81$ MHz is placed in-line with the probe and oscilloscope, such that any frequency components greater than $f_{-3dB} = 98.63$ MHz

are attenuated by at least 3 dB, as shown in Figure 8.

In order to synchronize the oscilloscope with the AES routine, a trigger, $x_{trig}[n]$, was added to the program, and is shown encapsulating the signal of interest in Figure 11. The trigger was set to go high just before entering the AES loop, and low just after it left the AES loop. All samples that were not between the rising and falling edge of the trigger pulse were discarded during the post-collection processing step. The same trigger signal, $x_{trig}[n]$, was used to determine which samples in each signal were discarded to ensure the signals remained the same length.

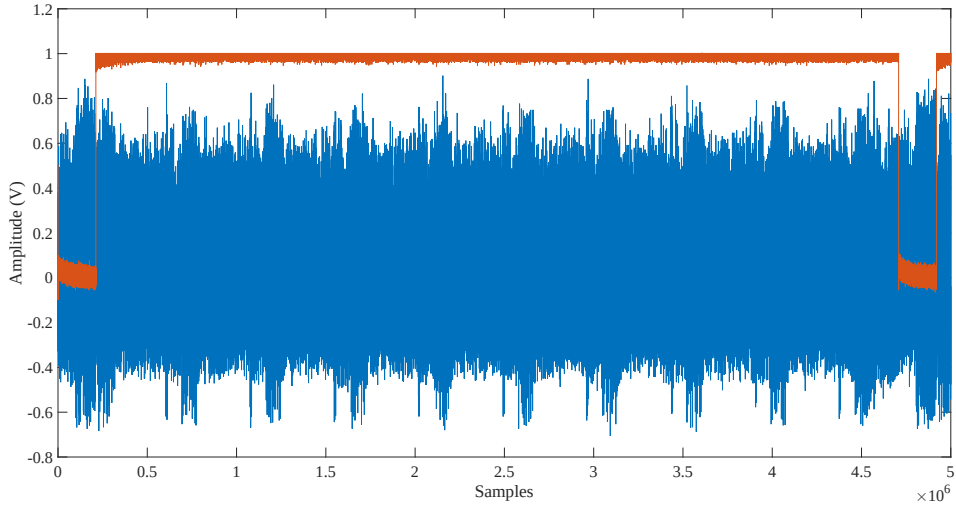


Figure 11. Trigger waveform (orange) shown encapsulating the signal of interest. Any samples that fall outside of the rising and falling edge of the trigger waveform are discarded, including the additional LED code.

3.4 Post-Collection Processing

This section describes the post-collection processing of the collected signals $x_c[n]$. This step is performed to generate a transformed waveform $x_t[n]$ to be used as an input to the CBAD algorithm that requires less processing power than the original collected signal, while still achieving the desired benchmarks. As stated in Section 3.3.3, a total of N_{RS} samples that fall outside the rising edges of $x_{trig}[n]$ are removed from each

signal, creating a signal $x_f[n]$ that is $N_{samps} = N_{CS} - N_{RS}$ samples in length.

The post-collection processing is performed with five steps, which are shown in Figure 12 and described in following sections.

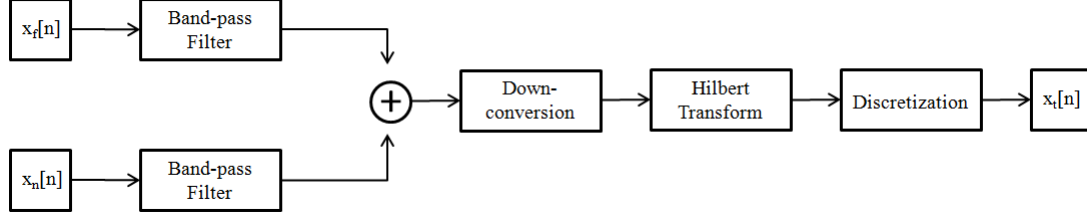


Figure 12. Post-collection processing block diagram. The collected signal $x_f[n]$ and power-scaled noise $x_n[n]$ are like-filtered and added together to achieve the desired SNR_A . Next the signal is down-converted, the Hilbert transform is applied to it, and finally the signal is discretized. The resulting signal $x_t[n]$ is then used as the input to the CBAD algorithm.

3.4.1 Filtering.

After $x_f[n]$ was normalized, it was digitally filtered using a $N_{ord} = 16$ Butterworth Band-Pass Filter (BPF), with a center frequency $f_{BP} = 57.28$ MHz and a bandwidth $W_{BP} = 7$ MHz. The frequency $f_h = 57.28$ MHz was selected as the frequency of interest using the criteria and process provided in Section 3.3.2. To establish W_{BP} , filters with $W_{BP} = [1, 3, 5, 7, 9, 11]$ MHz were applied separately, and $W_{BP} = 7$ MHz provided the best results out of the considered frequency bandwidths. The frequency response of the digital filter is shown in Figure 13, and the filtered signal is shown in Figure 14.

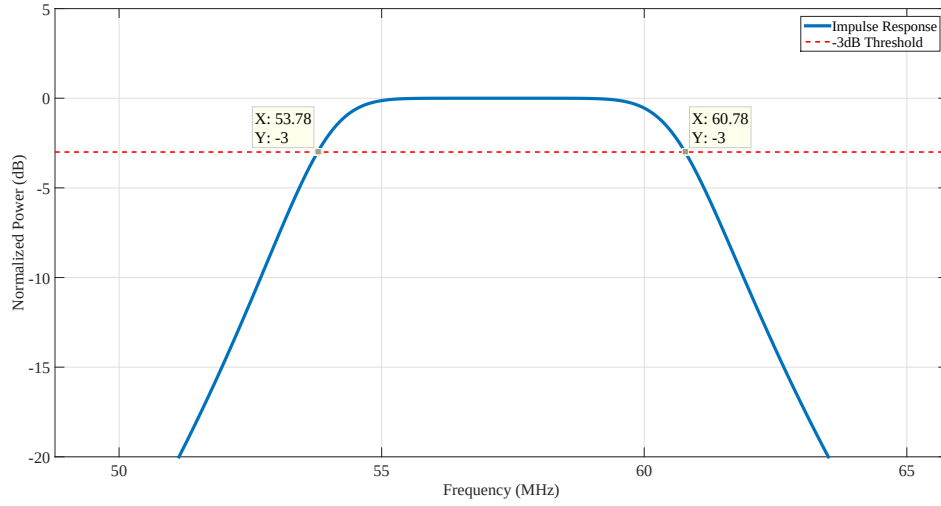


Figure 13. Impulse Frequency Response of an 8th-order Butterworth Band Pass Filter (BPF) with a center frequency $f_{BP} = 57.28$ MHz and a -3.0 dB bandwidth of $W_{BP} = 7.0$ MHz.

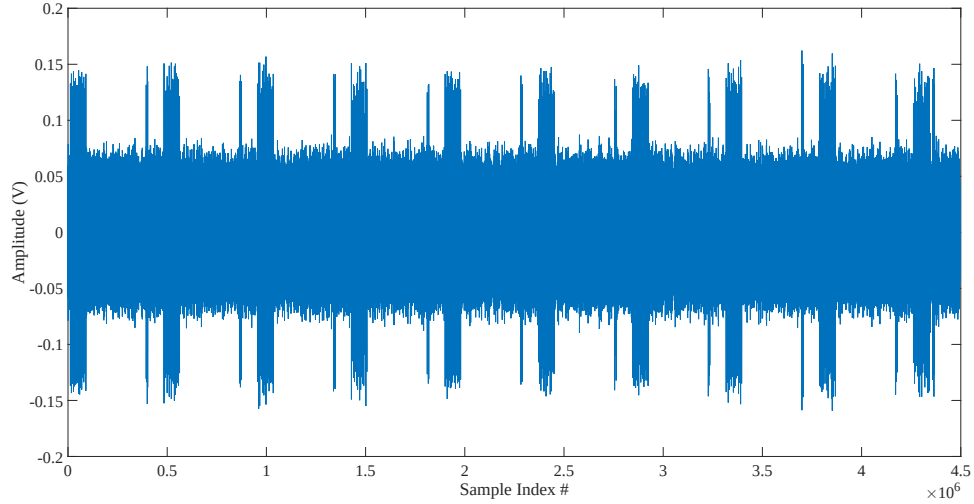


Figure 14. Single *Normal* signal after applying the filter described in Section 3.4.1

3.4.2 Signal-to-Noise Ratio Scaling.

In practice, the collected Signal-to-Noise Ratio (SNR), SNR_c , may be degraded due to environmental factors. A signal's SNR is calculated by taking the ratio of the signal power (P_S) and the noise power (P_N) in decibels (dB). In order to simulate

degraded environmental conditions and evaluate the robustness of the implemented process without taking additional collections at varying levels of probe height, like-filtered AWGN, $x_n[n]$ was added to $x_f[n]$ to generate the desired analysis SNR (SNR_A) and noisy signal $x_{noisy}[m]$. When calculating the scale factor for the noise realizations, it was assumed that the collected signal power is infinite, meaning it was assumed that the collected signal was noiseless and that all emissions contributed to the signal power. This was done because when considering UREs, there is no way to clearly distinguish between the signal of interest and noise. For this research, $\text{SNR}_A \in [-40, 40]$ dB at varying increments for a total of $N_{SNR} = 19$ SNR_A values, with $N_{Nz} = 5$ independent noise realizations at each SNR_A . Initially, the SNR_A step size was in increments of $\text{SNR}_\Delta = 10$ dB. However, the results showed that finer granularity was necessary for $-25\text{dB} \leq \text{SNR}_A \leq 15\text{dB}$, so for $-25\text{dB} \leq \text{SNR}_A \leq 15\text{dB}$ an incremental size of $\text{SNR}_\Delta = 5$ dB was used, and was further decreased for $-15\text{dB} \leq \text{SNR}_A \leq -1\text{dB}$ to $\text{SNR}_\Delta = 2$ dB. When calculating the value of SNR_A , it was assumed that the collected signal $x_c[n]$ is a noiseless signal at infinite SNR (i.e., $P_N = 0$).

The $N_{Nz} = 5$ noise realizations generated for each signal are concatenated with each other, for a total testing matrix size of $N_{tst} \times (N_{Nz} \cdot N_{samps}) \times N_{SNR}$, and total training matrix size of $N_{trng} \times (N_{Nz} \cdot N_{samps}) \times N_{SNR}$.

3.4.3 Down-conversion.

Once the noisy signal $x_{noisy}[n]$ was generated, it was multiplied by a sinusoid with Intermediate Frequency (IF) $f_{IF} = 61.28$ MHz in order to down-convert the signal to a near-baseband center frequency $f_{DC} = 4.0$ MHz. The down-converted signal $x_{DC}[n]$ was then filtered using a LPF Butterworth filter, with a cutoff frequency of $f_{CO} = 7.5$ MHz. The impulse response for the digital LPF is shown in Figure 15, and the down-converted signal is shown in Figure 16.

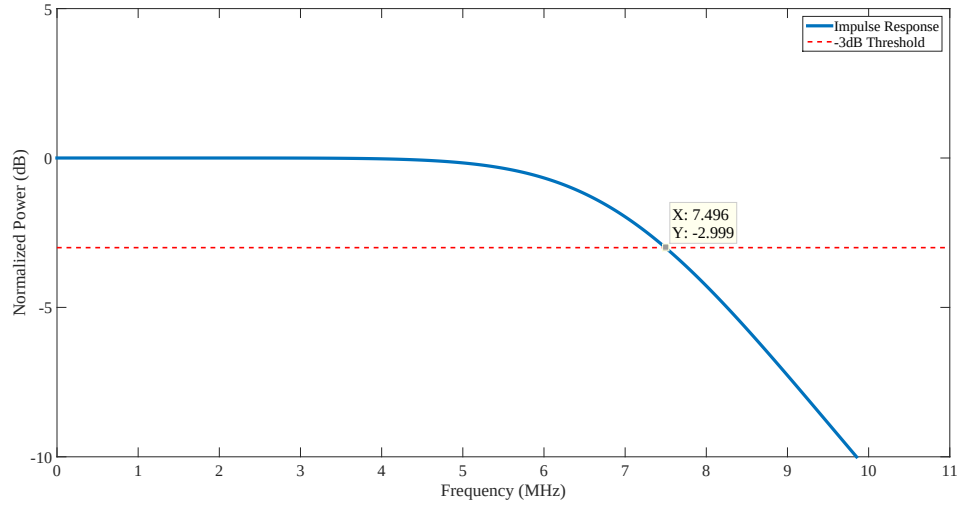


Figure 15. Impulse Frequency Response of an 8th-order Butterworth LPF with cutoff frequency $f_{CO} = 7.5$ MHz

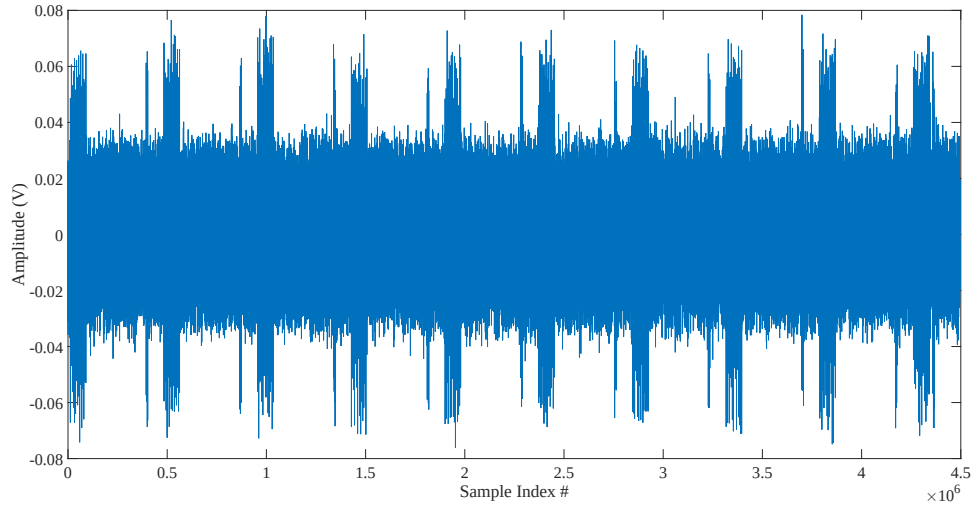


Figure 16. Single Time Domain *Normal* signal after down-conversion and baseband filtering

3.4.4 Hilbert Transform.

Once the signal is down-converted, a Hilbert transform is applied using the MATLAB[®] `hilbert` function. The Hilbert transform was used in both previous AFIT work and speech recognition applications, provides an amplitude estimate, and was shown to

improve anomaly detection over taking just the absolute value of the signal [37,69,81].

The MATLAB[®] `hilbert` function generates a transformed discrete sequence $H[x_{DC}[n]]$ for the real-valued sequence $x_{DC}[n]$ and returns a representation of the signal having In-phase (I) and Quadrature (Q) components. To obtain the instantaneous amplitude of $x_{DC}[n]$, the magnitude response $x_H[n] = |H[x_{DC}[n]]|$ is calculated using MATLAB's[®] `abs` command. The magnitude of the Hilbert-transformed signal is shown in Figure 17.

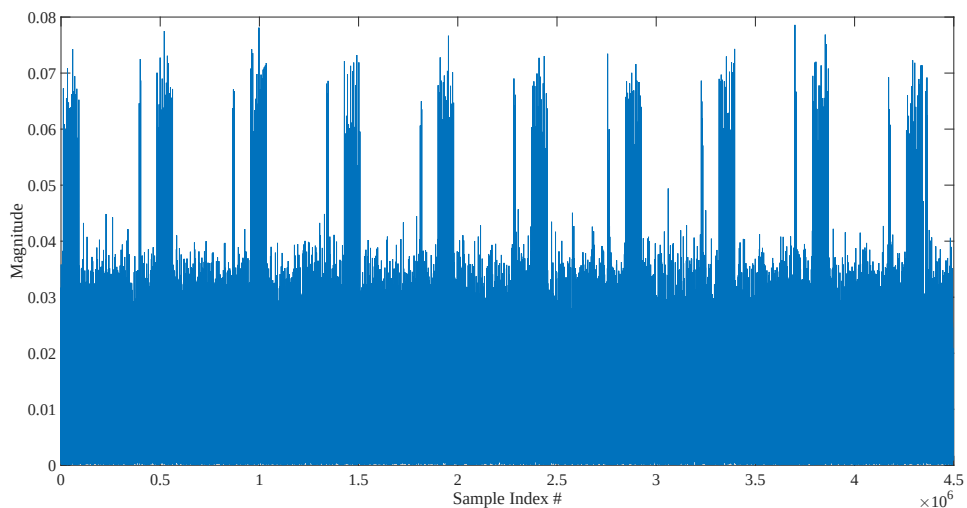


Figure 17. Single Time Domain *Normal* signal after applying the Hilbert-Transform and calculating the magnitude of the resulting signal

3.4.5 Jitter Removal.

Jitter is the deviation of a clock edge from its specification value. The MSP430F5529 has a published clock frequency of $f_{clk} = 1.049$ MHz [90], however experimental results show an observed clock frequency $f_{clk} \approx 1.0567$ MHz that varies between [1.0081 1.1062] MHz.

Every DUT suffered from a substantial amount of jitter, up to $N_j = 13$ samples difference between clock pulses, which caused issues with signal alignment and number of clock pulses within the trigger. As discussed in [67], manufacturers of Integrated

Circuits (ICs) will sometimes intentionally vary a clock frequency by a small amount at a low-frequency rate, thereby inducing jitter. This practice spreads the clock energy in the frequency domain, and reduces the amplitude of the clock harmonic emission at any individual frequency.

To overcome the alignment issue, each signal in $x_H[n]$ had $N_k = 20$ samples kept after each rising and falling edge of the clock signal, while the remaining samples were discarded. Values of $N_k = \{10, 18, 19, 20, 21, 22, 30, 40, 50, 60, 70, 80, 90, 100\}$ were used when determining the number of samples to keep, and $N_k = 20$ samples was the lowest value that maintained the overall shape of the signal. Because the number of clock pulses within the trigger varied the signal sizes also varied after the jitter removal process. To overcome this issue all signals were shortened to the smallest sample length signal by removing the last samples.

3.4.6 Non-Uniform Decimation.

The final step of the post-collection processing stage generates the fully transformed signal, $x_t[n]$, using non-uniform decimation, and is similar to a peak detector in analog signals combined with a sampling process. This process reduces the total number of samples in the signal while maintaining the amplitude envelope of the signal, thus making computation less intensive when the subsequent signal $x_t[n]$ is used as the input to the CBAD algorithm. The decimation process takes $N_d = 900$ consecutive samples of $x_H[n]$, selects the maximum value of the $N_d = 900$ samples, and discards the remaining samples. The resulting waveform $x_t[n]$ is an envelope of the Hilbert-transformed waveform $x_H[n]$ with a signal length of N_{samps}/N_d .

The final step in the post-collection processing step is to normalize $x_t[n]$ by amplitude. Anomaly detection performance was improved by performing a final normalization before input to the CBAD algorithm. The final transformed signal is shown in Figure 18.

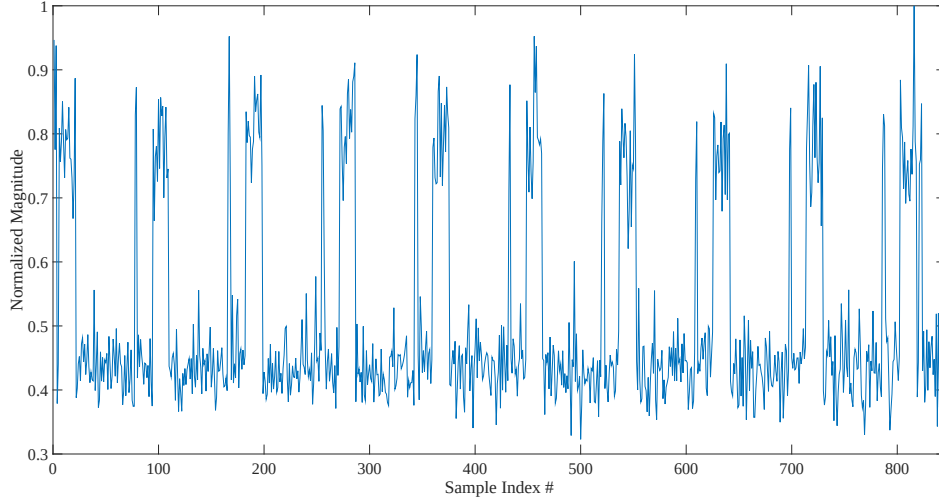


Figure 18. Single Time Domain *Normal* signal after non-uniform decimation, the final stage of the post-collection processing

3.5 Correlation-Based Anomaly Detection Algorithm

This section describes the overall process of generating a statistic, creating a reference signal, determining the statistic threshold, and the operating condition declaration.

3.5.1 Statistic Generation.

Figure 19 shows the overarching process of creating a statistic based on two signal inputs, the reference signal $x_R[n]$ and the transformed collected signal $x_t[n]$. First, the MATLAB's[®] `xcorr` function is applied to create an autocorrelation sequence $C_{RR}[k]$ and a cross-correlation sequence $C_{RT}[k]$.

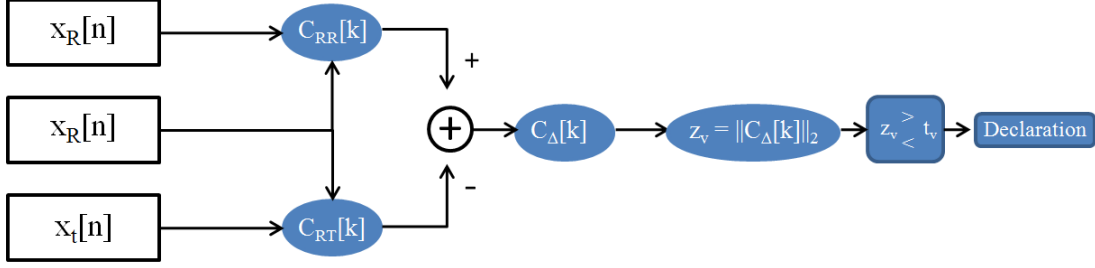


Figure 19. The CBAD statistic generation and declaration process. Statistics are generated by calculating the 2-norm of the difference between two correlation vectors. The statistic is compared to a threshold value, t_v , and a declaration is made [81].

Next, $C_{RT}[k]$ is subtracted from $C_{RR}[k]$ in order to create a single difference vector $C_{\Delta}[k]$. Note that if the transformed signal $x_t[n]$ was identical to the reference signal $x_R[n]$ the resulting vector would be a zero vector since $C_{RR}[k] = C_{RT}[k]$.

Lastly, the MATLAB[®] `norm` function is applied to $C_{\Delta}[k]$ to generate the final statistic. The `norm` function returns the 2-norm of the difference vector, and is calculated according to [57]:

$$\|C_{\Delta}[k]\|_2 = \sqrt{\lambda_{\max}(C_{\Delta}[k]^H C_{\Delta}[k])} , \quad (7)$$

where $\lambda_{\max}(C_{\Delta}[k]^H C_{\Delta}[k])$ is the maximum eigenvalue of the matrix created by multiplying the hermitian transpose of a vector and the original vector together.

The signals $x_t[n]$ used in this research were too large to provide the desired results as-is. To address this issue, the input signal was divided into $N_r = 33$ regions, and separate statistics were created for each region. The statistics were stored in a vector until the statistic for every region had been calculated, at which time the 2-norm of the statistic vector was calculated. The resulting value was used as the final statistic in determining the operating condition of the DUT.

3.5.2 Reference Signal Generation.

A reference signal is generated for each of the $N_r = 33$ regions described above and for each SNR_A value. A training matrix of $N_{trng} = 500$ *Normal* signals is used when determining the reference signal. These signals are only used for generating the reference signal and threshold values and are not used as testing signals when obtaining results. In order to account for potential collection bias, the $N_{trng} = 500$ *Normal* signals are selected from a larger matrix consisting of $N_{tot} = 1500$ signals in an interleaving pattern (i.e., select signal 1,4,7,10,...,1500). The mean of the training matrix is appended, for a total number of training signals $N_{trng} = 501$ at each SNR_A .

To select the most effective reference signal, each of the training signals is separately used as the reference signal against every other training set signal in the CBAD statistic generation process. Therefore, each signal generates $N_{tstats} = 500$ training statistics. The mean of these statistics is taken and stored in a vector. Once every training signal has acted as the reference signal, the minimum value in the mean vector is determined, and the corresponding signal is selected as the reference signal for that region and SNR_A since the minimum value indicates that it is most similar to the other signals.

Once a reference signal is assigned for each region, the regions are concatenated to form a full signal at each SNR_A . The mean of all $N_{trng} = 500$ signal was selected for all $N_r = 33$ regions and at all SNR_A values.

3.5.3 Threshold Selection.

A threshold is necessary to make the binary decision of whether the unknown input signal $x_t[n]$ is *Normal* or *Anomalous*. After the reference signal is determined, it is used in the CBAD process along with the training signals, generating $N_{tstats} = 500$ statistics. These statistics create a Probability Mass Function (PMF) for the training

statistics z_{trng} at each SNR_A .

The training statistics are then sorted in descending order and the lowest test statistic exceeding $N_{Low} = 90\%$ of the calculated training statistics is selected as the threshold t_v . The threshold values are determined for each SNR_A considered. The PMF and selected threshold value for z_{trng} at $\text{SNR}_A = 40$ dB for DUT11 is shown in Figure 20.

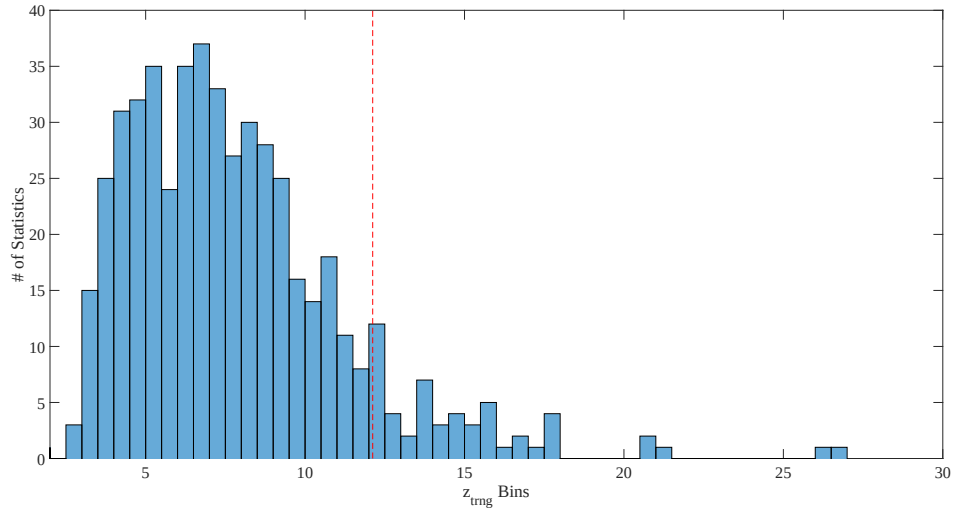


Figure 20. PMF of $N_{trng} = 500$ training statistics (blue) generated by the red lot reference signal from DUT11 at $\text{SNR}_A = 40$ dB, with 90% threshold displayed (red dotted line). Any testing statistic greater than the threshold value t_v will be declared *Anomalous*.

3.5.4 Declaration.

Once a threshold is established, a declaration can be made about the current operating condition of a particular DUT. A $N_{sig} = 1000$ testing signals from both *Normal* and *Anomalous* conditions (for a total of $N_{tst} = 2000$ signals) from each device are used to verify CBAD performance in this research. Statistics are generated for each of the $N_{tst} = 2000$ signals using the appropriate reference signal based on manufacturing lot. The generated statistics z_v represent the similarity between an unknown signal and a known reference signal, and the closer the statistic is to zero the

more similar the unknown signal is to the reference signal. Therefore, if a statistic z_v exceeds the threshold, the signal is declared as *Anomalous*. Conversely, if a statistic z_v is less than the threshold, the signal is declared as *Normal*.

For reasons described in Chapter IV, the reference signal must come from the same manufacturing lot in order to achieve the desired performance.

IV. Results

This chapter provides results of Microcontroller Unit (MCU) operation condition verification using the Correlation-Based Anomaly Detection (CBAD) process. Section 4.1 discusses the specific criteria by which the CBAD performance is evaluated. Section 4.1 defines the performance evaluation criteria and the benchmarks used to evaluate CBAD performance. Section 4.2 presents the results derived from comparing a reference signal with collections from the same device the reference signal is generated.

It was determined that a reference signal generated from one manufacturing lot could not produce the desired results when correlated with a signal collected from a device manufactured in a different lot. Both results derived from comparing devices from the same lot and comparing devices from a different lot are presented in Section 4.3. Lot discrimination was investigated after determining lot origin affected anomaly detection performance, and is discussed in Section 4.4. Section 4.5 gives the overall results of the research with every reference signal and device permutation.

4.1 Performance Evaluation Criteria

Before discussing the results, it is important to understand the performance metrics used in this research. The terms *Anomaly detection* and *Normal verification* are related to whether an operation is verified as *Normal* or declared as *Anomalous* as indicated in Table 3.

Table 3. All possible outcomes of the CBAD process

Actual	Declared	Outcome
Normal	Normal	True Normal Verification
Normal	Anomalous	False Anomaly Detection
Anomalous	Normal	False Normal Verification
Anomalous	Anomalous	True Anomaly Detection

The metrics used in regards to *Normal* verification and *Anomalous* declaration are described as follows, where N_{tst} is the total number of *Normal* statistics generated when calculating *Normal* verification rates, or the total number of *Anomalous* statistics generated when calculating *Anomalous* detection rates:

- True Anomaly Detection Rate (TADR): The percentage of signals correctly declared Anomalous is calculated by

$$\text{TADR} = 100 \times \frac{\text{True Anomaly Detections}}{N_{tst}} \quad (8)$$

- True Normal Verification Rate (TNVR): The percentage of signals correctly declared Normal is calculated by

$$\text{TNVR} = 100 \times \frac{\text{True Normal Verifications}}{N_{tst}} \quad (9)$$

- False Anomaly Detection Rate (FADR): The percentage of Normal signals incorrectly declared Anomalous is calculated by

$$\text{FADR} = 100 - \text{TNVR} \quad (10)$$

- False Normal Verification Rate (FNVR): The percentage of Anomalous signals incorrectly declared Normal is calculated by

$$\text{FNVR} = 100 - \text{TADR} \quad (11)$$

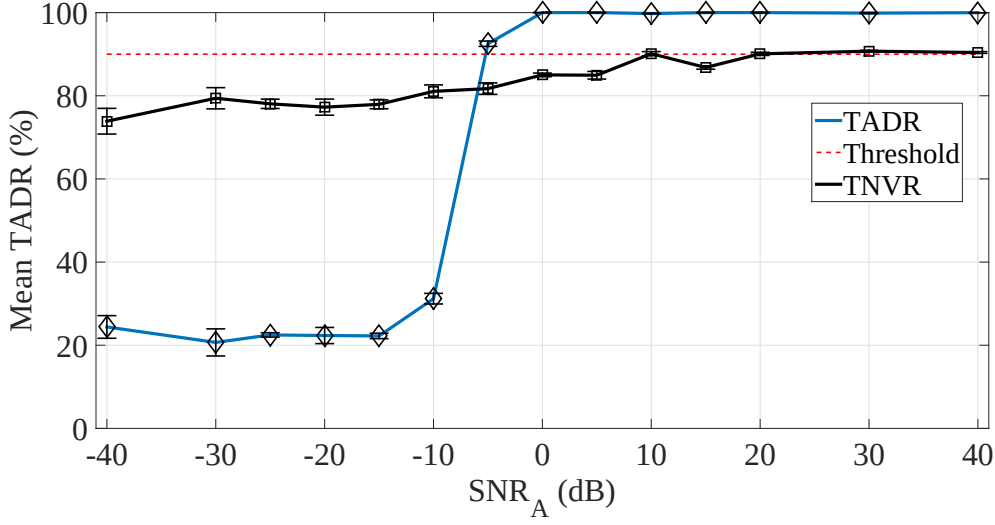
- Equal Error Rate (EER): The value at which the $\text{FADR} = \text{FNVR}$.

The goal of this research was to achieve benchmark performance of $\text{TADR} \geq 90\%$ while maintaining $\text{FADR} \leq 10\%$, as established in previous Air Force Institute of Technology (AFIT) research [79–81, 96]. Using the same benchmarks as previous research supports an accurate comparison between methodologies. The main presentation vehicles for performance results are plots showing SNR_A versus TADR and Receiver Operating Characteristic (ROC) curves. ROC curves displayed are for the first value of SNR_A where $\text{EER} \leq 10\%$, unless the device/reference signal combination never met the benchmarks, in which case the ROC curve at the highest SNR_A value is shown.

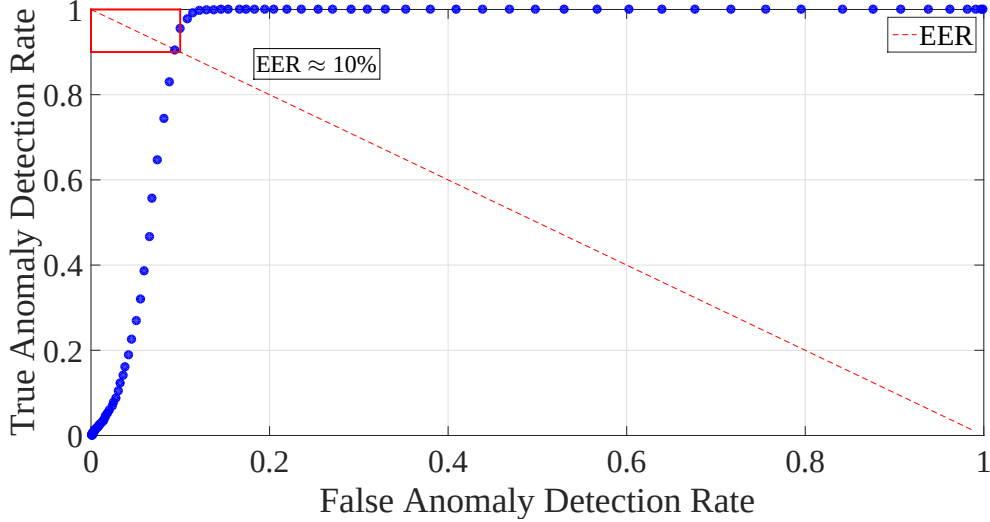
4.2 Like-Device Performance

The results presented in this section show the effectiveness of the CBAD process when using a reference signal against the Device Under Test (DUT) from which the reference signal is generated. Because the DUTs were selected from two manufacturing lots, a reference signal was generated from Unintentional Radio Frequency Emissions (UREs) collected from DUT3 for anomaly detection in devices from Lot 1 (Blue lot) and from DUT11 for anomaly detection in devices from Lot 2 (Red lot). From this point on, the reference signal generated from DUT3 will be called the blue reference signal, and the reference signal generated from DUT11 will be called the red reference signal.

The first test case correlates the blue reference signal with unknown signals from DUT3, the same device from which the blue reference signal was generated, using $N_{tst} = 1000$ *Anomalous* testing signals and $N_{tst} = 1000$ *Normal* testing signals that had never been used in the CBAD process prior to this point. The *Normal* verification performance does not differ as much as *Anomalous* detection performance at lower SNR_A values since the reference signal and threshold are adapted using $N_{trng} = 500$ *Normal* training signals at each SNR_A . The threshold is based on verifying 90% of the *Normal* training signals. Figure 21a shows that the CBAD process meets the desired performance set forth in Section 4.1 with $\text{TADR} \geq 90\%$ for $\text{SNR}_A \geq -5$ dB, while $\text{TNVR} \geq 90\%$ for $\text{SNR}_A > 20$ dB ($\text{TNVR} = 90\%$ is equivalent to $\text{FADR} = 10\%$). The plot is created by averaging the TADR across $N_{Nz} = 5$ noise realizations using the CBAD anomaly detection results for $N_{tst} \times N_{Nz} = 5000$ *Anomalous* signals for each SNR_A considered. In addition to the SNR_A versus TADR plot, the ROC curve for using the blue reference signal versus DUT3 at $\text{SNR}_A = 5$ dB is presented in Figure 21b. Once $\text{SNR}_A < 5$ dB the performance benchmark of $\text{EER} \leq 90\%$ was not met.



(a) SNR_A versus TADR, blue lot device DUT3. 95% confidence intervals shown in black

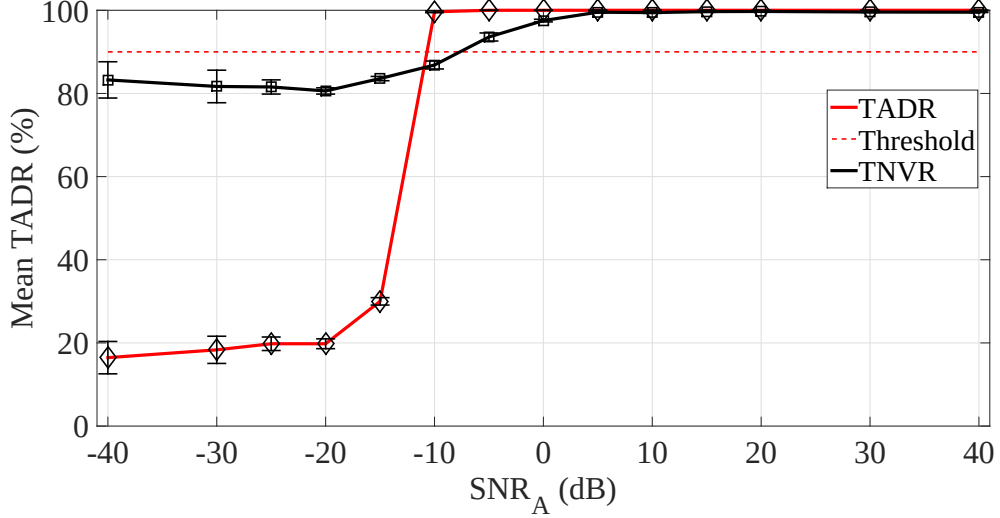


(b) ROC curve for DUT3 at $\text{SNR}_A = 5$ dB

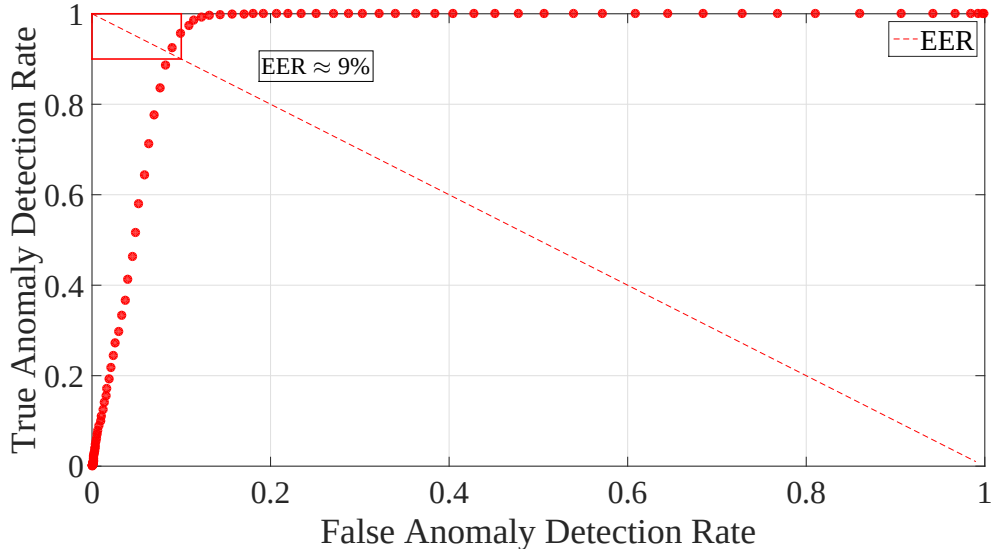
Figure 21. Single device performance correlating the blue reference signal with DUT3, the device from which the blue reference signal was generated. Figure 21a shows the SNR_A versus TADR plot, and Figure 21b shows the ROC curve at $\text{SNR}_A = 5$ dB. The benchmark of $\text{EER} \leq 10\%$ is met for all $\text{SNR}_A \geq 5$ dB.

The next test case correlates the red reference signal with unknown signals from DUT11, the same device from which the red reference signal was generated. Figure 22a shows $\text{TADR} \geq 90\%$ for $\text{SNR}_A \geq -10$ dB, while $\text{TNVR} \geq 90\%$ for $\text{SNR}_A \geq -5$ dB. In addition to the SNR_A versus TADR plot, the ROC curve for correlating

the red reference signal with DUT11 is presented in Figure 22b. Once $\text{SNR}_A < -10$ dB, the performance benchmark of $\text{EER} \leq 10\%$ were not met.



(a) SNR_A versus TADR, red lot device DUT11. 95% confidence intervals shown in black



(b) ROC curve for DUT11 at $\text{SNR}_A = -10$ dB

Figure 22. Single device performance correlating the red reference signal with DUT11, the device from which the red reference signal was generated. Figure 22a shows the SNR_A versus TADR plot and Figure 22b shows the ROC curve at $\text{SNR}_A = -10$ dB. The benchmark of $\text{EER} \leq 10\%$ is met for all $\text{SNR}_A \geq -10$ dB.

4.3 Cross-Device Performance

This section describes the performance achieved by the CBAD process when using a reference signal generated from one device to detect *Anomalous* activity in another device. Two scenarios were investigated: cross-device/cross-lot and cross-device/like-lot. The performance when using reference signals from a different lot than that of the test signal did not meet the benchmarks outlined in Section 4.1.

4.3.1 Cross-Lot.

Two of the possible permutations of using a reference signal from a different lot are chosen for this section; however, the ROC curves generated for all other permutations can be found in Appendix A and have similar results.

The first test case for the scenario in which the reference signal and test signal come from different lots used the blue reference signal against the collected signals from red lot device, DUT6. The ROC curve in Figure 23 shows that the benchmark of $\text{EER} \leq 10\%$ is not attained regardless of the SNR_A considered.

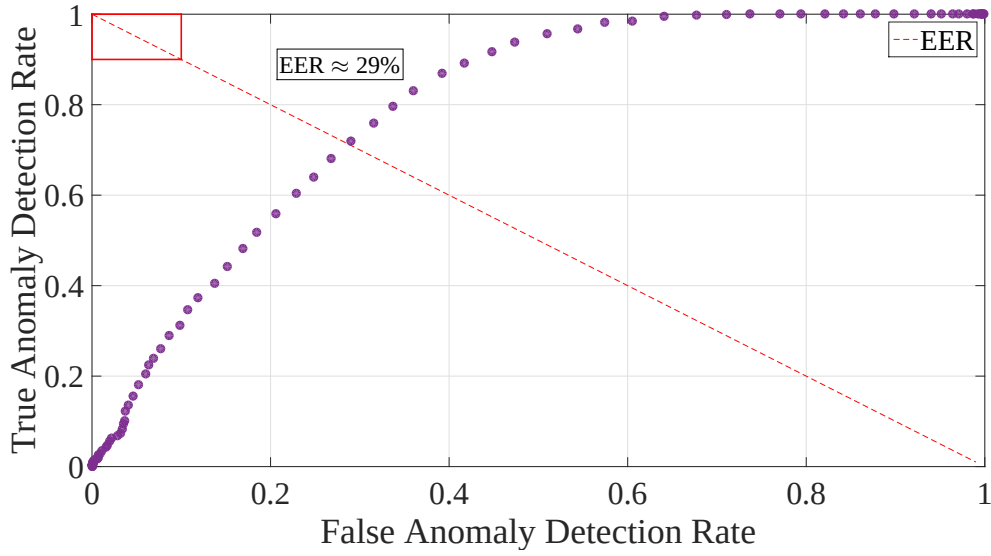


Figure 23. Case 1: ROC curve generated by using the blue reference signal generated from DUT3 against a red lot device, DUT6. The ROC curve was generated at $\text{SNR}_A = 40$ dB which is the highest investigated. Regardless of SNR_A the benchmark $\text{EER} \leq 10\%$ is not met.

The second test case for this scenario used the red reference signal against signals collected from a blue lot device, DUT4. Again, the ROC curve in Figure 24 shows that the benchmark of $\text{EER} \leq 10\%$ is not attained regardless of the SNR_A considered.

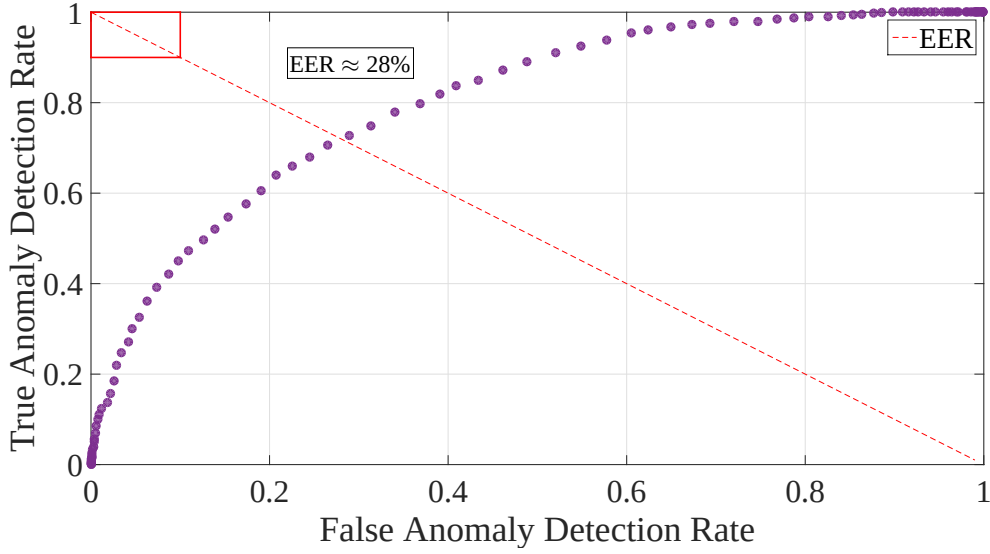


Figure 24. Case 2: ROC curve generated by using a red lot reference signal generated from DUT11 against a blue lot device DUT4. The desired benchmarks cannot be met regardless of SNR_A .

The results presented in this section show that cross-lot reference signals do not yield acceptable results. However, using a reference signal generated from a device within the same lot yielded results that met the desired benchmark $\text{EER} \leq 10\%$.

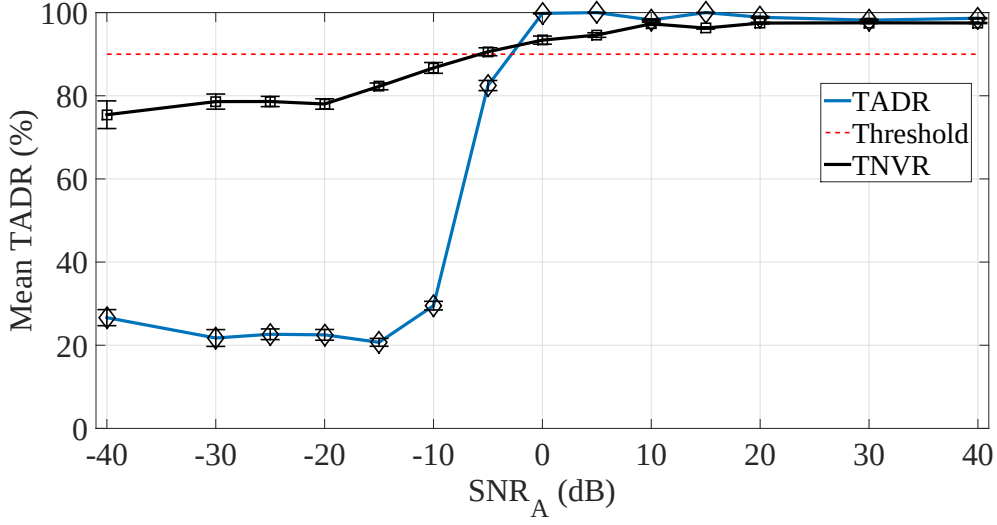
4.3.2 Like-Lot.

The test cases described in this section compare a reference signal from one lot to signals collected from another device in the same lot. The performance using the threshold determined during the reference signal generation step did not meet the established benchmarks for all devices. However, the ROC curves showed that a threshold existed that could produce the desired results. Therefore, the threshold was regenerated by comparing $N_{rs} = 200$ secondary training signals from the device being examined with the already determined reference signal and calculating the threshold in the same manner described in Section 3.5.3.

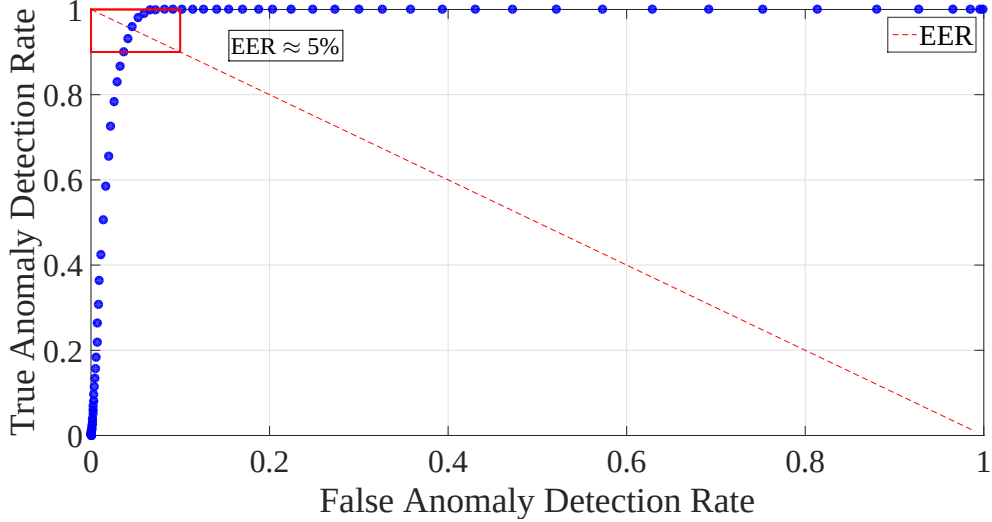
4.3.2.1 No Threshold Regeneration.

The results of the CBAD process without regenerating the threshold value are presented here. The ROC curves and SNR_A versus TADR plots of two permutations are presented. The remaining results for all other permutations can be found in Appendix A.

Figure 25 shows the ROC curve and the SNR_A versus TADR plot for the blue reference signal correlated with DUT4 from the blue lot. Figure 25a shows that the CBAD process is able to meet the benchmarks with the threshold calculated from the training statistics generated from the blue lot device DUT3. The ROC curve in Figure 25b shows a threshold exists such that the benchmarks can be met by observing that $\text{EER} \leq 10\%$ for $\text{SNR}_A \geq 5$ dB. The section of the ROC curve where $\text{EER} \leq 10\%$ is indicated in the figure by a red box.



(a) TADR versus SNR_A plot for blue reference signal versus DUT4

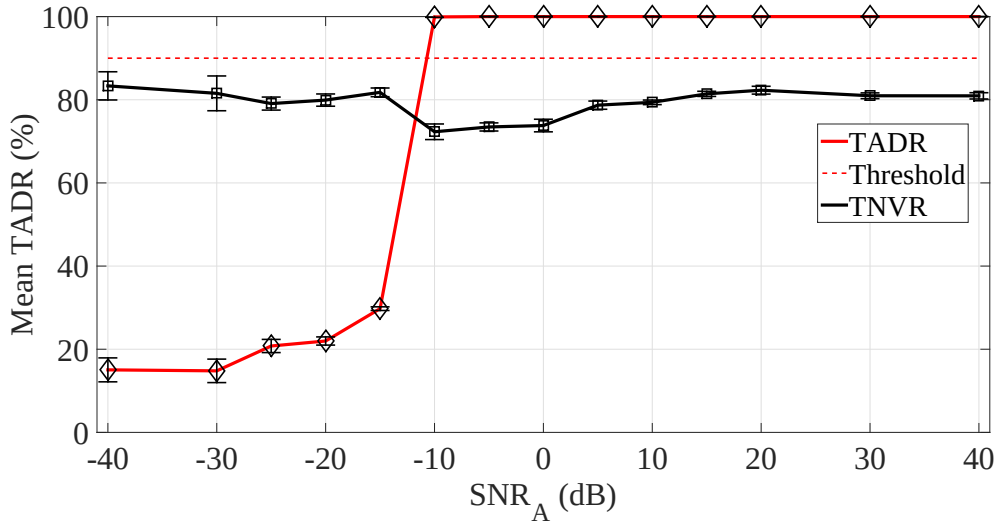


(b) ROC curve for DUT4 at $\text{SNR}_A = 5$ dB when correlated with the blue lot reference signal

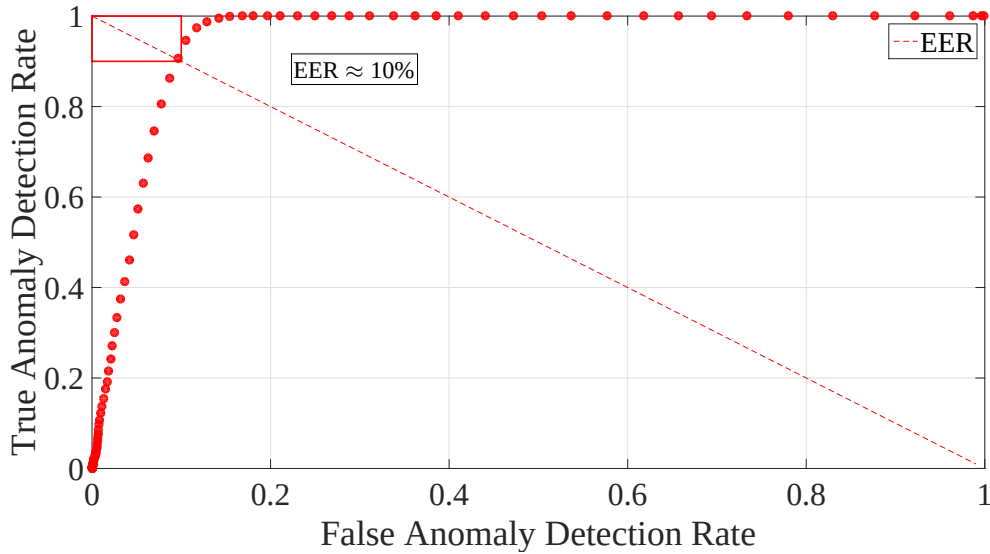
Figure 25. Performance plots for the blue reference signal correlated with signals collected from DUT4. Figure 25a shows the SNR_A versus TADR plot and Figure 25b shows the ROC curve at $\text{SNR}_A = 5$ dB

Figure 26 shows the ROC curve and the SNR_A versus TADR plot for the red reference signal generated using signals from DUT11 correlated with DUT5. While Figure 26a shows the benchmark of $\text{TADR} \geq 90\%$ is met at higher SNR_A values, $\text{TNVR} \leq 81\%$ at the same SNR_A values. In this case, the threshold is set too low based on selecting a threshold using a different device (DUT11 when generating the

reference signal) and is declaring more *Normal* signals as *Anomalous* than is allowed based on the benchmark of $\text{TNVR} \geq 90\%$. However, just as the previous test case, the ROC curve in Figure 26b shows that a threshold exists such that both benchmarks can be met if the correct threshold is selected.



(a) TADR versus SNR_A plot for Red reference signal versus DUT5 without regenerating the threshold



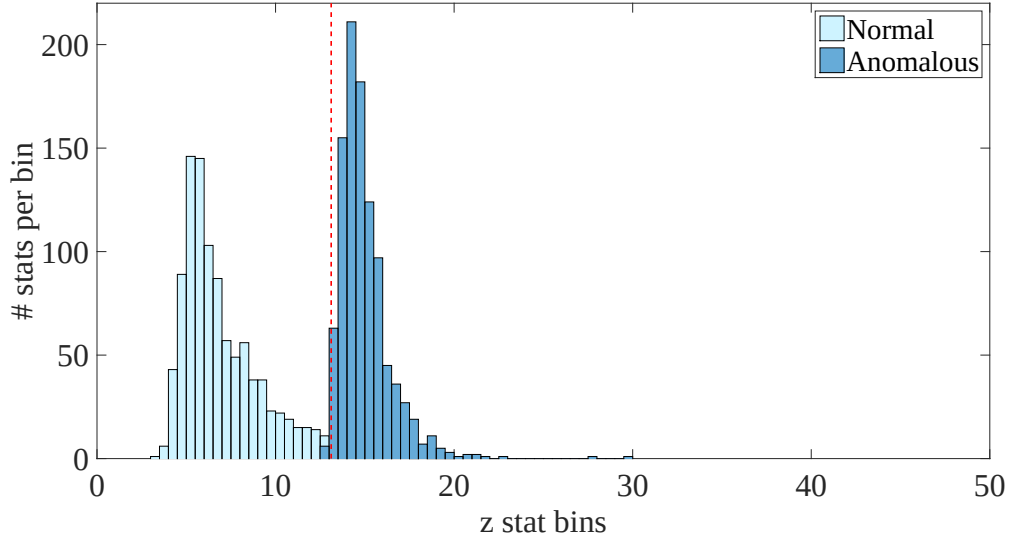
(b) ROC curve for DUT5 at $\text{SNR}_A = -5$ dB when correlated with the Red lot reference signal. Figure 26a shows the SNR_A versus TADR plot, and Figure 26b shows the ROC curve at $\text{SNR}_A = 5$ dB

Figure 26. Performance plots for the red reference signal correlated with signals collected from DUT5

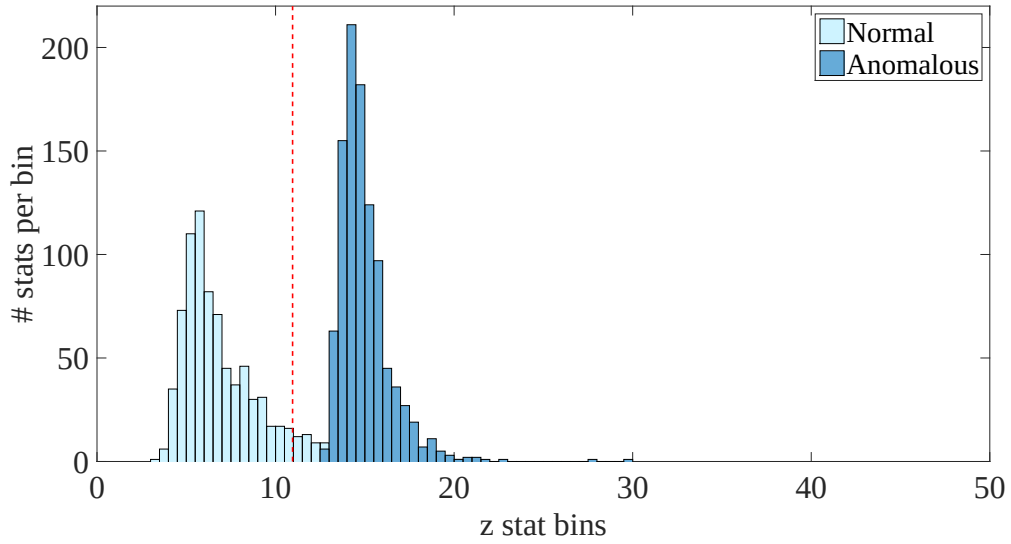
4.3.2.2 Threshold Regeneration.

Because the ROC curves for devices correlated with a reference signal from the same lot showed the possibility of meeting the benchmarks, the thresholds were regenerated for each device using the same process as described in Section 3.5.3. This process consisted of separating $N_{rs} = 200$ secondary training signals from the $N_{tst} = 1000$ testing signals to create a secondary training matrix. The new threshold was calculated in the same way as described in Section 3.5.3, but substituted the cross-device statistics for the like-device statistics when building the Probability Mass Function (PMF). The same two test cases described in the previous section are used to show the difference in performance after regenerating the threshold.

Figure 27 shows the PMFs of both *Normal* and *Anomalous* statistics for DUT4 when correlated with the blue reference signal at $\text{SNR}_A = 40$ dB, and the threshold used to make a declaration at that SNR. Figure 27a shows the original threshold based off of the original training matrix is too high to meet the benchmarks. Figure 27b shows the new threshold, and produces the results seen in Figure 28. With the new threshold, the CBAD process is shown to meet the desired benchmarks.



(a) Original threshold based on training statistics from DUT3.



(b) Regenerated threshold

Figure 27. PMFs of DUT4 statistics at $\text{SNR}_A = 40$ dB. Thresholds are shown before (27a) and after (27b) regenerating the threshold value. Note the *Normal* statistic PMF after the threshold has $N_{rs} = 200$ less statistics since they were used to calculate the threshold and were not reused.

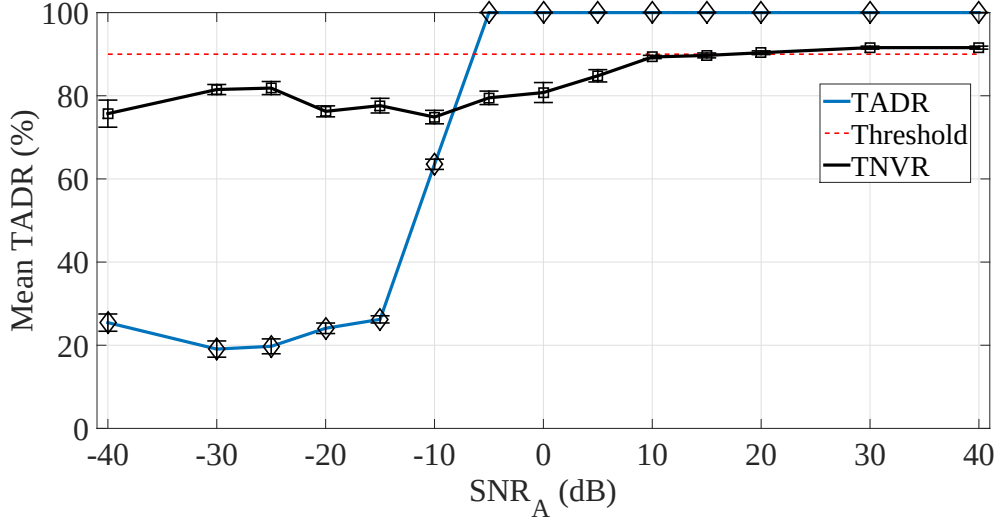
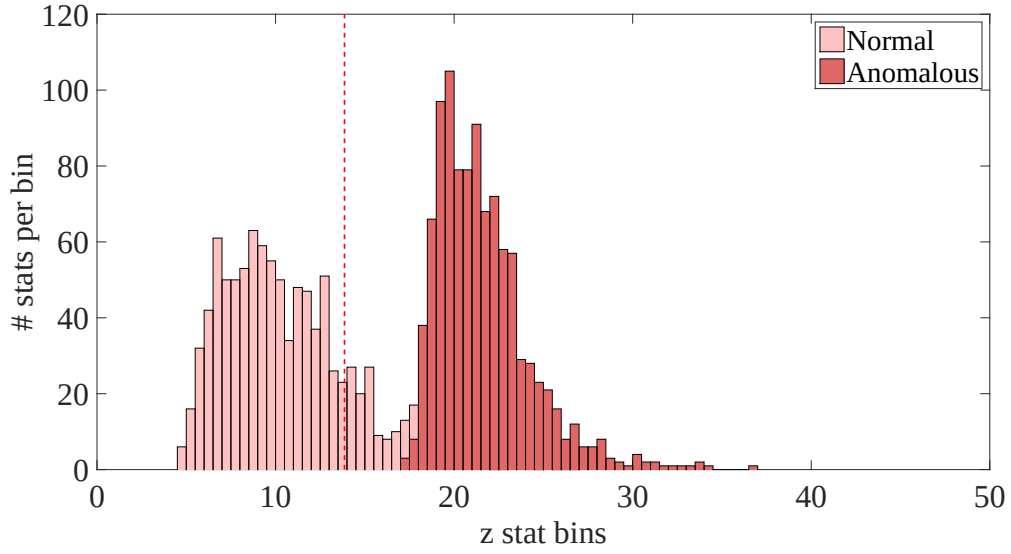
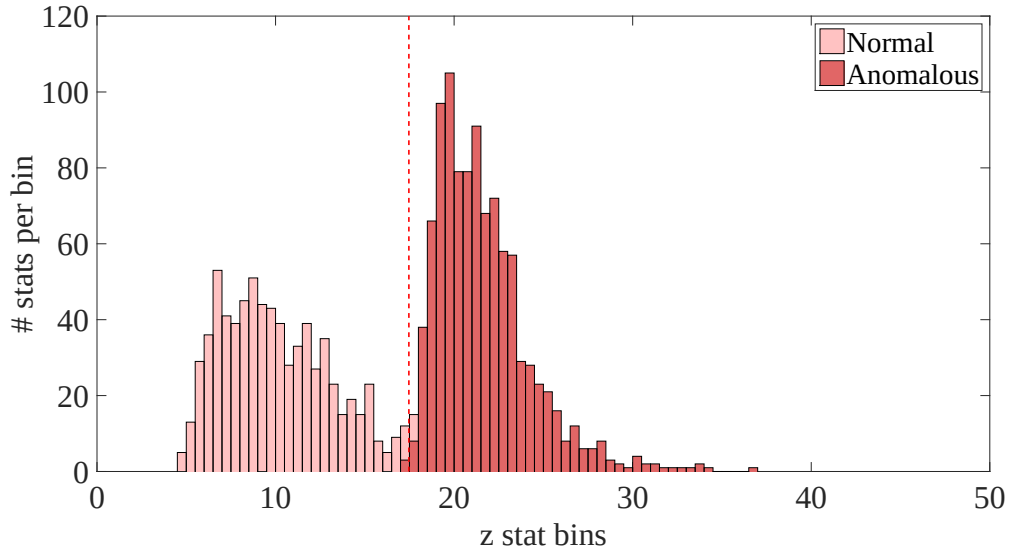


Figure 28. DUT4 SNR_A versus TADR plot after regenerating the threshold value. The benchmarks are still met after the regeneration for $\text{SNR}_A \geq 10$ dB.

Figure 29 shows the PMFs of both *Normal* and *Anomalous* statistics for red lot device DUT5 when correlated with the red reference signal at $\text{SNR}_A = 40$ dB, and the threshold used to make a declaration at that SNR. Figure 29a shows the original threshold based off of the original training matrix is too low to meet the benchmarks, and classifies more *Normal* signals as *Anomalous* than is allowed based on the benchmark of $\text{TNVR} \geq 90\%$. Figure 29b shows the new threshold, and produces the results seen in Figure 30. With the new threshold, the CBAD process is shown to meet the desired benchmark of $\text{EER} \leq 10\%$.



(a) Original threshold based on training statistics from DUT11.



(b) Regenerated threshold

Figure 29. PMFs of DUT5 statistics at $\text{SNR}_A = 40$ dB. Thresholds are shown before (29a) and after (29b) regenerating the threshold value. Note the *Normal* statistic PMF after the threshold has $N_{rs} = 200$ less statistics since they were used to calculate the threshold and were not reused.

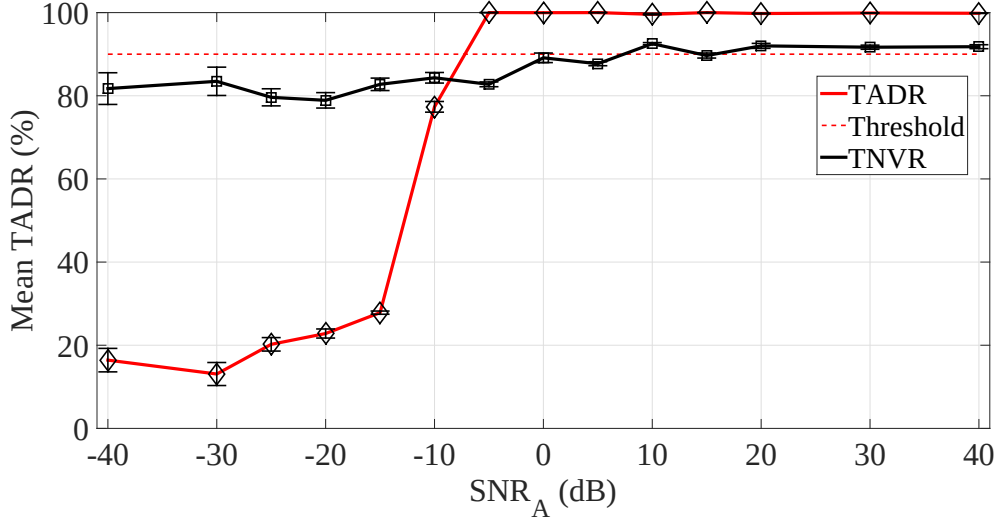


Figure 30. DUT5 TADR versus SNR_A plot after regenerating the threshold value. Once the threshold value is regenerated, the desired benchmarks are met for both TADR and TNVR.

4.4 Lot Discrimination

As noted previously, performance of the CBAD process was dependent on the manufacturing lot from which the DUT originated. Further investigation into lot discrimination was performed and is detailed here.

When investigating lot discrimination, all signals from all like-lot devices were combined, for a total of $N_{tst} = 3000$ signals in the red lot matrix, and $N_{tst} = 2000$ signals in the blue lot matrix. The use of both red and blue reference signals was investigated. For the purpose of evaluating lot discrimination, signals originating from the blue lot will be classified as *Normal* while signals originating from the red lot will be classified as *Anomalous* when using the blue reference signal. Similarly, signals originating from the red lot will be classified as *Normal* while signals originating from the blue lot will be classified as *Anomalous* when using the red reference signal. Using the same threshold regeneration process described in Section 4.3.2.2 the threshold was regenerated based on the new matrix using $N_{strng} = 500$ secondary training signals.

Figure 31 shows the ROC curve generated at $\text{SNR}_A = -5$ dB when using the blue reference signal and shows that CBAD performance achieves the benchmark $\text{EER} \leq 10\%$ for $\text{SNR}_A \geq -5$ dB.

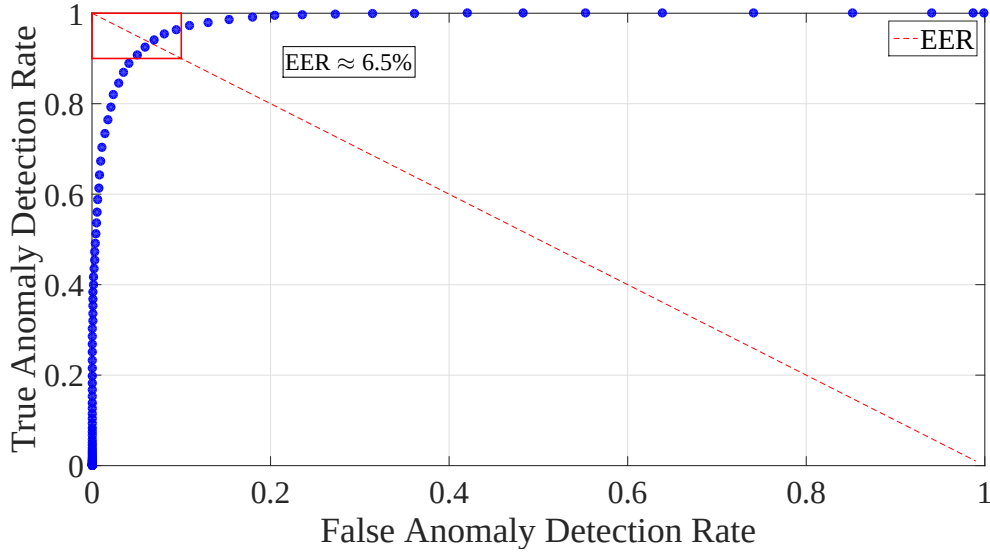


Figure 31. ROC curve generated at $\text{SNR}_A = -5$ dB with performing lot discrimination using the blue reference signal. Devices from the blue lot are considered *Normal* and devices from the red lot are considered *Anomalous* for the purpose of lot discrimination using the blue lot reference signal.

Figure 32 shows the PMFs generated with the $N_{tst} = 3000$ red lot statistics and $N_{tst} = 1500$ blue lot statistics after regenerating the threshold value using the blue reference signal. Note that the red lot statistics create a bimodal distribution. By comparing Figures 52, 58 and 66 in Appendix A, one can see that the statistics from DUT6 and DUT11 closely align when the blue reference signal is used while the statistics from DUT5 do not, thus creating a bimodal distribution.

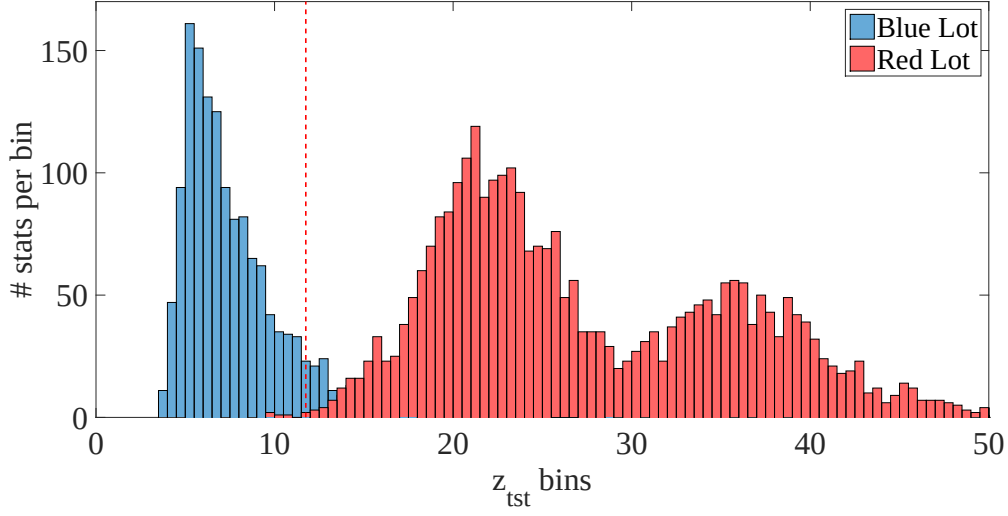


Figure 32. Red and blue lot PMFs generated with $N_{tst} = 3000$ red statistics and $N_{tst} = 1500$ blue statistics when correlated against the blue reference signal.

Figure 33 shows the rate at which devices that do not originate from the blue manufacturing lot are detected as *Anomalous*. The use of *Anomalous* in this case refers to devices that do not come from the same lot as the reference signal. *Normal* in this case refers to devices that are from the same lot as the reference signal. When $\text{SNR}_A \geq -10$ dB, the detection rate surpasses the benchmark of TADR $\text{TADR} \geq 90\%$. However, the TNVR does not meet the benchmark $\text{TNVR} \geq 90\%$ until $\text{SNR}_A \geq 15$ dB.

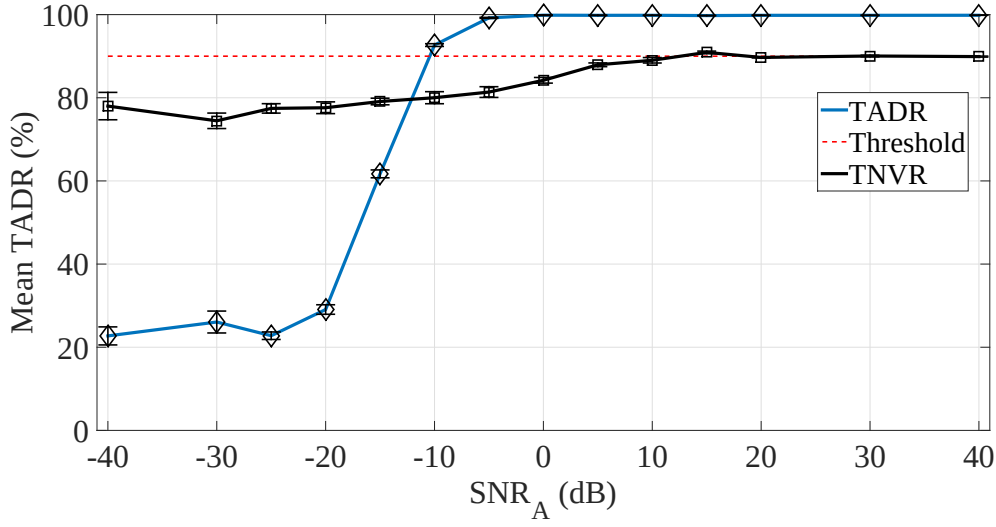


Figure 33. Lot discrimination SNR_A versus Cross-lot Detection plot generated using the blue reference signal.

Figure 34 shows the ROC curve generated at $\text{SNR}_A = -5$ dB when using the red reference signal and shows that CBAD performance achieves the benchmark $\text{EER} \leq 10\%$.

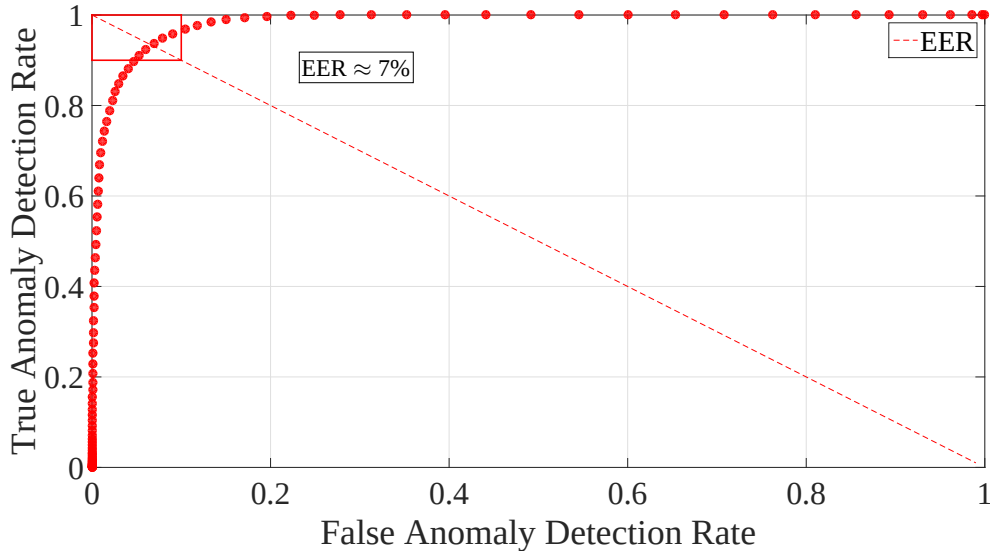


Figure 34. ROC curve generated at $\text{SNR}_A = -5$ dB with performing lot discrimination using the red reference signal. Devices from the red lot are considered *Normal* and devices from the blue lot are considered *Anomalous* for the purpose of lot discrimination using the red lot reference signal.

Figure 35 shows the PMFs generated with the $N_{tst} = 2500$ red lot statistics and $N_{tst} = 2000$ blue lot statistics after regenerating the threshold value using the red reference signal originating from DUT11. In contrast to the bimodal distribution seen in Figure 32, the blue lot statistics (*Anomalous* in this case) are not bimodal. Referencing Figures 72 and 80 shows that the statistics generated from using the red reference signal against both blue lot devices are similar and therefore do not create the same bimodal distribution seen in the previous case.

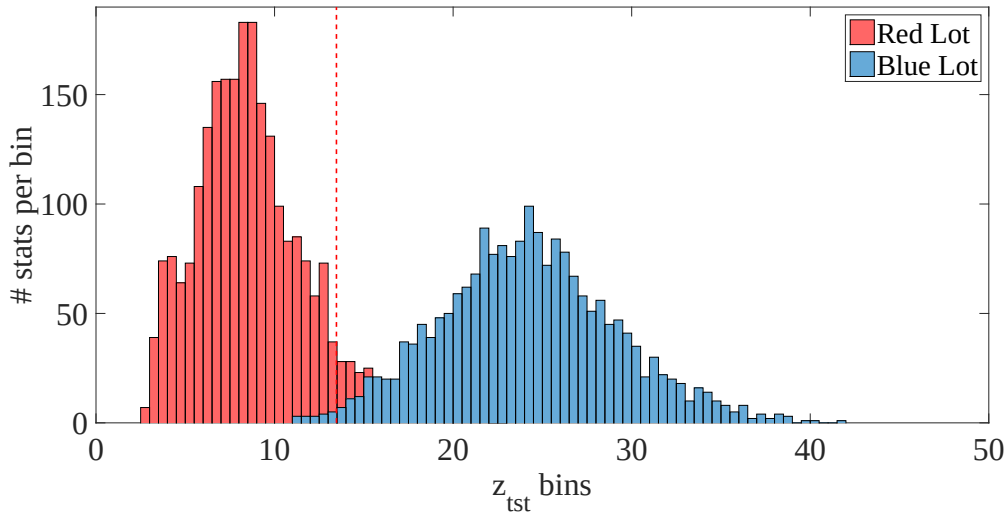


Figure 35. Red and blue lot PMFs generated with $N_{tst} = 2500$ red statistics and $N_{tst} = 2000$ blue statistics when correlated against the red reference signal.

Figure 33 shows the rate at which devices that do not originate from the red manufacturing lot are detected. When $\text{SNR}_A \geq -10$ dB, the detection rate surpasses the benchmark of $\text{TADR} \geq 90\%$. However, the TNVR does not meet the benchmark $\text{TNVR} \geq 90\%$ until $\text{SNR}_A \geq 5$ dB.

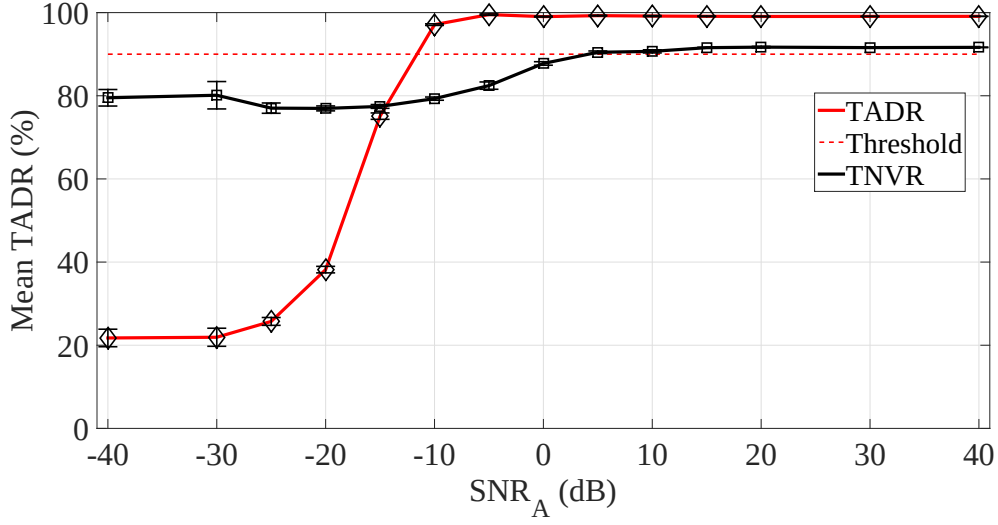


Figure 36. Lot discrimination SNR_A versus Cross-lot Detection plot generated using the red reference signal.

4.5 Consolidated Results

Table 4 shows the overall results of the research. Note that when using a reference signal generated from the same manufacturing lot the CBAD process is able to meet the benchmarks set forth in this research. In cases of cross-device, like-lot situations the threshold is regenerated using training signals from the DUT being evaluated. However, when using a reference signal generated from a different manufacturing lot the CBAD process is unable to meet the benchmark, regardless of the threshold chosen.

Table 4. Research results for every permutation of device and reference signal when used as inputs to the CBAD process. The benchmark condition to be met is $EER \leq 10\%$

		Blue Reference Signal	Red Reference Signal
Device Label	Lot Label	Meets Benchmark?	Meets Benchmark?
DUT3	Blue - Reference	Yes	No
DUT4	Blue	Yes	No
DUT5	Red	No	Yes
DUT6	Red	No	Yes
DUT11	Red - Reference	No	Yes

V. Conclusion

This chapter provides a summary of results for verifying the operating condition of a Microcontroller Unit (MCU). Section 5.1 provides a summary of the research purpose and findings. Section 5.2 presents recommendations for future work on verifying the operating condition of Integrated Circuits (ICs).

5.1 Research Summary

5.1.1 Purpose.

Microcontrollers are found in a large variety of electronic devices, and are even found in the peripherals (sensors, printers, etc.) of Supervisory Control and Data Acquisition (SCADA) systems and Industrial Control Systems (ICSs). Unfortunately, these particular systems that run critical infrastructures in the United States (US) have been lagging in implementing the most current security measures and are still vulnerable to cyber attack [32, 100].

Current methods of cyber attack detection rely on the upper layers of the Open Systems Interconnect (OSI) model [12], which are relatively easy for attackers to manipulate. This research proposes using the Correlation-Based Anomaly Detection (CBAD) algorithm, as developed in previous Air Force Institute of Technology (AFIT) research [81] to detect anomalous behavior on a microcontroller using Physical Layer attributes. The Physical Layer approach provides benefits in the realm of cyber security as signal emissions are based on the actual physical construction of the devices and are not feasibly replicated. This research adds to AFIT's ever-growing Radio Frequency Intelligence (RFINT) field of study in exploiting Physical Layer attributes using Side Channel Analysis (SCA) [13–16, 25, 39, 42, 72, 79–81, 96]

Research contributions were made by leveraging the CBAD process [81] and ap-

plying it to MCUs to detect anomalous behavior by verifying normal operation. Performance of operation verification was assessed by evaluating Signal-to-Noise Ratio (SNR) versus True Anomaly Detection Rate (TADR), and by generating Receiver Operating Characteristic (ROC) curves at the lowest value of SNR_A where the benchmark of Equal Error Rate (EER) $\leq 10\%$.

5.1.2 CBAD Results.

Collections came from a total of $N_D = 5$ devices, which were selected from two different manufacturing lots. $N_{Nz} = 5$ noise realizations of power-scaled, like-filtered Additive White Gaussian Noise (AWGN) were added to all signals to achieve the desired SNR_A in the input sequences. Two reference signals (one from each manufacturing lot) were generated using $N_{trng} = 500$ training signals from each reference device. Each reference signal was used as the reference input against all other transformed Device Under Test (DUT) collections, and the resulting statistics were assessed against the benchmarks discussed previously.

CBAD is shown to provide an effective means for detecting anomalous operations as represented by the *Normal* and *Anomalous* Advanced Encryption Standard (AES) programs if the reference signal is generated from a device within the same manufacturing lot as the DUT. This research found that if a reference signal from one manufacturing lot is compared with an unknown signal from the same lot, the EER $\leq 10\%$ benchmark was met in all cases when $\text{SNR}_A \geq 0$ dB. However, if a reference signal from one manufacturing lot is compared with an unknown signal from a different lot, the benchmark could not be met at any investigated value of SNR_A . Using this knowledge, it is also possible to discriminate between lots using the CBAD process. Device lot discrimination was successfully demonstrated for values of $\text{SNR}_A \geq -5$ dB. These preliminary results support further research into the use of the CBAD process

for Software (SW) and Hardware (HW) anomaly detection.

5.2 Future Work

The results presented here demonstrate the capability of CBAD in determining MCU operating conditions and provide preliminary results demonstrating potential of using CBAD for the purpose of HW discrimination. Results of this research motivate further investigation, including:

1. Alternate IC Devices: Signals used in this research were collected from a MSP430 microcontroller representative of MCUs found in embedded systems. Research could be further expanded by using Unintentional Radio Frequency Emissions (UREs) collected from other IC devices such as Field Programmable Gate Arrays (FPGAs) or microprocessors.
2. Radio Frequency Distinct Native Attributes (RF-DNA) versus CBAD: This research demonstrated the effectiveness of using the CBAD algorithm in determining MCU operating conditions. Previous AFIT research has been mainly focused on using RF-DNA to verify device identity. A comparison of the performance, timing and necessary resources can be made between the two methodologies.
3. Alternate Collection Equipment: The equipment used in this research is relatively expensive. Finding a probe and receiver that cost less, yet provide similar results could be beneficial when considering CBAD use in an operational environment.
4. Multiple Chip Collection Locations: This research generated a reference signal from one device per manufacturing lot and from the same chip location each

time. It was then compared with emissions collected at the same chip location for each DUT. This research can be expanded by investigating whether a reference signal collected at one location can be used to determine the operating condition of a signal collected at a different location or whether combining reference signals from multiple locations can provide more insight into device operations.

5. Device Discrimination: This research demonstrated CBAD's ability to discriminate between manufacturing lots. Investigation into whether it is possible to discriminate between devices manufactured in the same lot could prove beneficial and could be directly compared with methods such as RF-DNA in terms of efficiency and performance.

Appendix A

This appendix contains all results from comparing both red and blue reference signals to each Device Under Test (DUT).

Blue Reference vs. DUT3

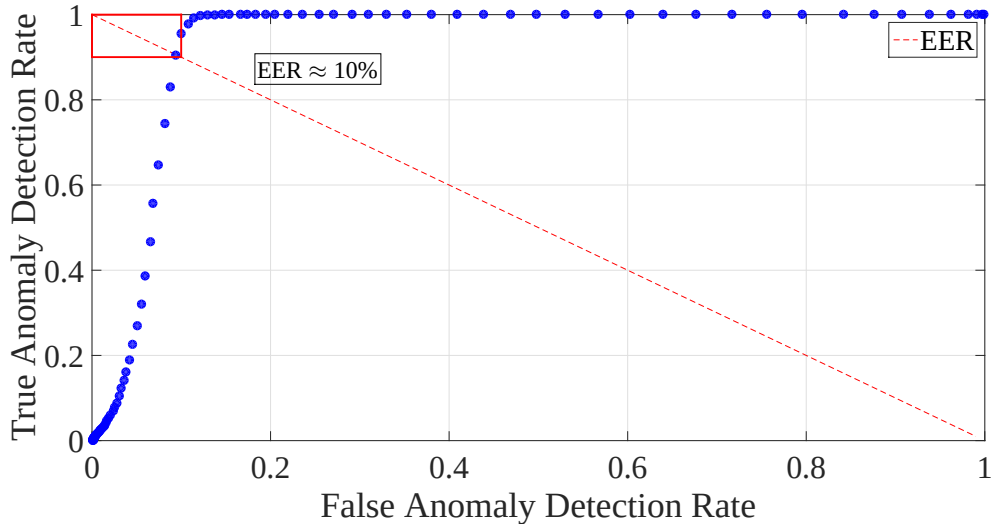


Figure 37. ROC Curve for blue device DUT3 using the blue reference signal at $\text{SNR}_A = 5\text{dB}$

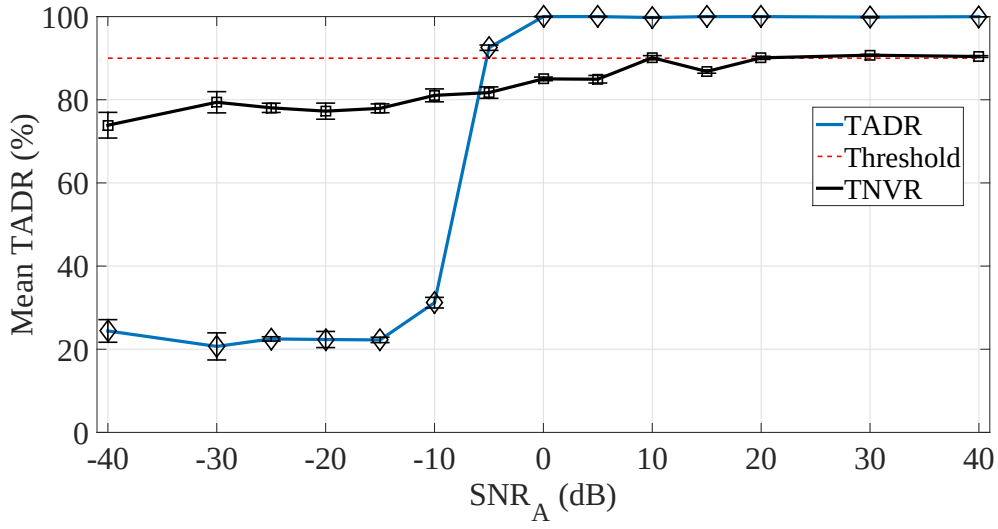


Figure 38. TADR vs. SNR_A for blue device DUT3 using the blue reference signal. No threshold regeneration is applied to this device since it generated the reference signal

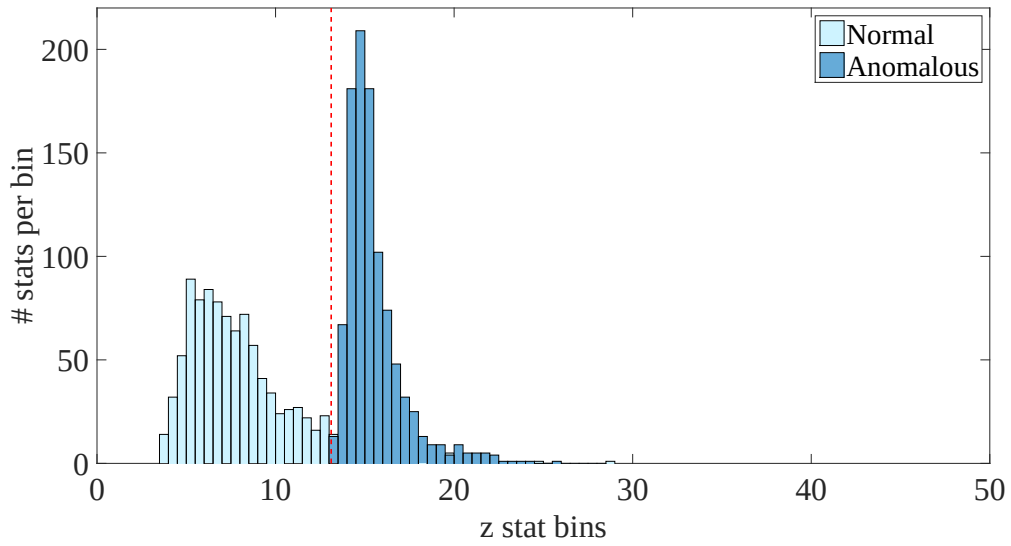


Figure 39. PMFs of DUT3 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition. Anything greater than the threshold value is declared as *Anomalous*

Blue Reference vs. DUT4



Figure 40. ROC Curve for blue device DUT4 using the blue reference signal at $\text{SNR}_A = 5\text{dB}$

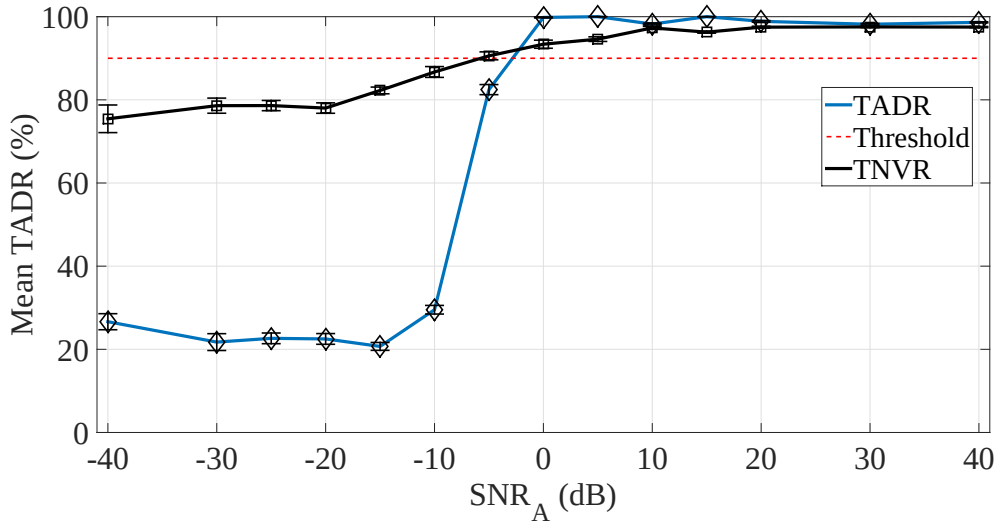


Figure 41. TADR vs. SNR_A for blue device DUT4 using the blue reference signal before regenerating the threshold value

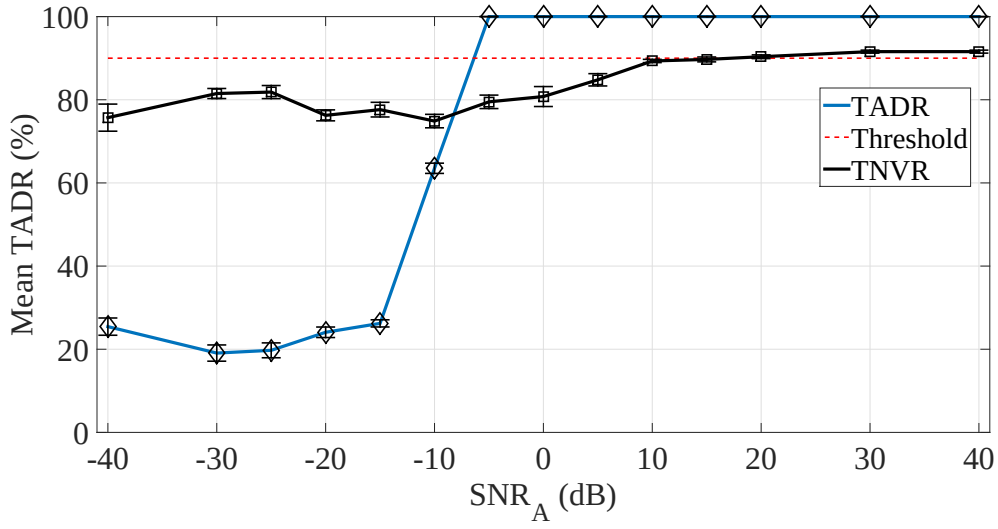


Figure 42. TADR vs. SNR_A for blue device DUT4 using the blue reference signal after regenerating the threshold value

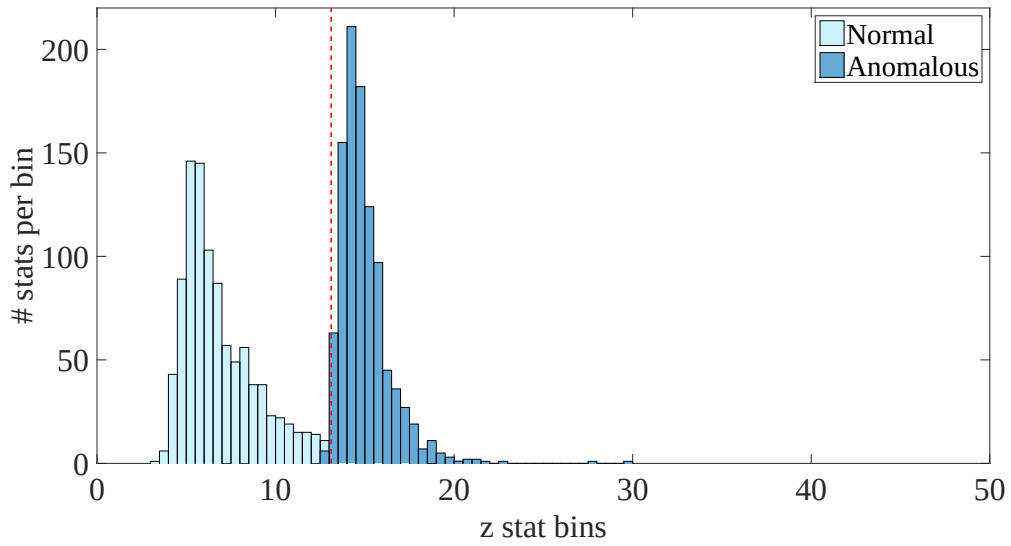


Figure 43. PMFs of DUT4 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition before regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

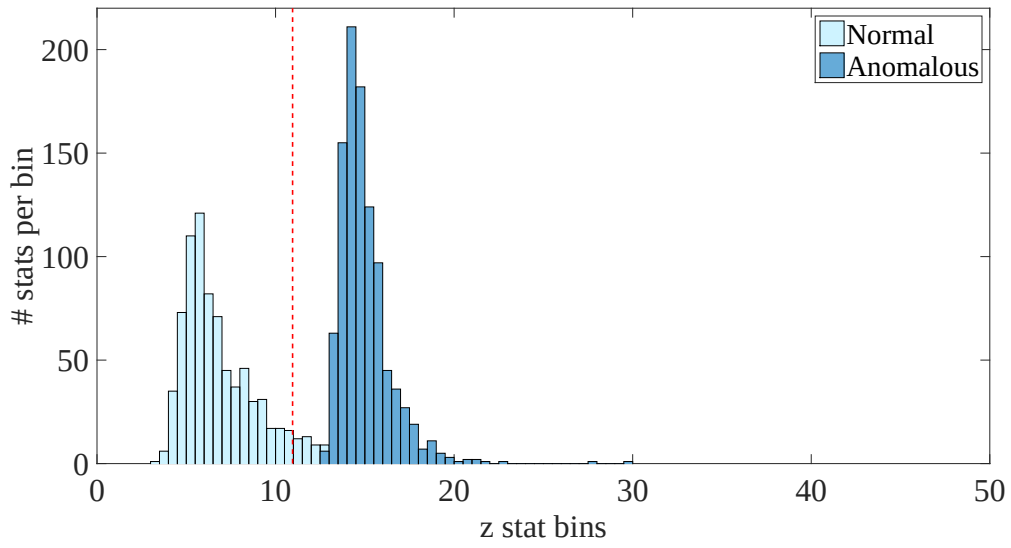


Figure 44. PMFs of DUT4 statistics at $\text{SNR}_A = 40 \text{ dB}$ and threshold (red dashed line) used to declare the operating condition after regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

Blue Reference vs. DUT5

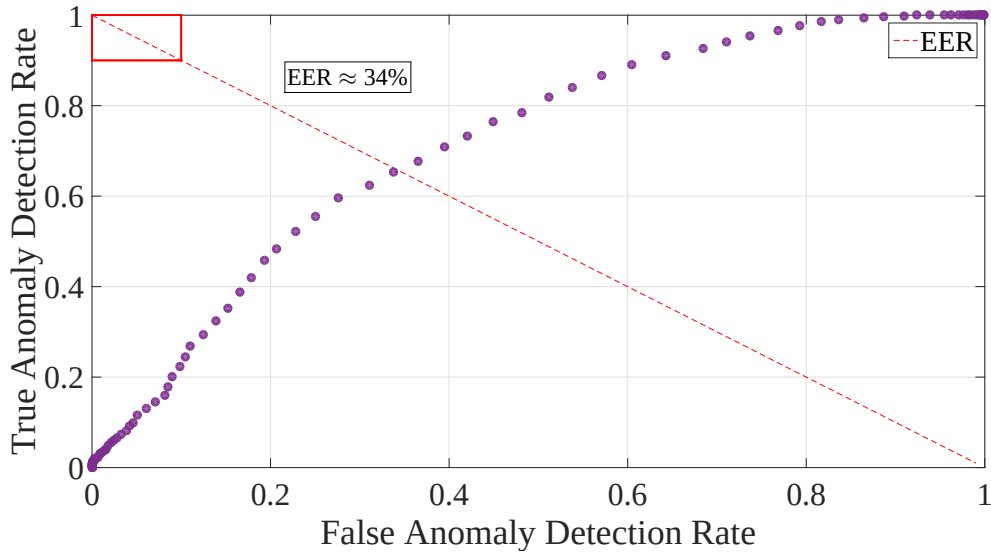


Figure 45. ROC Curve for blue device DUT5 using the blue reference signal at $\text{SNR}_A = 40 \text{ dB}$

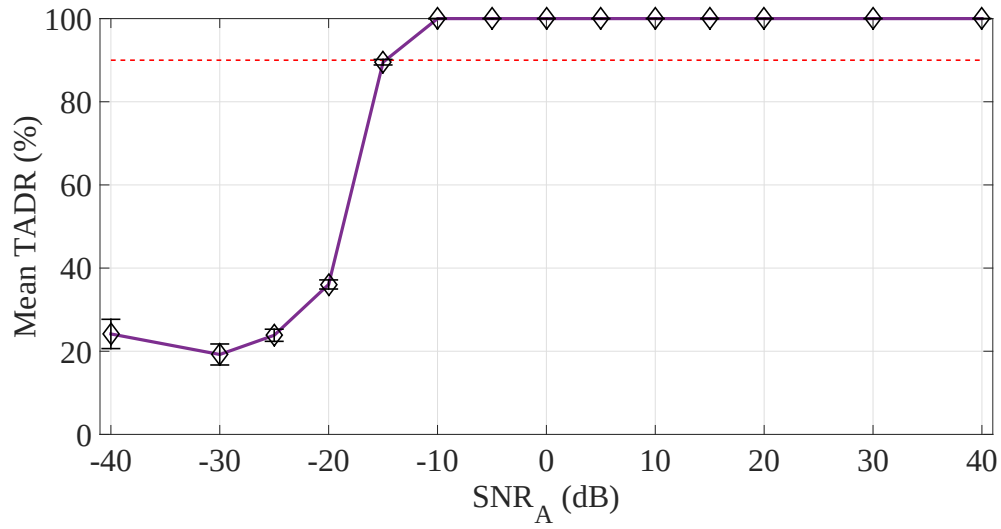


Figure 46. TADR vs. SNR_A for blue device DUT5 using the blue reference signal before regenerating the threshold value

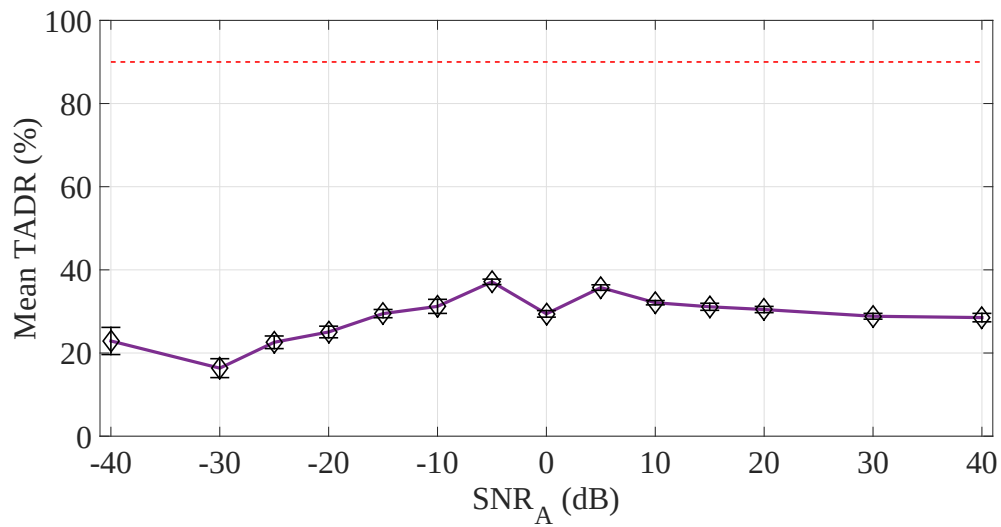


Figure 47. TADR vs. SNR_A for blue device DUT5 using the blue reference signal after regenerating the threshold value

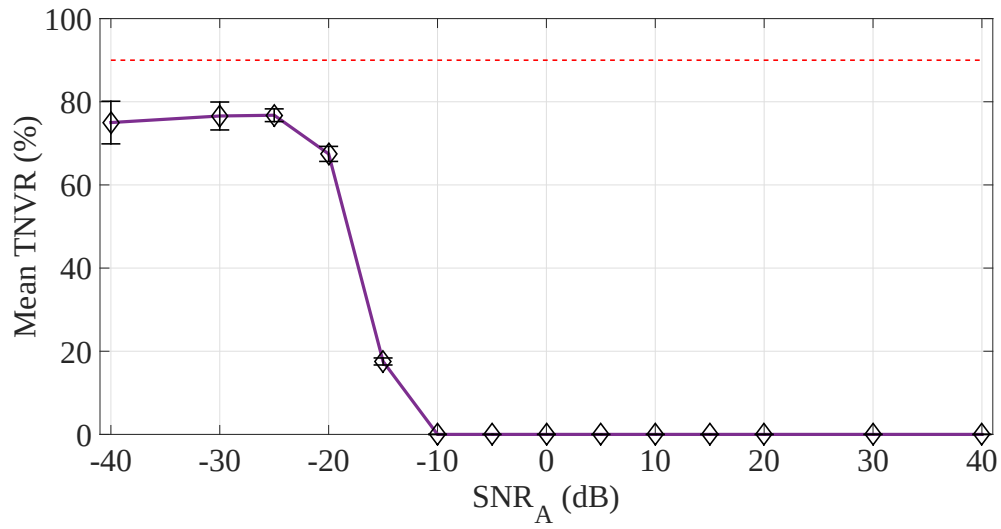


Figure 48. TNVR vs. SNR_A for blue device DUT5 using the blue reference signal before regenerating the threshold value

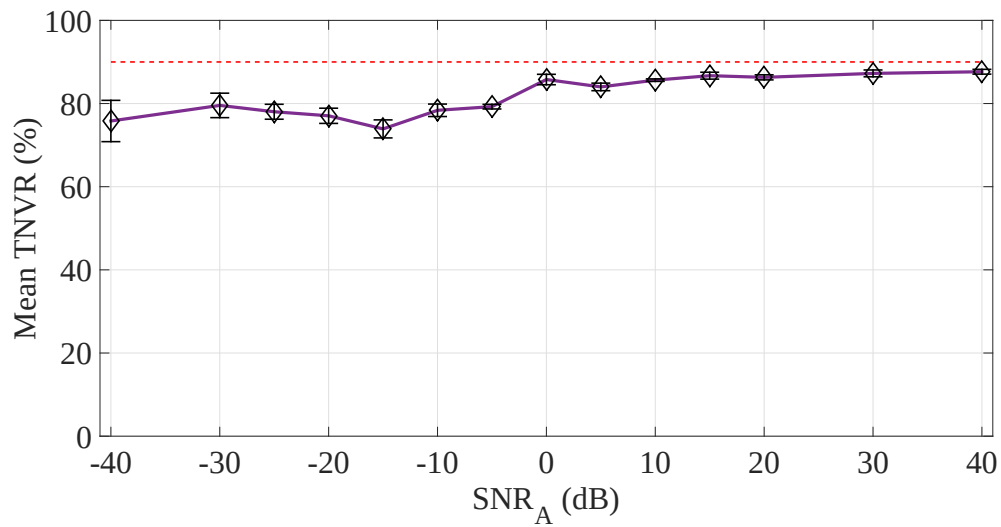


Figure 49. TNVR vs. SNR_A for blue device DUT5 using the blue reference signal after regenerating the threshold value

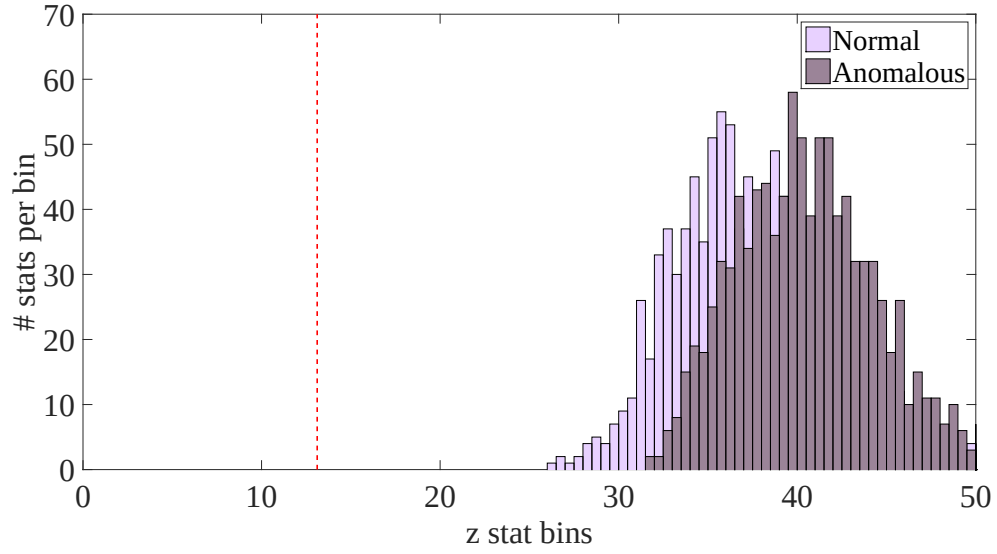


Figure 50. PMFs of DUT5 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition before regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

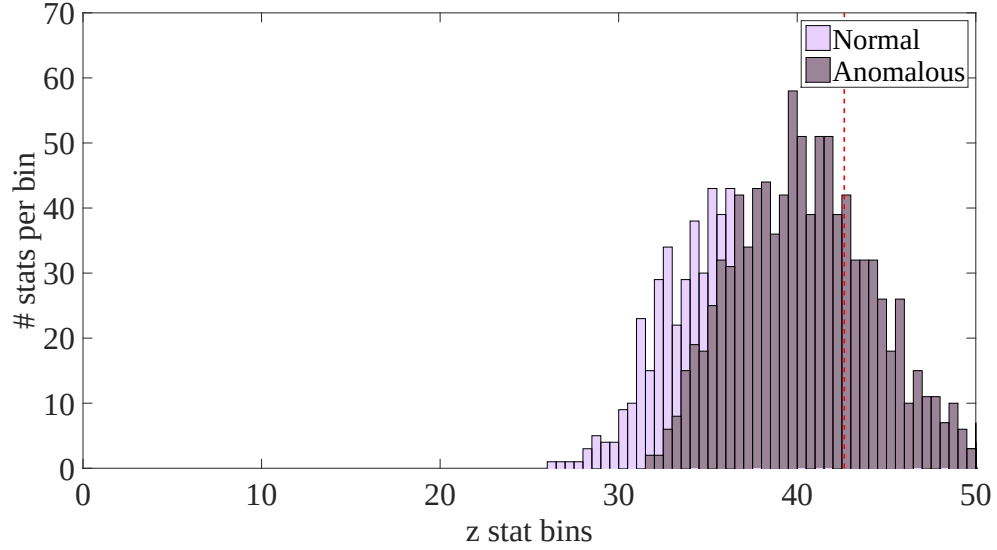


Figure 51. PMFs of DUT5 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition after regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

Blue Reference vs. DUT6

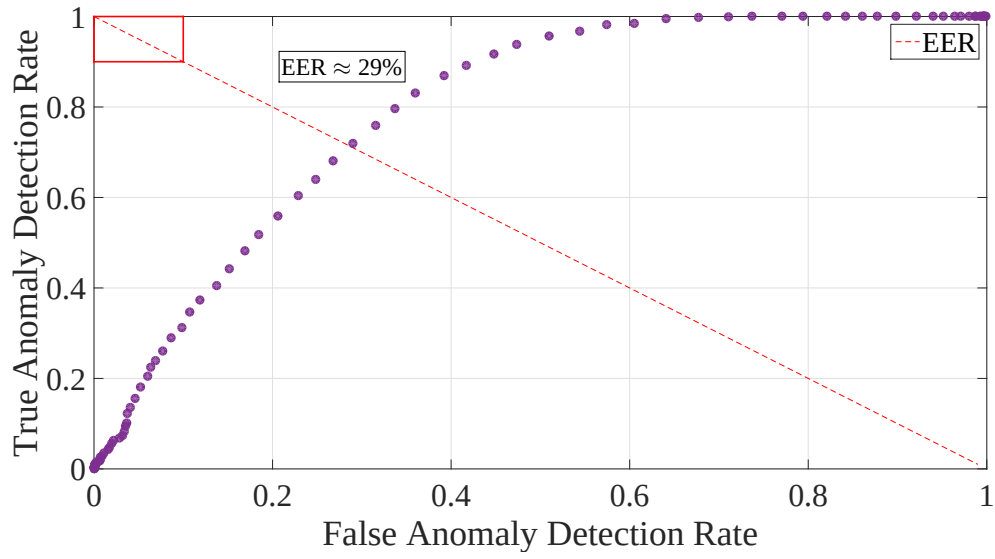


Figure 52. ROC Curve for blue device DUT6 using the blue reference signal at $\text{SNR}_A = 40\text{dB}$

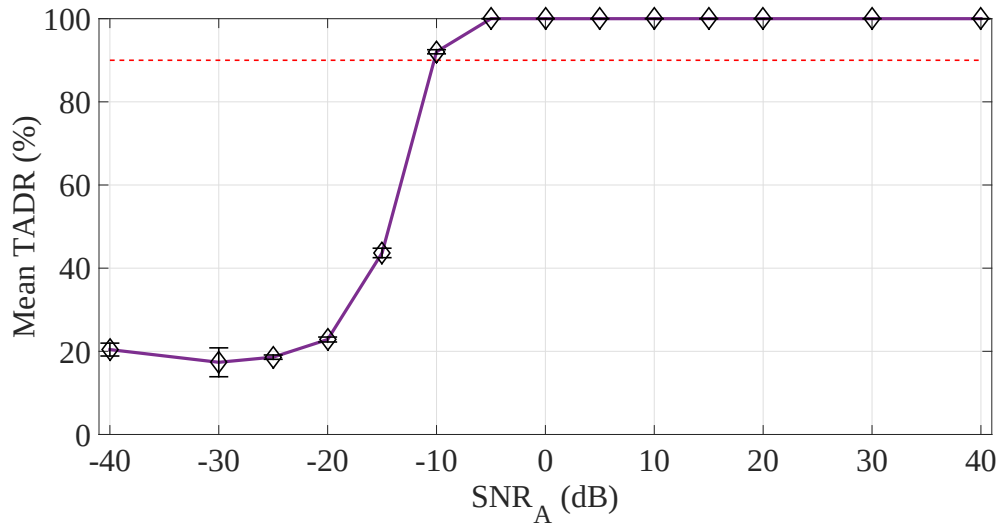


Figure 53. TADR vs. SNR_A for blue device DUT6 using the blue reference signal before regenerating the threshold value

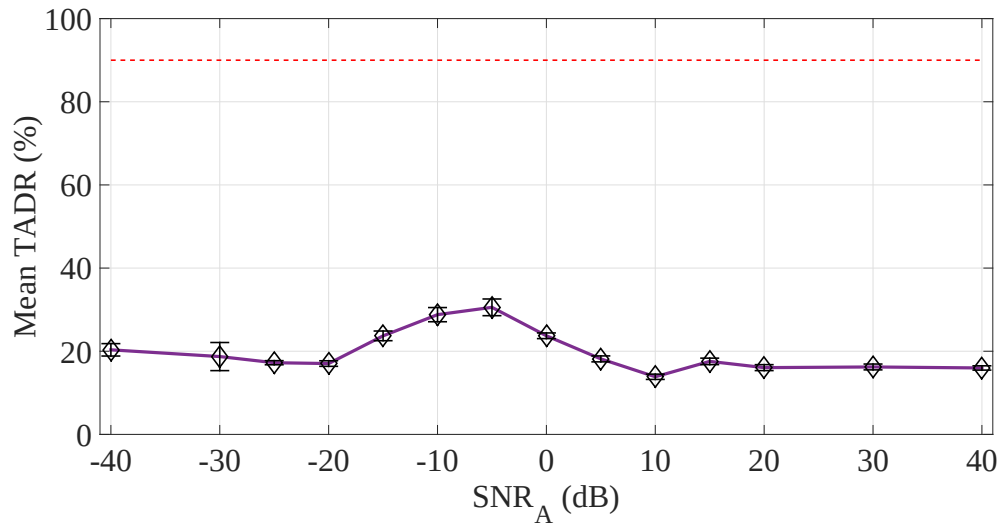


Figure 54. TADR vs. SNR_A for blue device DUT6 using the blue reference signal after regenerating the threshold value

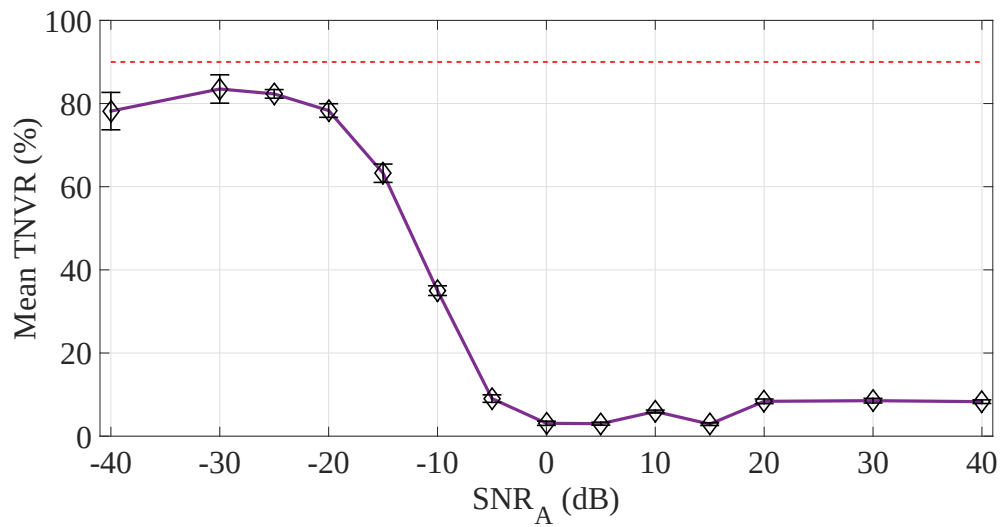


Figure 55. TNVR vs. SNR_A for blue device DUT6 using the blue reference signal before regenerating the threshold value

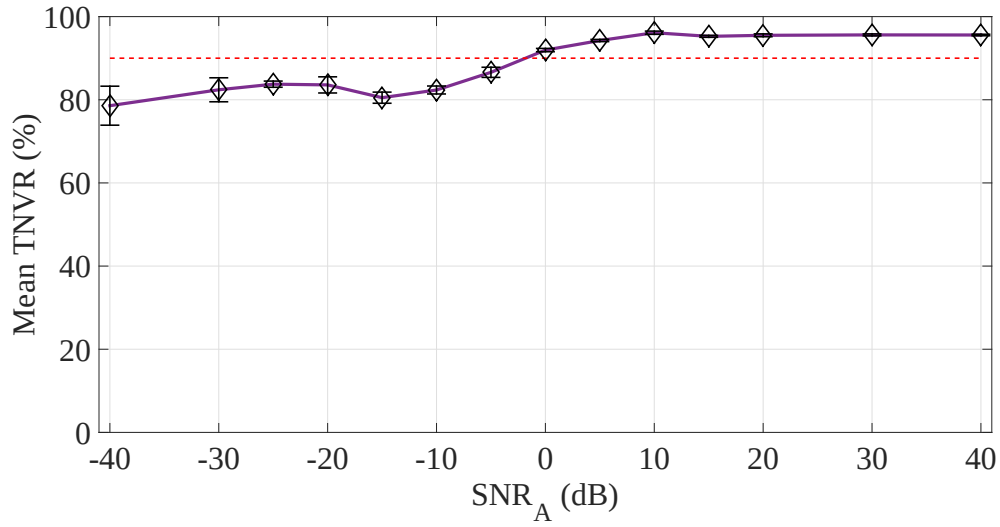


Figure 56. TNVR vs. SNR_A for blue device DUT6 using the blue reference signal after regenerating the threshold value

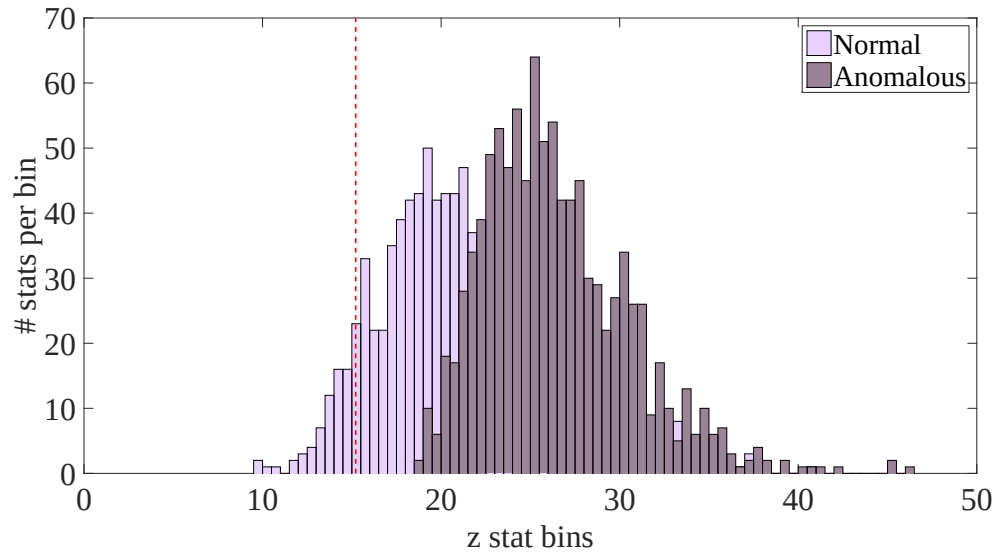


Figure 57. PMFs of DUT6 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition before regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

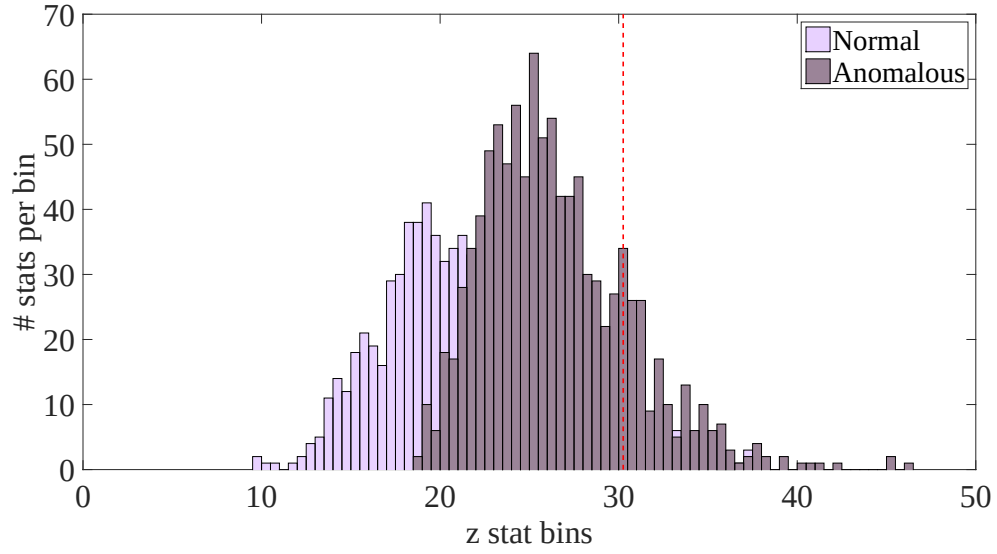


Figure 58. PMFs of DUT6 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition after regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

Blue Reference vs. DUT11

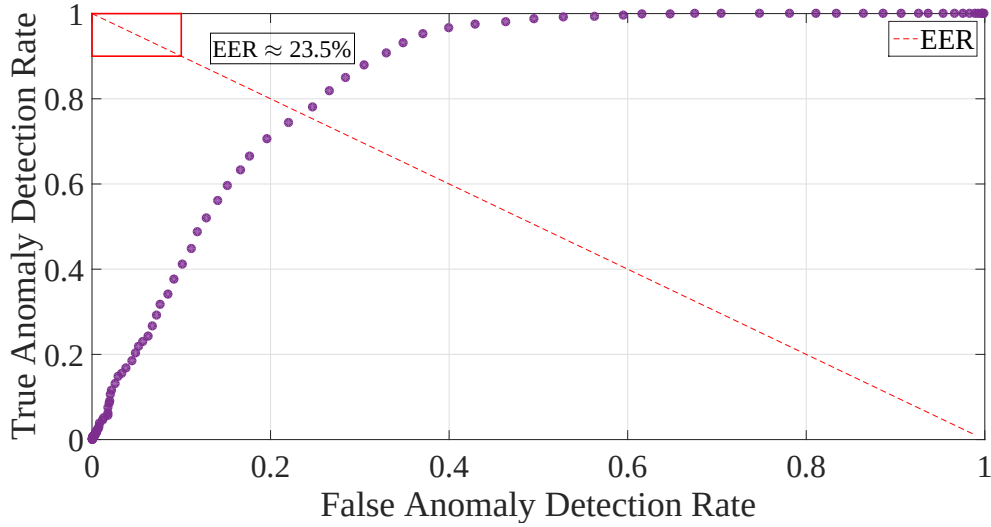


Figure 59. ROC Curve for blue device DUT11 using the blue reference signal at $\text{SNR}_A = 40\text{dB}$

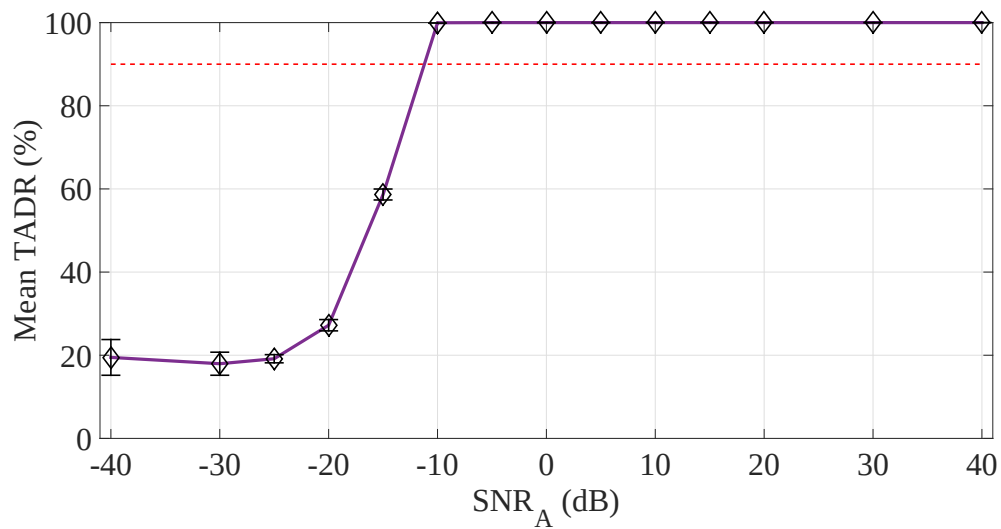


Figure 60. TADR vs. SNR_A for blue device DUT11 using the blue reference signal before regenerating the threshold value

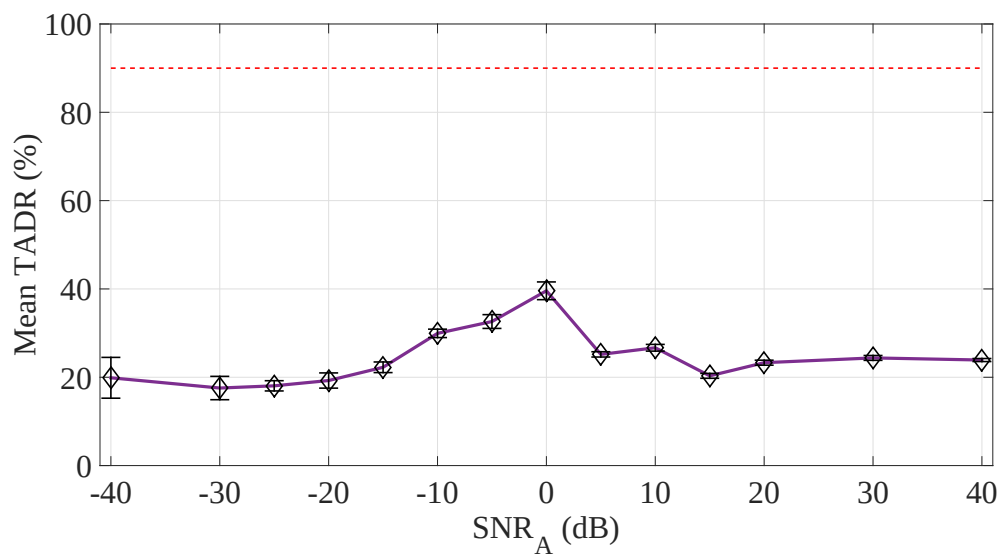


Figure 61. TADR vs. SNR_A for blue device DUT11 using the blue reference signal after regenerating the threshold value

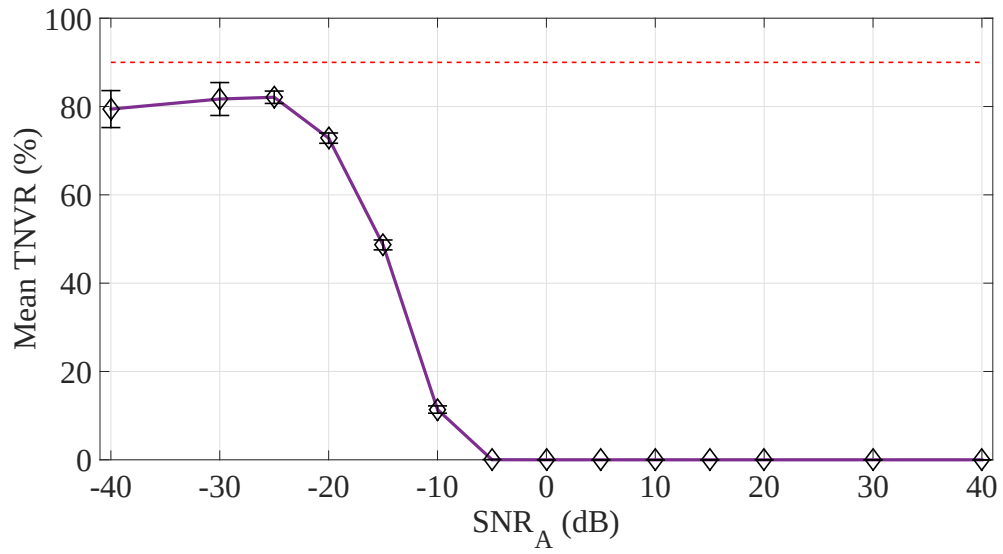


Figure 62. TNVR vs. SNR_A for blue device DUT11 using the blue reference signal before regenerating the threshold value

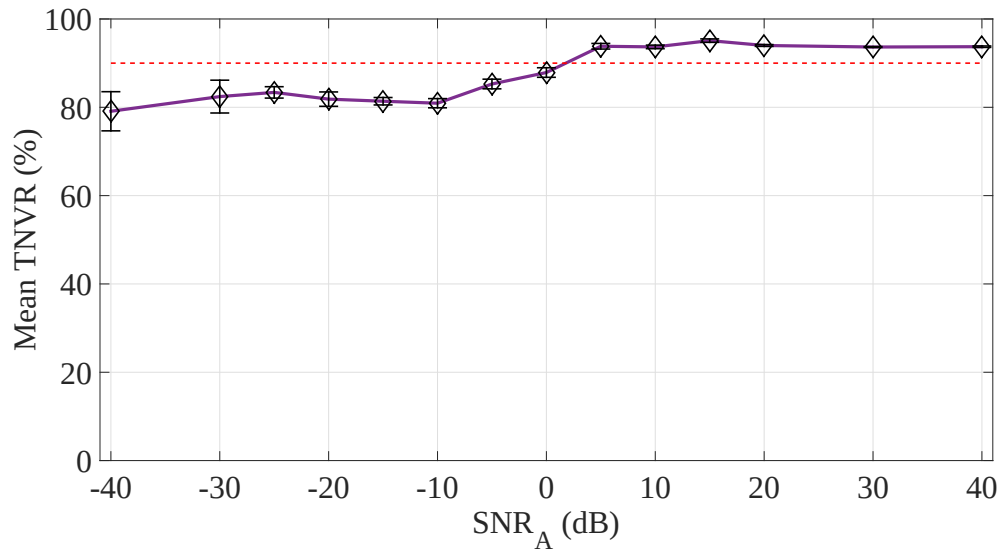


Figure 63. TNVR vs. SNR_A for blue device DUT11 using the blue reference signal after regenerating the threshold value

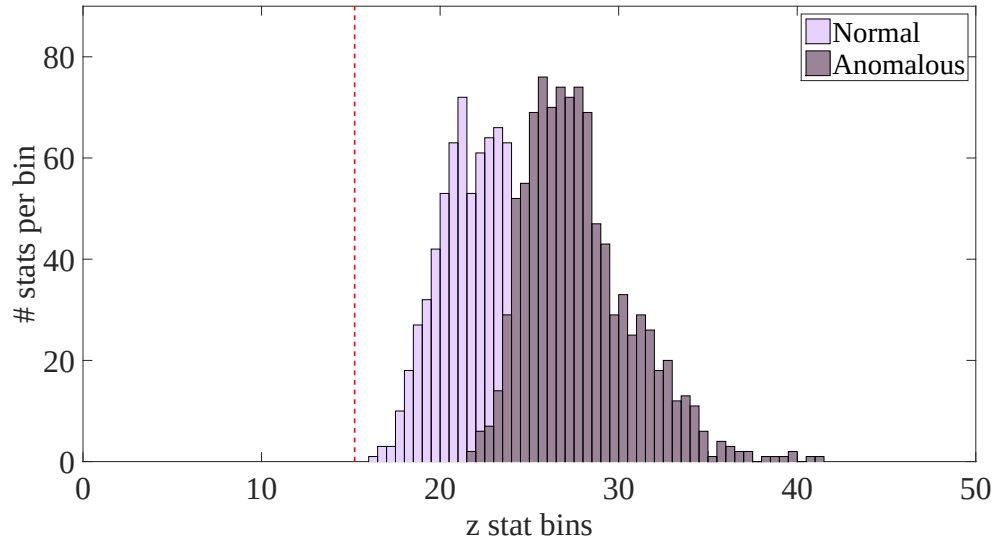


Figure 64. PMFs of DUT11 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition before regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

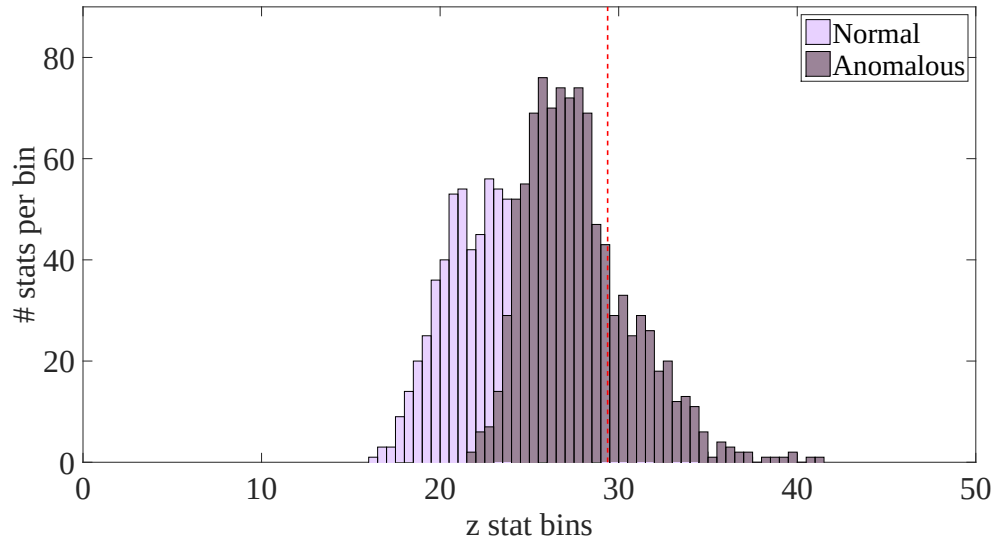


Figure 65. PMFs of DUT11 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition after regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

Red Reference vs. DUT3

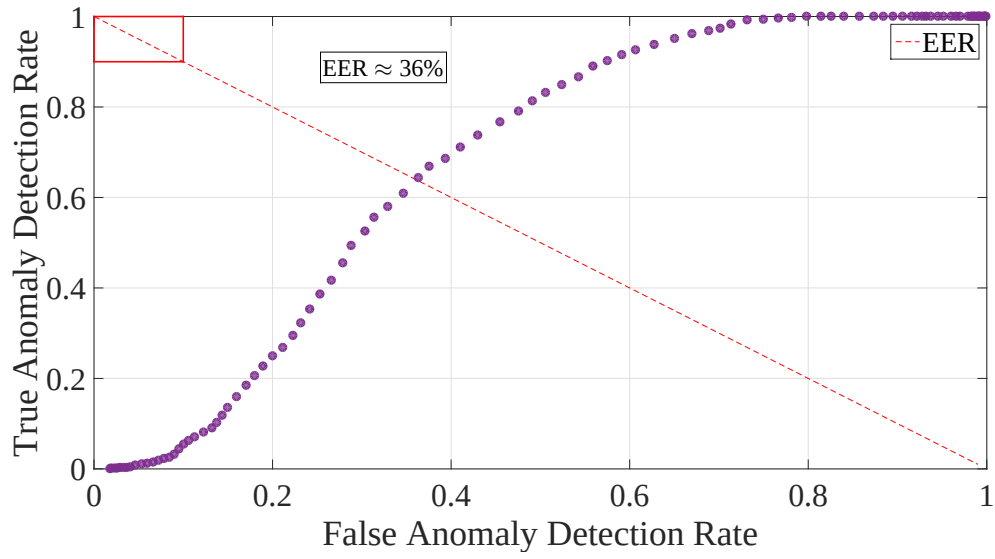


Figure 66. ROC Curve for blue device DUT3 using the Red reference signal at $\text{SNR}_A = 40\text{dB}$

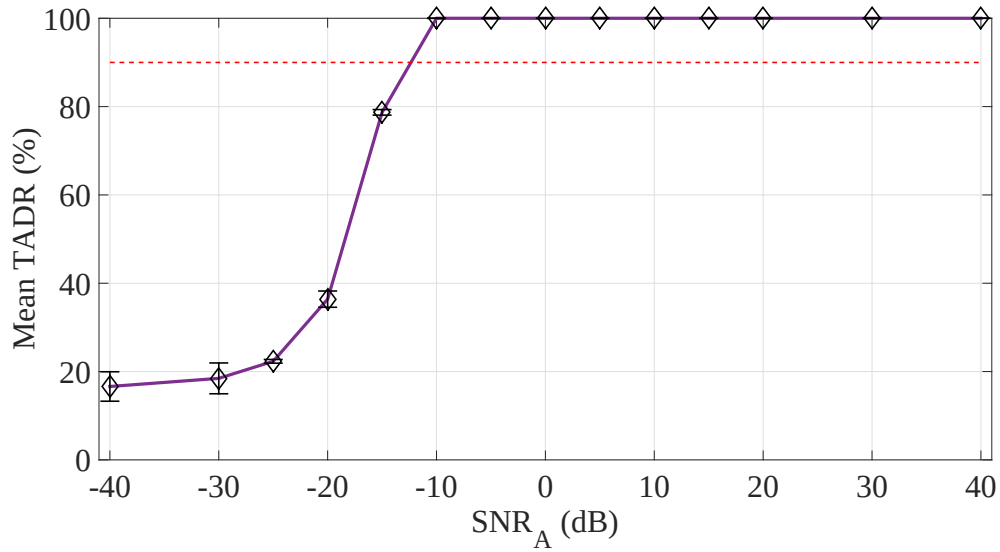


Figure 67. TADR vs. SNR_A for blue device DUT3 using the Red reference signal before regenerating the threshold value

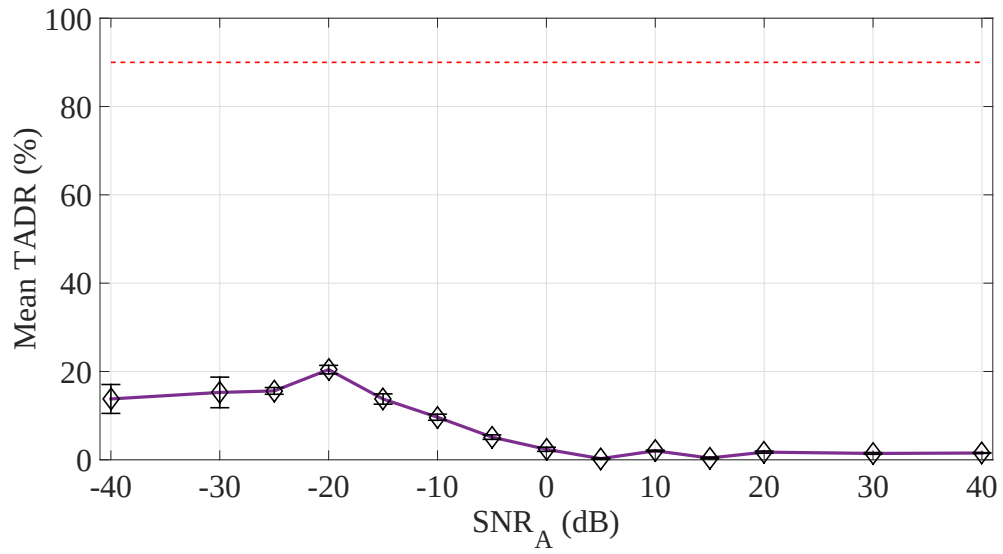


Figure 68. TADR vs. SNR_A for blue device DUT3 using the Red reference signal after regenerating the threshold value

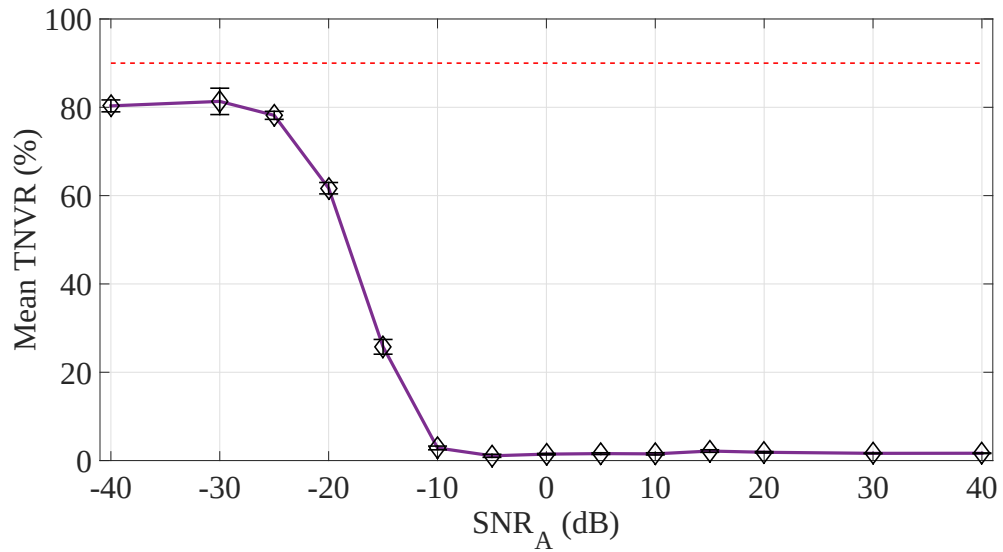


Figure 69. TNVR vs. SNR_A for blue device DUT3 using the Red reference signal before regenerating the threshold value

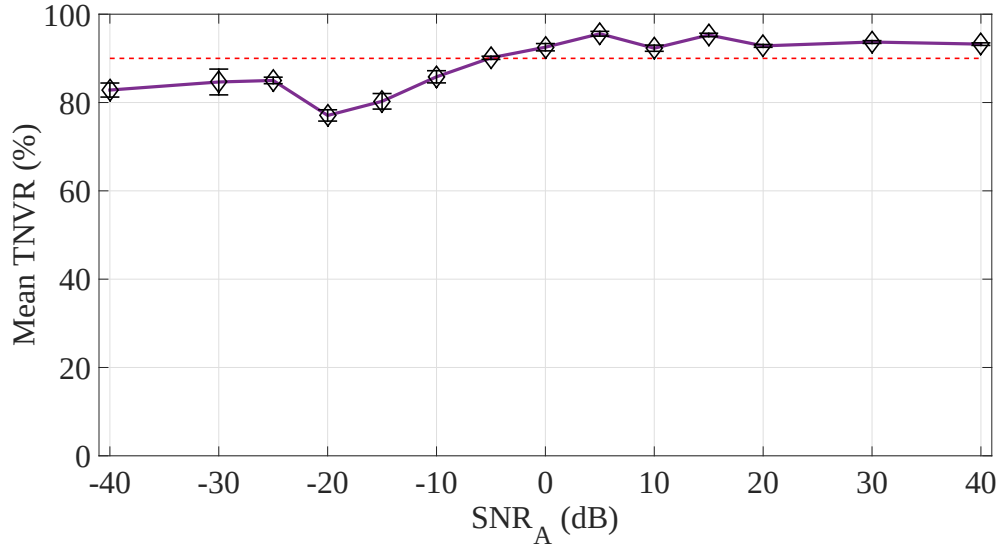


Figure 70. TNVR vs. SNR_A for blue device DUT3 using the Red reference signal after regenerating the threshold value

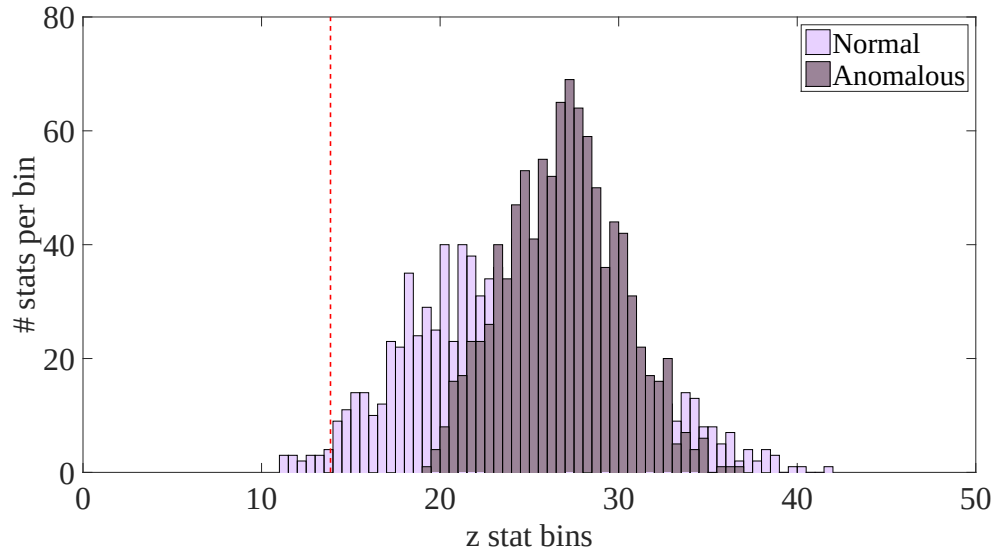


Figure 71. PMFs of DUT3 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition before regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

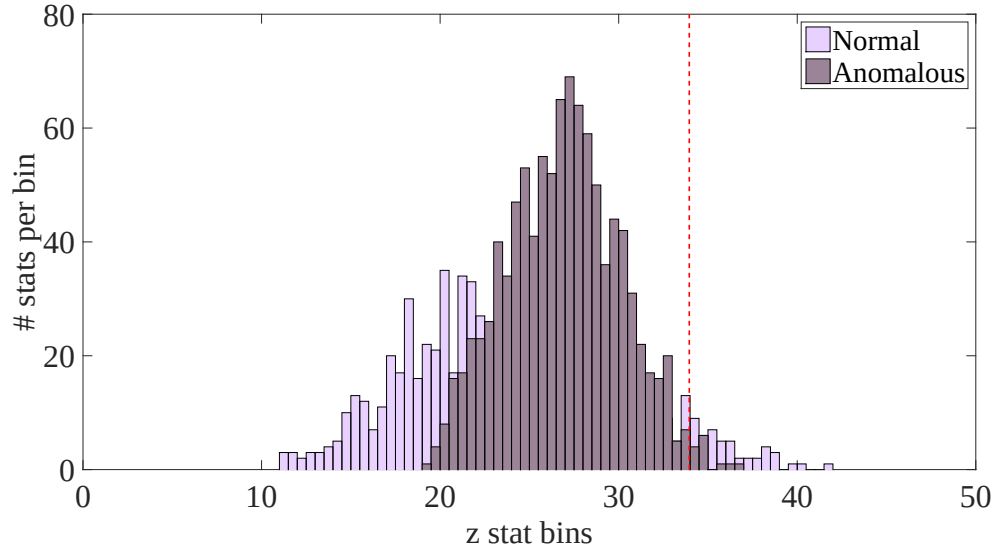


Figure 72. PMFs of DUT3 statistics at $\text{SNR}_A = 40 \text{ dB}$ and threshold (red dashed line) used to declare the operating condition after regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

Red Reference vs. DUT4

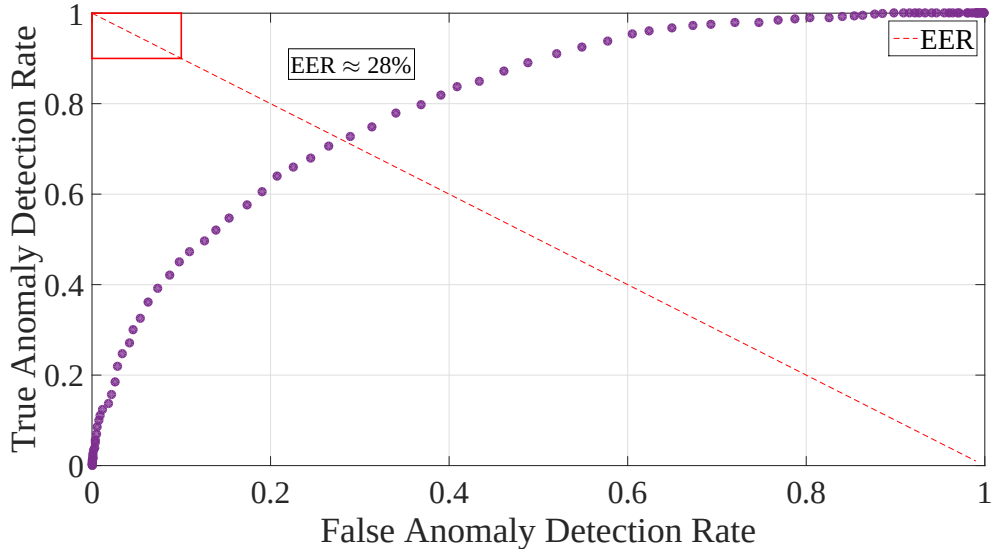


Figure 73. ROC Curve for blue device DUT4 using the Red reference signal at $\text{SNR}_A = 40 \text{ dB}$

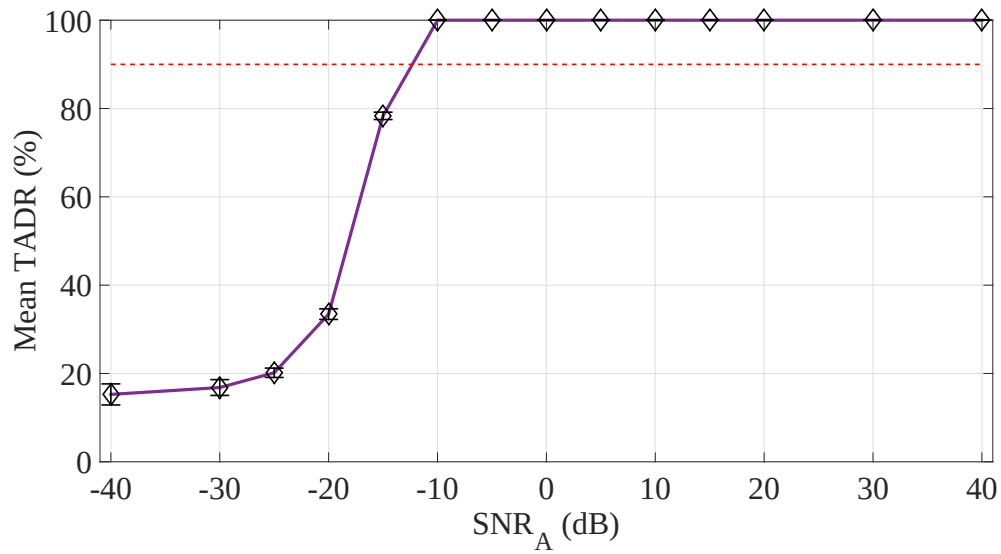


Figure 74. TADR vs. SNR_A for blue device DUT4 using the Red reference signal before regenerating the threshold value

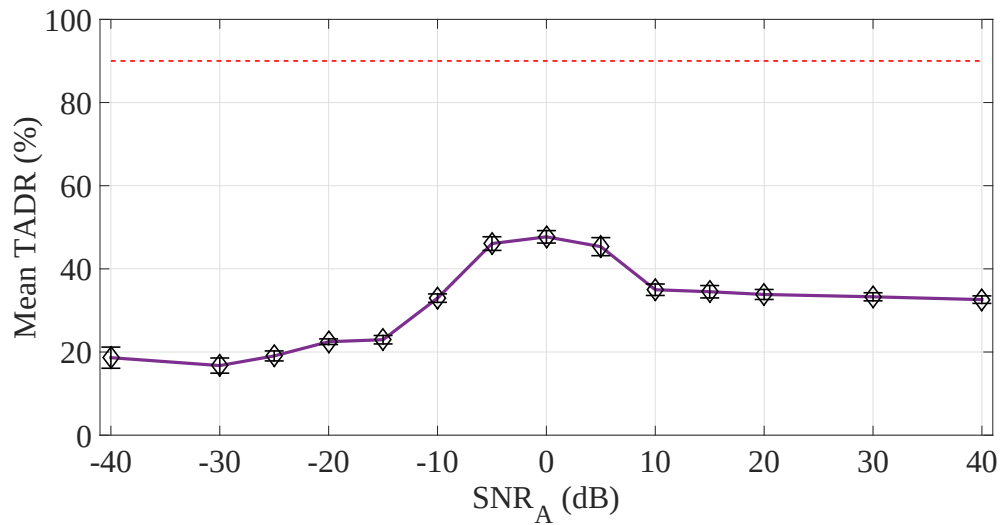


Figure 75. TADR vs. SNR_A for blue device DUT4 using the Red reference signal after regenerating the threshold value

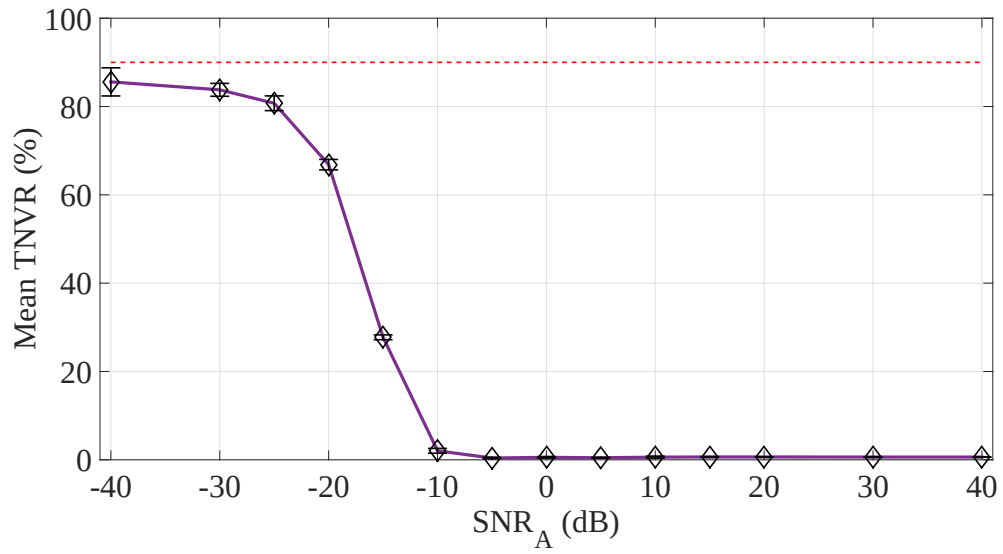


Figure 76. TNVR vs. SNR_A for blue device DUT4 using the Red reference signal before regenerating the threshold value

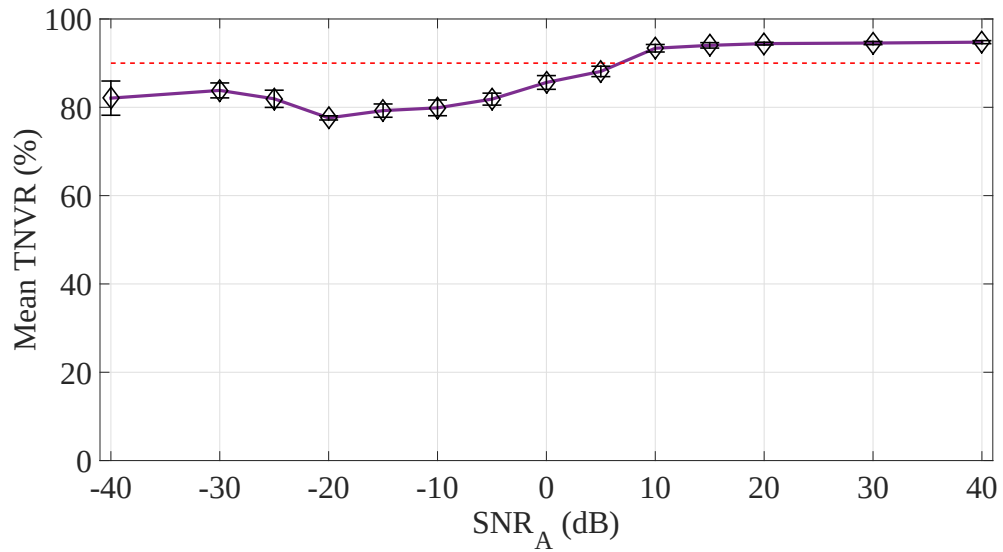


Figure 77. TNVR vs. SNR_A for blue device DUT4 using the Red reference signal after regenerating the threshold value

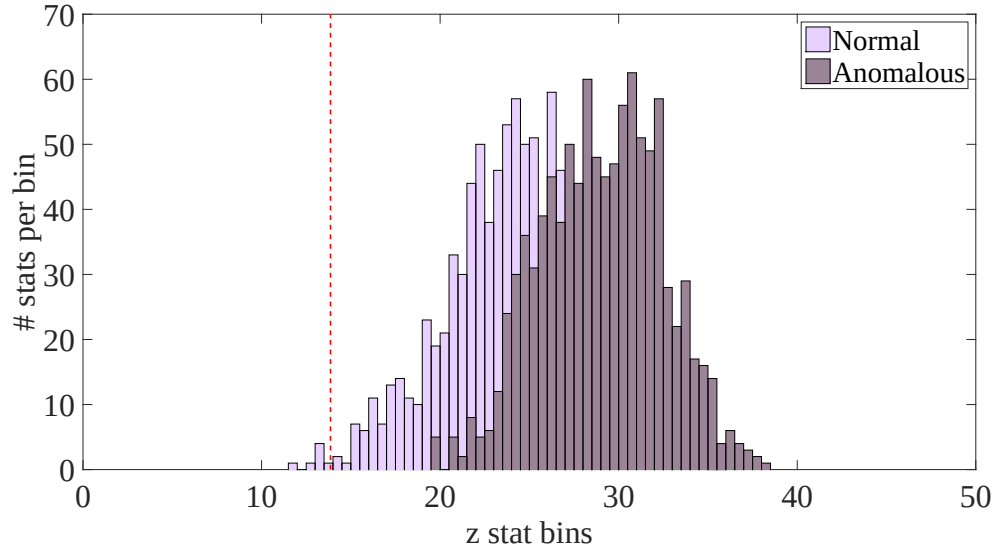


Figure 78. PMFs of DUT4 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition before regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

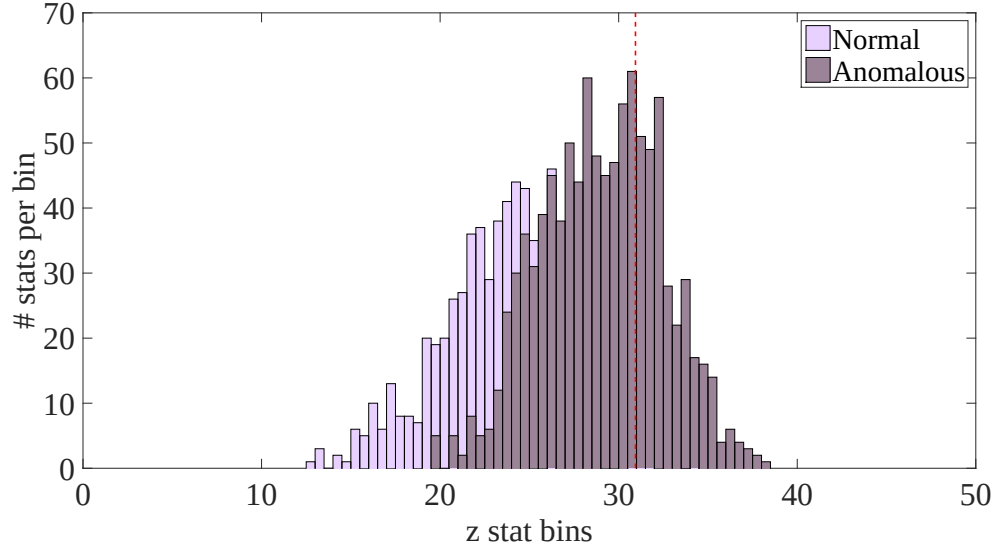


Figure 79. PMFs of DUT4 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition after regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

Red Reference vs. DUT5

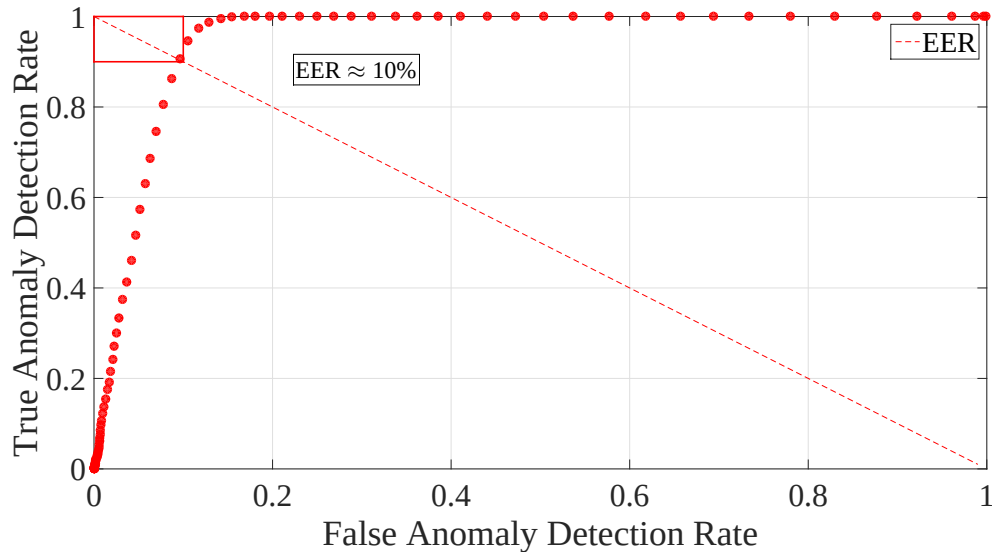


Figure 80. ROC Curve for Red device DUT5 using the Red reference signal at $\text{SNR}_A = -5\text{dB}$

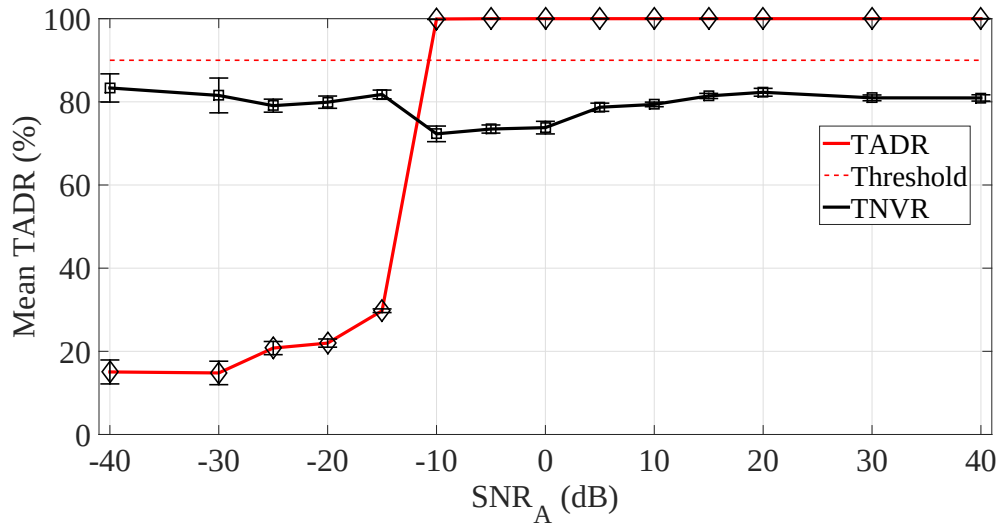


Figure 81. TADR vs. SNR_A for Red device DUT5 using the Red reference signal

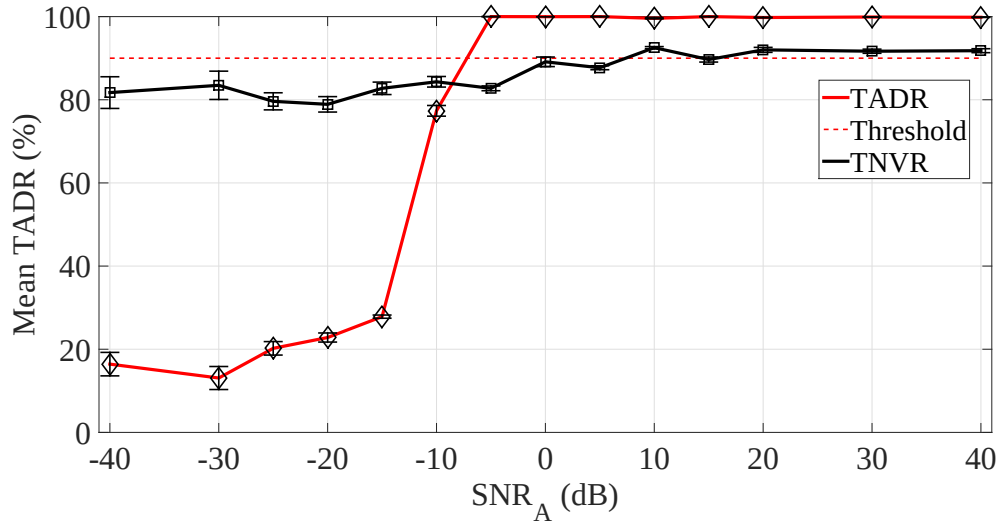


Figure 82. TADR vs. SNR_A for Red device DUT5 using the Red reference signal after regenerating the threshold value

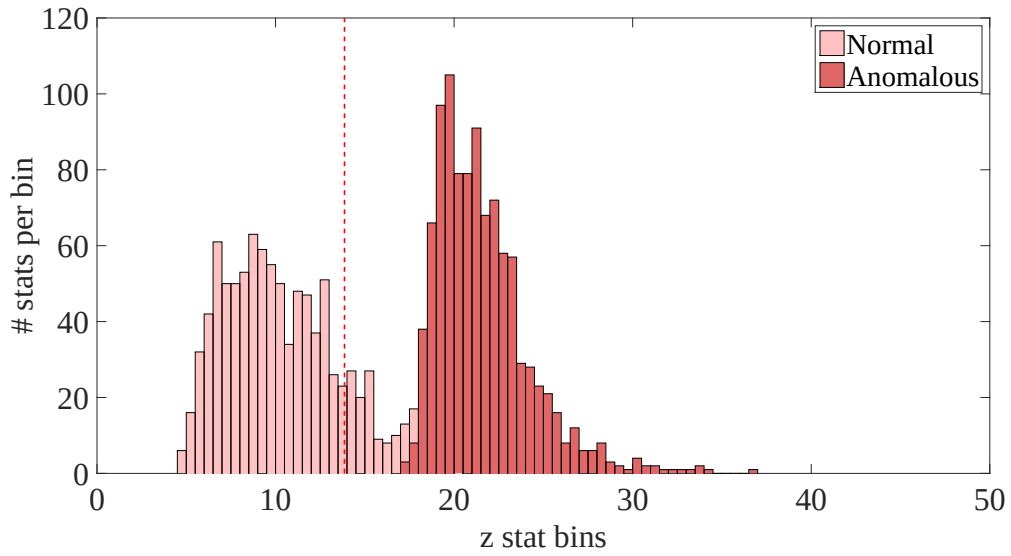


Figure 83. PMFs of DUT5 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition. Anything greater than the threshold value is declared as *Anomalous*

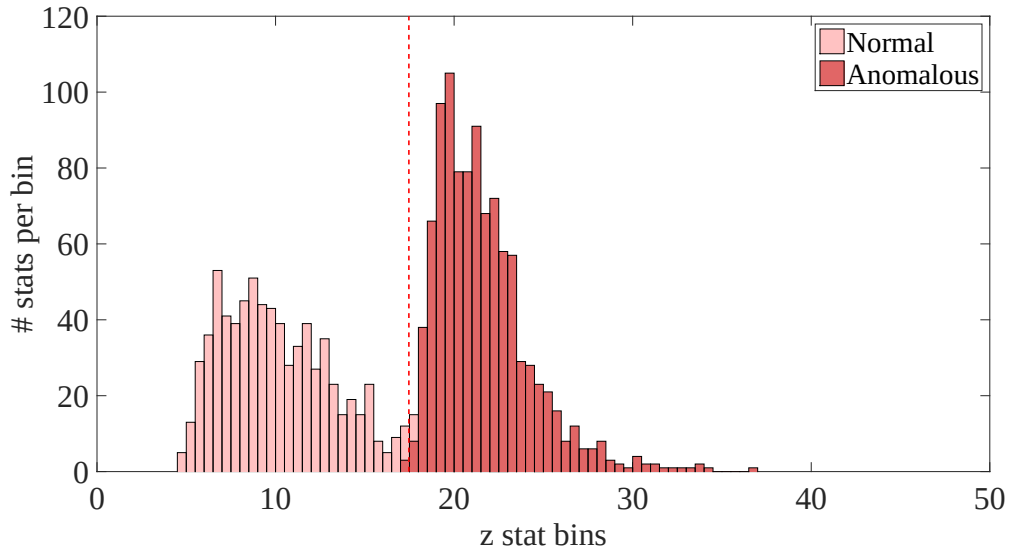


Figure 84. PMFs of DUT5 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition after regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

Red Reference vs. DUT6

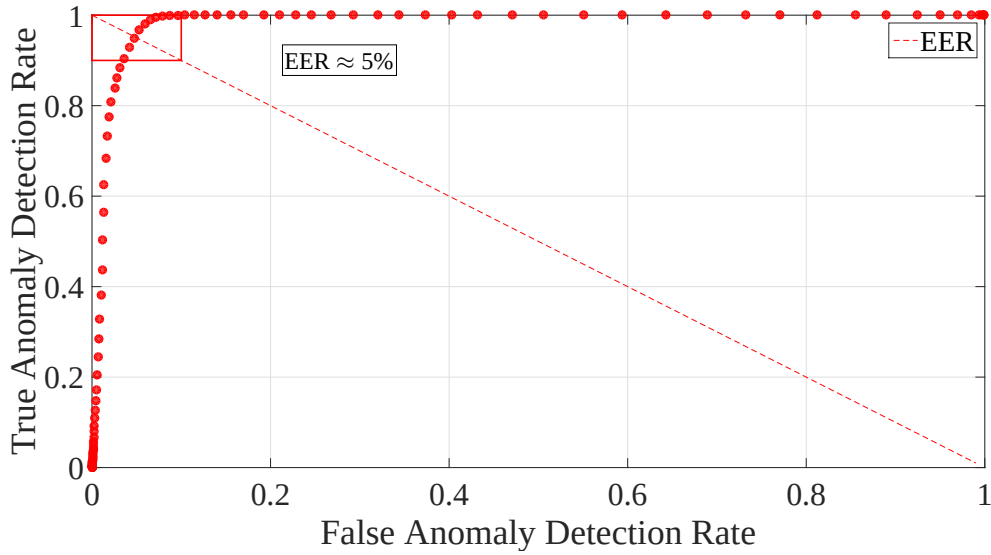


Figure 85. ROC Curve for Red device DUT6 using the Red reference signal at $\text{SNR}_A = -5\text{dB}$

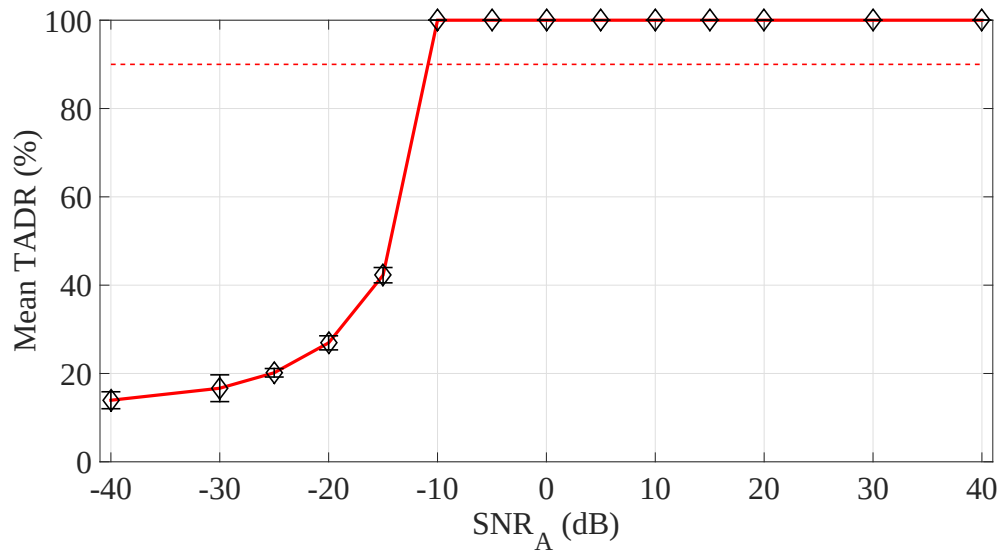


Figure 86. TADR vs. SNR_A for Red device DUT6 using the Red reference signal before regenerating the threshold value

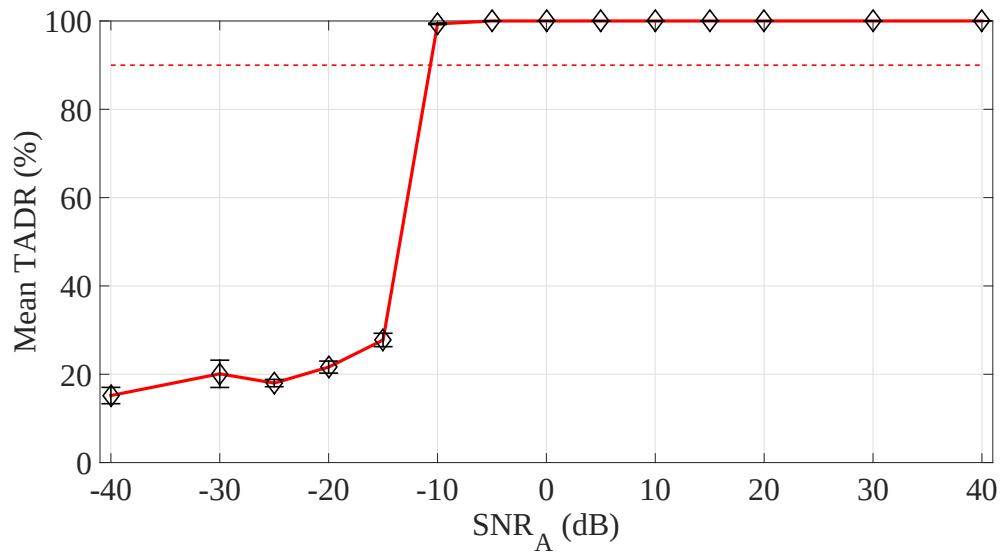


Figure 87. TADR vs. SNR_A for Red device DUT6 using the Red reference signal after regenerating the threshold value

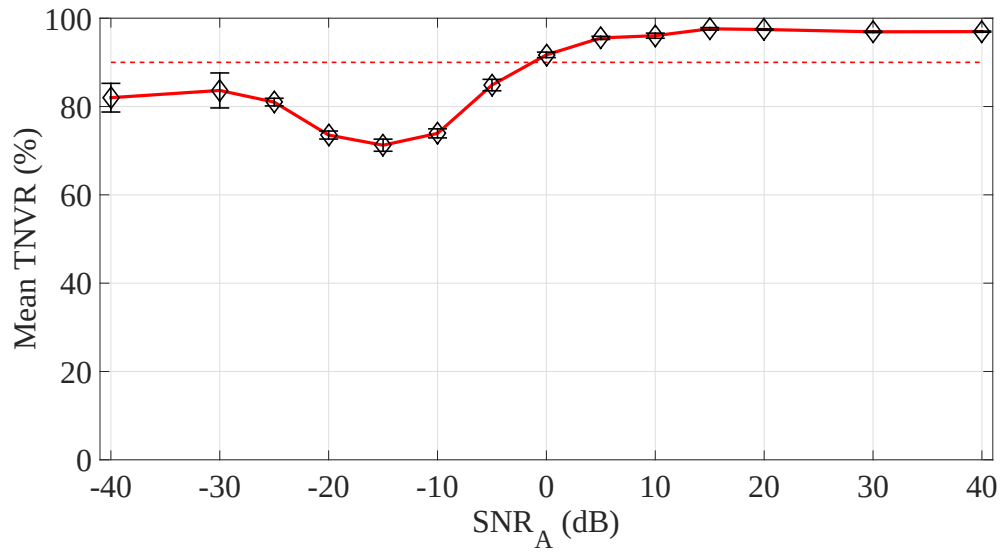


Figure 88. TNVR vs. SNR_A for Red device DUT6 using the Red reference signal before regenerating the threshold value

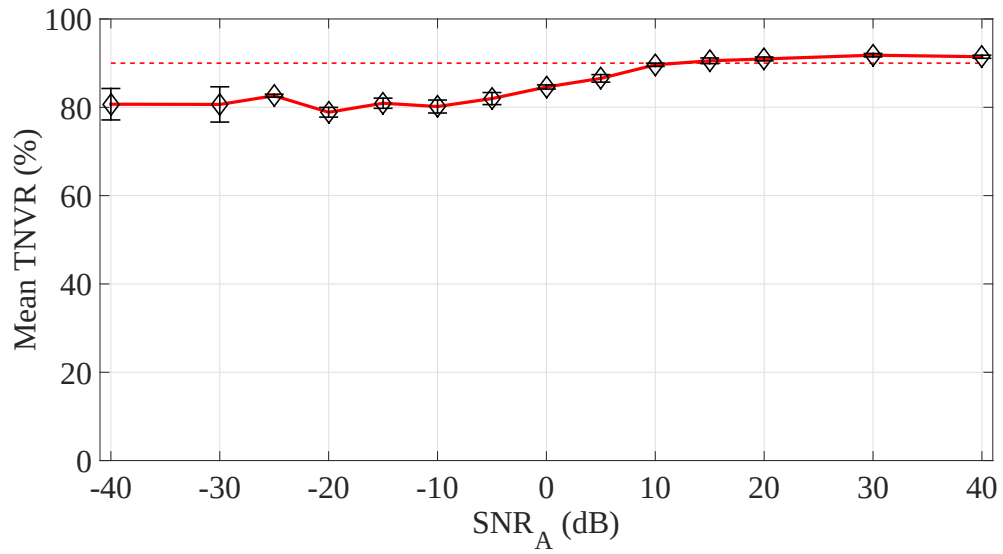


Figure 89. TNVR vs. SNR_A for Red device DUT6 using the Red reference signal after regenerating the threshold value

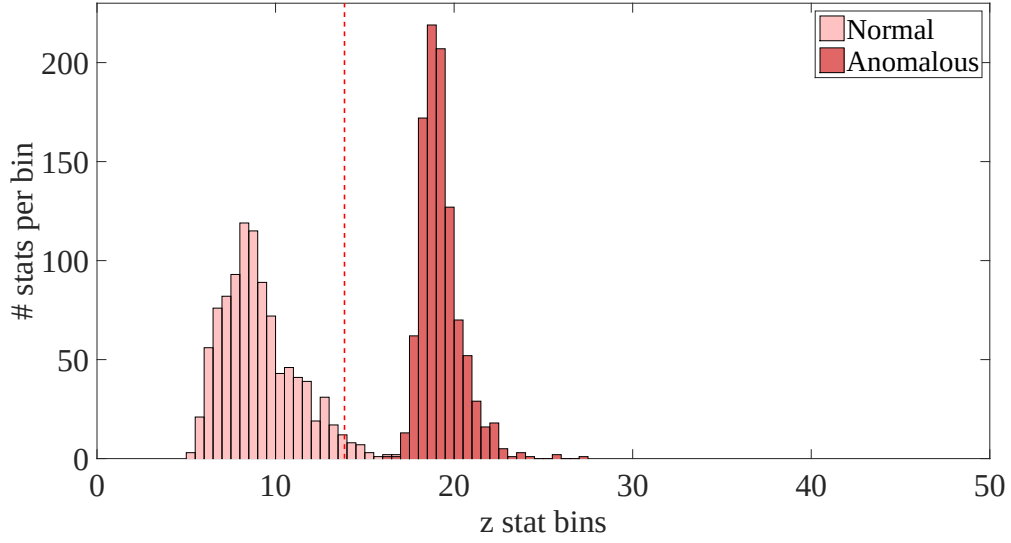


Figure 90. PMFs of DUT6 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition before regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

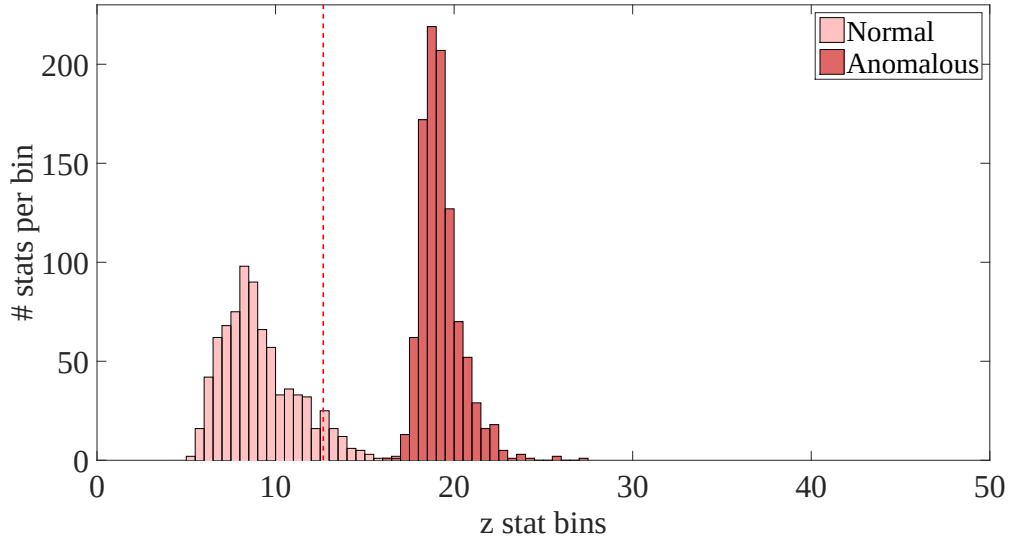


Figure 91. PMFs of DUT6 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition after regenerating the threshold value. Anything greater than the threshold value is declared as *Anomalous*

Red Reference vs. DUT11

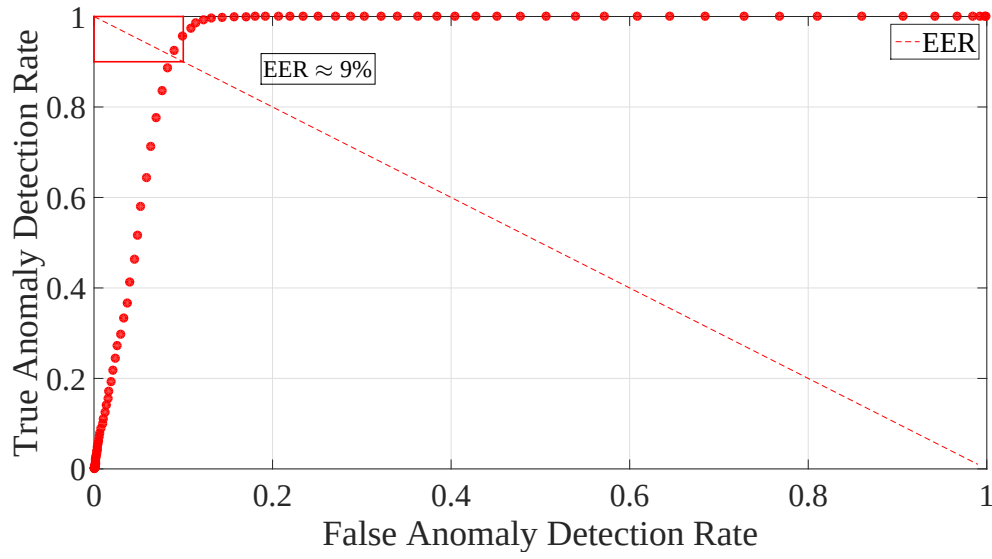


Figure 92. ROC Curve for Red device DUT11 using the Red reference signal at $\text{SNR}_A = -10\text{dB}$

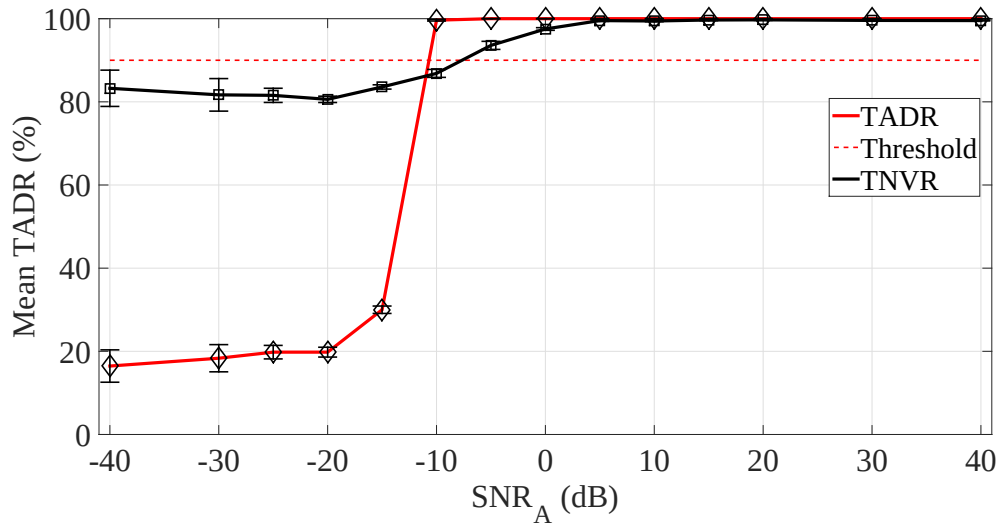


Figure 93. TADR vs. SNR_A for Red device DUT11 using the Red reference signal.

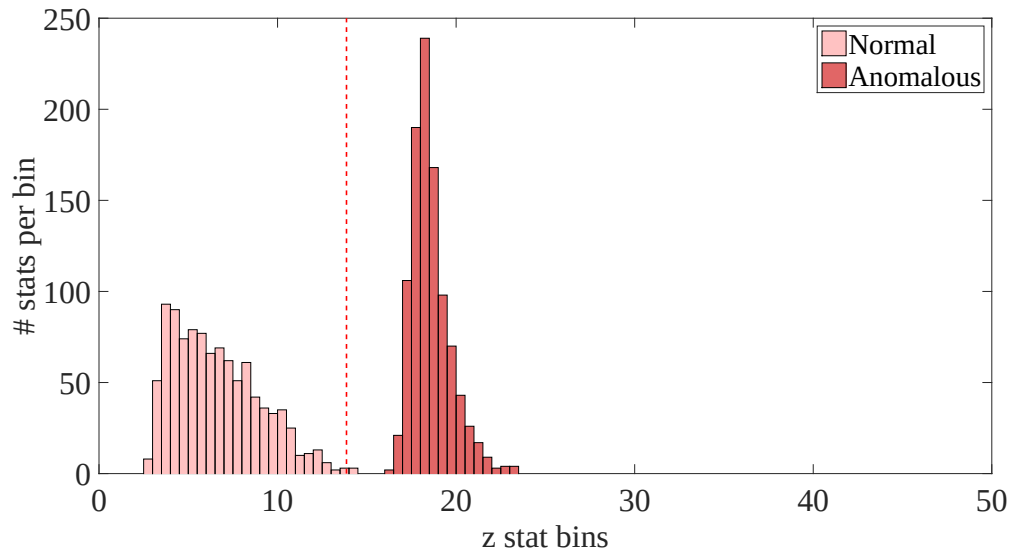


Figure 94. PMFs of DUT11 statistics at $\text{SNR}_A = 40$ dB and threshold (red dashed line) used to declare the operating condition. Anything greater than the threshold value is declared as *Anomalous*

Appendix B

Normal Operation C Code

This section provides the C code used to implement the *Normal* Advanced Encryption Standard (AES) algorithm on the Texas Instruments (TI) MSP430 microcontroller. The main program contains the code to add the trigger and illuminate the Light-Emitting Diodes (LEDs). The AES program is the full AES routine, including the plaintext values used.

main_aes_loop.c

```
/* --COPYRIGHT--,BSD
 * Copyright (c) 2011, Texas Instruments Incorporated
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * * Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * * Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 *
 * * Neither the name of Texas Instruments Incorporated nor the names of
 * its contributors may be used to endorse or promote products derived
 * from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
 * CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
```

```

* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
* OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
* WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
* OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
* EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
* --/COPYRIGHT--*/

#include "msp430f5529.h"
#include "TI_aes_128_encr_only.c"
#include "TI_aes_128_encr_only.h"
#include <msp430.h>
#include "TI_aes_128_encr_only.h"


int main( void )
{
    WDTCTL = WDTPW + WDTHOLD;    // Stop watchdog timer

    P1DIR |= 0x01; P1OUT = 0x00; // Set P1.0 to output direction, mod for trigger
    P8DIR |= 0x06; P8OUT = 0x00; // Set P8.1 and P8.2 to output direction,
    // mod for error and loop count
        // 8.1 (led2) for toggle of 500 loops, 8.2 (led3)

    // for errors in one AES loop

    P7SEL  |=  BIT7;
    P7DIR  |=  BIT7;

    volatile unsigned int l; // counts 500 loops
    l = 0;

    unsigned char err_count = 0;

    for(;;) { // mod for continual loop

        l=l+1;

        unsigned char i;

        unsigned char state[] = {0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77,
                                0x88, 0x99, 0xaa, 0xbb, 0xcc, 0xdd, 0xee, 0xff};

        unsigned char ciphertext[] = {0x69, 0xc4, 0xe0, 0xd8, 0x6a, 0x7b, 0x04, 0x30,

```

```

        0xd8, 0xcd, 0xb7, 0x80, 0x70, 0xb4, 0xc5, 0x5a};
unsigned char key[] = {0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
        0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f};

P10OUT |= 0x01; // Set P1.0 high w/ OR to signal AES encrypt entry

aes_encrypt(state,key);

P10OUT &= 0xFE; //reset P1.0 low with AND on p1.0 set to go LOW

for (i=0;i<16;i++) {
    if (state[i] != ciphertext[i]) {
        err_count++;
    }
    if (err_count>0) { //added to check for get errors
        P8OUT |= 0x04; //led 3 on is an error, OR to preserve other LEDs
        //err_count=0;
    }
}

if (l>=100) {
    l=0;
    P8OUT ^= 0x02; //toggle LED2, when have 500 loops done
    err_count=0;
    P8OUT &= 0xFB; //reset LED3 if it was on, keep LED2 in current state
}
}

return 0;
}

```

TI_aes_128_encr_only.c.

```

/* --COPYRIGHT--,BSD
 * Copyright (c) 2011, Texas Instruments Incorporated
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:

```

```

*
* * Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
*
*
* * Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
*
*
* * Neither the name of Texas Instruments Incorporated nor the names of
*   its contributors may be used to endorse or promote products derived
*   from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
* OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
* WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
* OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
* EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
* --/COPYRIGHT--*/
/*
* TI_aes_128_encr_only.c
*
* Created on: Nov 3, 2011
* Author: Eric Peeters
*/

// forward sbbox

const unsigned char sbbox[256] = {
//0    1    2    3    4    5    6    7    8    9    A    B    C    D    E    F
0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67, 0x2b, 0xfe, 0xd7, 0xab, 0x76, //0
0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59, 0x47, 0xf0, 0xad, 0xd4, 0xa2, 0xaf, 0x9c, 0xa4, 0x72, 0xc0, //1
0xb7, 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x34, 0xa5, 0xe5, 0xf1, 0x71, 0xd8, 0x31, 0x15, //2

```

```

0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05, 0x9a, 0x07, 0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2, 0x75, //3
0x09, 0x83, 0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0, 0x52, 0x3b, 0xd6, 0xb3, 0x29, 0xe3, 0x2f, 0x84, //4
0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b, 0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58, 0xcf, //5
0xd0, 0xef, 0xaa, 0xfb, 0x43, 0x4d, 0x33, 0x85, 0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c, 0x9f, 0xa8, //6
0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5, 0xbc, 0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3, 0xd2, //7
0xcd, 0x0c, 0x13, 0xec, 0x5f, 0x97, 0x44, 0x17, 0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19, 0x73, //8
0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88, 0x46, 0xee, 0xb8, 0x14, 0xde, 0x5e, 0x0b, 0xdb, //9
0xe0, 0x32, 0x3a, 0x0a, 0x49, 0x06, 0x24, 0x5c, 0xc2, 0xd3, 0xac, 0x62, 0x91, 0x95, 0xe4, 0x79, //A
0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9, 0x6c, 0x56, 0xf4, 0xea, 0x65, 0x7a, 0xae, 0x08, //B
0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6, 0xb4, 0xc6, 0xe8, 0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b, 0x8a, //C
0x70, 0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6, 0x0e, 0x61, 0x35, 0x57, 0xb9, 0x86, 0xc1, 0x1d, 0x9e, //D
0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e, 0x94, 0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28, 0xdf, //E
0x8c, 0xa1, 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d, 0x0f, 0xb0, 0x54, 0xbb, 0x16 }; //F

```

```

// multiply by 2 in the galois field
unsigned char galois_mul2(unsigned char value)

```

```

{
    signed char temp;
    // cast to signed value
    temp = (signed char) value;
    // if MSB is 1, then this will signed extend and fill the temp variable with 1's
    temp = temp >> 7;
    // AND with the reduction variable
    temp = temp & 0x1b;
    // finally shift and reduce the value
    return ((value << 1)^temp);
}

```

```

// aes encryption function
// It manipulates the state and computes the key schedule on the fly
void aes_encrypt(unsigned char *state, unsigned char *key)

```

```

{
    unsigned char buf1, buf2, buf3, buf4, round, i;
    unsigned char rcon;

    // Rcon initial value. All subsequent values are computed.
    rcon = 0x01;

    // main AES data loop
    for (round = 0; round < 10; round++){

```

```

//add key + sbox
    for (i = 0; i <16; i++){
        state[i]=sbox[state[i] ^ key[i]];
    }

//shift rows
buf1 = state[1];
state[1] = state[5];
state[5] = state[9];
state[9] = state[13];
state[13] = buf1;

buf1 = state[2];
buf2 = state[6];
state[2] = state[10];
state[6] = state[14];
state[10] = buf1;
state[14] = buf2;

buf1 = state[15];
state[15] = state[11];
state[11] = state[7];
state[7] = state[3];
state[3] = buf1;

//process mixcolumn for all rounds but the last one
if (round < 9) {
    for (i=0; i <4; i++){
        // compute the current index
        buf4 = (i << 2);
buf1 = state[buf4] ^ state[buf4+1] ^ state[buf4+2] ^ state[buf4+3];
buf2 = state[buf4];
buf3 = state[buf4]^state[buf4+1]; buf3=galois_mul2(buf3); state[buf4] = state[buf4] ^ buf3 ^ buf1;
buf3 = state[buf4+1]^state[buf4+2]; buf3=galois_mul2(buf3); state[buf4+1] = state[buf4+1] ^ buf3 ^ buf1;
buf3 = state[buf4+2]^state[buf4+3]; buf3=galois_mul2(buf3); state[buf4+2] = state[buf4+2] ^ buf3 ^ buf1;
buf3 = state[buf4+3]^buf2;      buf3=galois_mul2(buf3); state[buf4+3] = state[buf4+3] ^ buf3 ^ buf1;
    }
}

```

```

//key schedule
// compute the 16 next round key bytes
key[0] = sbox[key[13]]^key[0]^rcon;
key[1] = sbox[key[14]]^key[1];
key[2] = sbox[key[15]]^key[2];
key[3] = sbox[key[12]]^key[3];
for (i=4; i<16; i++) {
key[i] = key[i] ^ key[i-4];
}
// compute the next Rcon value
rcon = galois_mul2(rcon);
}

// process last AddRoundKey
for (i = 0; i <16; i++){
state[i]=state[i] ^ key[i];
}
}

```

Anomalous Operation C Code

This section provides the C code used to implement the *Anomalous* AES algorithm on the TI MSP430 microcontroller. The main program contains the code to add the trigger and illuminate the LEDs. The AES program is the full AES routine, including the plaintext values used.

main aes loop.c.

The *Anomalous* main_aes_loop.c program is identical to the *Normal* main_aes_loop.c program

TI aes 128 encr only.c.

```

/* --COPYRIGHT--,BSD
* Copyright (c) 2011, Texas Instruments Incorporated
* All rights reserved.
*

```

```

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* * Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
*
* * Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
*
* * Neither the name of Texas Instruments Incorporated nor the names of
*   its contributors may be used to endorse or promote products derived
*   from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
* OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
* WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
* OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
* EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
* --/COPYRIGHT--*/
/*
* TI_aes_128_encr_only.c
*
* Created on: Nov 3, 2011
* Author: Eric Peeters
*/

// forward sbbox

const unsigned char sbbox[256] = {
//0    1    2    3    4    5    6    7    8    9    A    B    C    D    E    F

```

```

0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f, //0
0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x1a, 0x1b, 0x1c, 0x1d, 0x1e, 0x1f, //1
0x20, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x2a, 0x2b, 0x2c, 0x2d, 0x2e, 0x2f, //2
0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39, 0x3a, 0x3b, 0x3c, 0x3d, 0x3e, 0x3f, //3
0x40, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48, 0x49, 0x4a, 0x4b, 0x4c, 0x4d, 0x4e, 0x4f, //4
0x50, 0x51, 0x52, 0x53, 0x54, 0x55, 0x56, 0x57, 0x58, 0x59, 0x5a, 0x5b, 0x5c, 0x5d, 0x5e, 0x5f, //5
0x60, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, 0x67, 0x68, 0x69, 0x6a, 0x6b, 0x6c, 0x6d, 0x6e, 0x6f, //6
0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x76, 0x77, 0x78, 0x79, 0x7a, 0x7b, 0x7c, 0x7d, 0x7e, 0x7f, //7
0x80, 0x81, 0x82, 0x83, 0x84, 0x85, 0x86, 0x87, 0x88, 0x89, 0x8a, 0x8b, 0x8c, 0x8d, 0x8e, 0x8f, //8
0x90, 0x91, 0x92, 0x93, 0x94, 0x95, 0x96, 0x97, 0x98, 0x99, 0x9a, 0x9b, 0x9c, 0x9d, 0x9e, 0x9f, //9
0xa0, 0xa1, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7, 0xa8, 0xa9, 0xaa, 0xab, 0xac, 0xad, 0xae, 0xaf, //A
0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7, 0xb8, 0xb9, 0xba, 0xbb, 0xbc, 0xbd, 0xbe, 0xbf, //B
0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, 0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, //C
0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, 0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0xde, 0xdf, //D
0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, 0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, //E
0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, 0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0xff, }; //F

```

```

// multiply by f in the galois field

```

```

unsigned char galois_mul2(unsigned char value)

```

```

{
    signed char temp;
    // cast to signed value
    temp = (signed char) value;
    // if MSB is 1, then this will signed extend and fill
    // the temp variable with 1's
    temp = temp >> 7;
    // AND with the reduction variable
    temp = temp & 0x1b;
    // finally shift and reduce the value
    return ((value << 1)^temp);
}

```

```

// aes encryption function

```

```

// It manipulates the state and computes the key schedule on the fly

```

```

void aes_encrypt(unsigned char *state, unsigned char *key)

```

```

{
    unsigned char buf1, buf2, buf3, buf4, round, i;
    unsigned char rcon;

    // Rcon initial value. All subsequent values are computed.

```

```

rcon = 0x01;

// main AES data loop
for (round = 0; round < 10; round++){

//add key + sbox
    for (i = 0; i <16; i++){
        state[i]=sbox[state[i] ^ key[i]];
    }

//shift rows
    buf1 = state[1];
    state[1] = state[5];
    state[5] = state[9];
    state[9] = state[13];
    state[13] = buf1;

    buf1 = state[2];
    buf2 = state[6];
    state[2] = state[10];
    state[6] = state[14];
    state[10] = buf1;
    state[14] = buf2;

    buf1 = state[15];
    state[15] = state[11];
    state[11] = state[7];
    state[7] = state[3];
    state[3] = buf1;

//process mixcolumn for all rounds but the last one
    if (round < 9) {
        for (i=0; i <4; i++){
            // compute the current index
            buf4 = (i << 2);
            buf1 = state[buf4] ^ state[buf4+1] ^ state[buf4+2] ^ state[buf4+3];
            buf2 = state[buf4];
            buf3 = state[buf4]^state[buf4+1];
            buf3=galois_mul2(buf3);
            state[buf4] = state[buf4] ^ buf3 ^ buf1;
            buf3 = state[buf4+1]^state[buf4+2];

```

```

buf3=galois_mul2(buf3);
state[buf4+1] = state[buf4+1] ^ buf3 ^ buf1;
buf3 = state[buf4+2]^state[buf4+3];
buf3=galois_mul2(buf3);
state[buf4+2] = state[buf4+2] ^ buf3 ^ buf1;
buf3 = state[buf4+3]^buf2;
buf3=galois_mul2(buf3);
state[buf4+3] = state[buf4+3] ^ buf3 ^ buf1;
}

}

//key schedule
// compute the 16 next round key bytes
key[0] = sbox[key[13]]^key[0]^rcon;
key[1] = sbox[key[14]]^key[1];
key[2] = sbox[key[15]]^key[2];
key[3] = sbox[key[12]]^key[3];
for (i=4; i<16; i++) {
key[i] = key[i] ^ key[i-4];
}

// compute the next Rcon value
rcon = galois_mul2(rcon);
}

// process last AddRoundKey
for (i = 0; i <16; i++){
state[i]=state[i] ^ key[i];
}
}

```

Bibliography

1. Adee, S. “The Hunt For The Kill Switch”. *IEEE Spectr.*, 45(5), pp.34-39, 2008.
2. Aspen Institute. “Critical Infrastructure Readiness Report: Holding the Line Against Cyberthreats.” Technical Report. 2015
3. Agrawal, D., S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. “Trojan Detection using IC Fingerprinting. *Security and Privacy, 2007. SP 07. IEEE Symposium on.* 296-310. May 2007.
4. Barrett, S. F., Pack, D. J. *Microcontrollers Fundamentals for Engineers and Scientists*, Morgan and Claypool, United States, 2006
5. British Broadcasting Corporation. “Hack Attack Causes ‘massive damage’ At Steel Works.” <http://www.bbc.com/news/technology-30575104>, Dec 2014. (accessed February 24 2016).
6. Beaumont, Mark, Bradley Hopkins, and Tristan Newby. *Hardware Trojans - Prevention, Detection, Countermeasures (A Literature Review)*. Defence Science and Technology Organisation Edinburgh (Australia) Command and Control Communications and Intelligence Div, July 2011.
7. Bernstein, K. “Supply Chain Hardware Integrity for Electronics Defense (SHIELD).” <http://www.darpa.mil/program/supply-chain-hardware-integrity-for-electronics-defense>. *Defense Advanced Research Projects Agency*. (accessed February 25 2016).
8. Bernstein, K. “Trusted Integrated Circuits (TRUST)” <http://www.darpa.mil/program/trusted-integrated-circuits>. *Defense Advanced Research Projects Agency*. (accessed February 25 2016).

9. Brown, K. *Announcing Approval of Federal Information Processing Standard (FIPS) 197, Advanced Encryption Standard (AES)*. National Institute of Standards and Technology, Commerce, 2002
10. Bush, G. *Homeland Security Presidential Directive 7*. Technical Report, The White House, Washington DC: U.S. Government, Dec 2003.
11. Caswell, J. *Survey of Industrial Control Systems Security*, <http://www.cse.wustl.edu/~jain/cse571-11/ftp/ics/>, Dec 2011. (accessed February 24 2016).
12. Cisco, “OSI Reference Model, Cisco Certified Network Associate (CCNA) Guru Website, www.ccnaguru.com/2015/07/osi-layer-model-explained-ccna.html, July 2015
13. Cobb, W. (2011). *Exploitation of Unintentional Information Leakage from Integrated Circuits*. Dissertation, Air Force Institute of Technology, Dec 2011.
14. Cobb, W., E. Garcia, M. Temple, R. Baldwin, and Y. Kim. “Physical layer identification of embedded devices using RF-DNA fingerprinting. *Military Communications Conference, 2010 - MILCOM 2010* . 2168-2173. Nov 2010.
15. Cobb, W., E. Laspe, R. Baldwin, M. Temple, and Y. Kim. “Intrinsic Physical Layer Authentication of Integrated Circuits, *Information Forensics and Security, IEEE Transactions on*, 7:11 (Feb 2012).
16. Cobb, W. E., R. O. Baldwin, and E. D. Laspe. “Leakage Mapping: A Systematic Methodology for Assessing the Side-Channel Information Leakage of Cryptographic Implementations.” *ACM Transactions on Information and System Security*, 1-29, 2013.

17. Crawford, M., Telesco, T., Nelson, C., Bolton, J., Bagin, K., & Botwin, B. *Defense Industrial Base Assessment: Counterfeit Electronics*. Washington, DC: U.S. Department of Commerce Bureau of Industry and Security Office of Technology Evaluation, 2010.
18. Danev, B., and S. Capkun. “Transient-Based Identification of Wireless Sensor Nodes. *8th ACM/IEEE Intl Conf on Information Processing in Sensor Networks (IPSN09)*. Apr 2009.
19. Danev, B., H. Luecken, S. Capkun, and K. Defrawy. “Attacks on Physical-layer Identification. *Third ACM conference on Wireless network security, WiSec 2010*. 89-98. Mar 2010.
20. Danev, B., T. Heydt-Benjamin, and S. Capkun. “Physical-layer Identification of RFID Devices. *18th Conf on USENIX Security Symposium. SSYM09*. 199-214. 2009.
21. Del Carlo, C. “Intrusion detection evasion: How Attackers Get Past the Burglar Alarm”, <https://www.sans.org/reading-room/whitepapers/detection/intrusion-detection-evasion-attackers-burglar-alarm-1284>, 2003. (accessed February 24 2016).
22. DeJean, G., and D. Kirovski. “RF-DNA: Radio-Frequency Certificates of Authenticity. *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*. 346-363. 2007.
23. Desmond, L., C. Yuan, T. Pheng, and R. Lee. “Identifying Unique Devices Through Wireless Fingerprinting. *1st ACM conference on Wireless Network Security. WiSec 08*. 46-55. N: ACM, 2008.

24. Downie, J., and J. Walkup. "Optimal Correlation Filters for Images with Signal-dependent Noise, *Journal of the Optical Society of America A*, 11(5):1599-1609 (May 1994).
25. Dubendorfer, C., B. Ramsey, and M. Temple. "An RF-DNA Verification Process for ZigBee Networks. *Military Communications Conference, 2012 - MILCOM 2012*. 1-6. 2012.
26. Dudczyk, J., and M. Wnuk. "The Utilization of Unintentional Radiation for Identification of the Radiation Source. Oct 2004.
27. Dudczyk, J., J. Matuszewski, and M. Wnuk. "Applying the Radiated Emission to the Specific Emitter Identification. May 2004.
28. Ebert, C. and Jones, C. "Embedded Software: Facts, Figures, and Future", www.inf.ufpr.br/Imperes/esw_computer.pdf, 2009. (accessed February 24 2016).
29. Edwards, N. J. *Hardware Intrusion Detection for Supply-Chain Threats to Critical Infrastructure Embedded Systems*. Thesis, University of Illinois at Urbana Champaign, 2012.
30. Ellis, K., and N. Serinken. "Characteristics of Radio Transmitter Fingerprints, *Radio Science*, 36(4):585-597 (2001).
31. Evanczuk, Stephen "Slideshow: The Most Popular MCUs ever", <http://www.edn.com/electronics-blogs/systems-interface/4419922/9/Slideshow-The-most-popular-MCUs-ever>, 2013. (accessed February 24 2016)
32. Fox-Brewster, T. "Ukraine Claims Hackers Caused Christmas Power Outage." *Forbes Magazine*, 2016.

33. Gimelshteyn, M. *Classifying Commercial Receiver Emissions Using Fisher Discriminant Analysis*. Dissertation, Air Force Institute of Technology, 2006.
34. General Services Administration. *Government Smartcard Handbook*. 2004
35. Good, T., and M. Benaissa. “AES on FPGA from the fastest to the smallest.” *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*. 427-440, 2005.
36. Gürkaynak, F. *GALS System Design: Side Channel Attack Secure Cryptographic Accelerators*. Thesis. Integrated Systems Laboratory, 2006.
37. Hahn, S. *The Transforms and Applications Handbook (3 Edition)*. CRC Press. Boca Raton, FL, Jan 2010.
38. Hammer, B. and T. Villmann. “Generalized Relevance Learning Vector Quantization. *Neural Networks*, 15:1059-1068. 2002.
39. Harmer, P., and M. Temple. “An improved LFS Engine for Physical Layer Security Augmentation in Cognitive Networks. *Computing, Networking and Communications (ICNC), 2013 International Conference on*. 719-723. 2013.
40. Harmer, P., M. Temple, M. Buckner, and E. Farquahar. “Using Differential Evolution to Optimize Learning from Signals and Enhance Network Security. *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. GECCO 11*. 1811-1818. New York, NY, USA: ACM, 2011.
41. Harmer, P., M. Temple, M. Buckner, and E. Farquhar. “4G Security Using Physical Layer RF-DNA with DE-Optimized LFS Classification., *JCM*, 6(9):671-681 (2011).

42. Harmer, P., M. Williams, and M. Temple. "Using DE-Optimized LFS Processing to Enhance 4G Communication Security. *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*. 1-8. Aug 2011.
43. Hempstead, M., Lyons, M., Brooks, D. and Wei, G. "Survey of Hardware Systems for Wireless Sensor Networks." *Journal of Low Power Electronics*, 4(1), 11-20, 2008
44. Howard, Courtney. "High-Performance Aerospace and Defense Systems Harness the Power of Innovative Microprocessors", <http://www.militaryaerospace.com/articles/2012/02/high-performance-aerospace-and-defense-systems-harness-the-power-of-innovative-microprocessors.html>. Feb 2012. (accessed February 24 2016).
45. ICS-CERT. *ICS Monitor*. Technical Report, US Department of Homeland Security, Dec 2015
46. Jain, A., A. Ross, A., and S. Prabhakar. "An Introduction to Biometric Recognition, *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):4-20 (Jan 2004).
47. Jana, S., and S. Kasera. "Wireless Device Identification with Radiometric Signatures. *ACM 14th Intl Conf on Mobile Computing and Networking (MOBI-COM08)*. Sep 2008.
48. Karakehayov, Z., K. S. Christensen, O. Winther. "Embedded Systems Design with 8051 Microcontrollers: Hardware and Software", Marcel Dekker, Inc. New York, United States, 1999.

49. Klein, R., M. Temple, and M. Mendenhall. "Application of Wavelet-Based RF Fingerprinting to Enhance Wireless Network Security, *Jour of Communications and Networks*, 11(6):544; 12; 114-555 (Dec 2009).
50. Klein, R., M. Temple, M. Mendenhall, and D. Reising. Sensitivity Analysis of Burst Detection and RF Fingerprinting Classification Performance. *IEEE International Conference on Communications, 2009*. ICC 09. 1-5. Jun 2009.
51. Knapp, E. D. and J. T. Langill. *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems, 2nd Ed.*. Elsevier, Inc. Waltham, MA, United States, 2015
52. Kocher, P., J. Jaffe, and B. Jun. "Differential Power Analysis In Advances in Cryptology" *CRYPTO 99, Vol. 1666* , 388-397, Dec 1999.
53. Koziel, E. *Effects Of Architecture On Information Leakage Of A Hardware Advanced Encryption Standard Implementation*. Thesis. Air Force Institute of Technology, 2012.
54. Lee, E. A., Seshia, S. A., *Introduction to Embedded Systems - A Cyber-Physical Systems Approach, Second Edition*, LeeSeshia.org, 2015.
55. Mangard, S., N. Pramstaller and E. Oswald. "Successfully Attacking Masked AES Hardware Implementations." *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings.*, 157-171, 2005.
56. Mathworks, "Discrete-time Analytic Signal Using Hilbert Transform. Website
57. Mathworks, "Norm of Matrix or Vector." Website
58. McAfee Labs. *2016 Threats Predictions*. Technical Report, McAfee, 2015.

- 59. Miron, R. “Security & Microcontroller Design”, <http://www.radio-electronics.com/articles/processing-embedded/security-amp-microcontroller-design-144>, Apr 2015. (accessed February 24 2016).
- 60. Montminy, David P. *Enhancing Electromagnetic Side-Channel Analysis in an Operational Environment*. Dissertation, Air Force Institute of Technology, 2013.
- 61. Msgna, M., K. Markantonakis, and K. Mayes. “Precise instruction-level side channel profiling of embedded processors. *Information Security Practice and Experience, ser. Lecture Notes in Computer Science, X. Huang and J. Zhou, Eds.* Springer International Publishing, vol. 8434, 129-143, 2014.
- 62. Msgna, M., K. Markantonakis, D. Naccache, and K. Mayes. “Verifying software integrity in embedded systems: A side channel approach. *Constructive Side-Channel Analysis and Secure Design, ser. Lecture Notes in Computer Science, E. Prouff, Ed.* Springer International Publishing, vol. 8622, 261-280, 2014.
- 63. NASA. *EEE Parts Bulletin*. Technical Report, NASA, May/Jun 2011.
- 64. National Institute of Standards and Technology. Federal Information Processing Standards Publication 197 (Advanced Encryption Standard), 2001.
- 65. Nohl, K., S. Krißler, J. Lell. “BadUSB - On accessories that turn evil”, <https://srlabs.de/blog/wp-content/uploads/2014/07/SRLabs-BadUSB-BlackHat-v1.pdf>, 2014
- 66. Obama, B. *Executive Order : Improving Critical Infrastructure Cybersecurity*. Technical Report, The White House, Washington DC: U.S. Government, Feb 2013.
- 67. Ott, H. W. *Electromagnetic Compatibility Engineering*. John Wiley & Sons, Inc., 2009.

68. Paganini, P. "SCADA Systems Available for Sale in the Underground", <http://securityaffairs.co/wordpress/38134/cyber-crime/scada-systems-underground.html>, 2015. (accessed February 24 2016).
69. Potamianos, A., R. Potamianos, P. Maragos, and P. Maragos. *A Comparison of the Energy Operator and the Hilbert Transform Approach to Signal and Speech Demodulation*. Technical Report, School of Electrical & Computer Engineering, Georgia Institute of Technology, 1994.
70. Power Fingerprinting. "Supply Chain Protection: A White Paper on Counterfeit Detection." Technical Report, 2015.
71. Reising, D. *Exploitation of RF-DNA for Device Classification and Verification Using GRLVQI Processing*. PhD dissertation, Air Force Institute of Technology, 2012.
72. Reising, D., and M. Temple. "WiMAX Mobile Subscriber Verification using Gabor-Based RF-DNA Fingerprints. *Communications (ICC), 2012 IEEE International Conference on*. 1005-1010. 2012.
73. Reising, D., M. Temple, and M. Mendenhall. "Improved Wireless Security for GMSK Based Devices Using RF Fingerprinting, *Int. J. Electron. Secur. Digit. Forensic*, 3 :41-59 (Mar 2010).
74. Reising, D., M. Temple, and M. Oxley. "Gabor-based RF-DNA Fingerprinting for Classifying 802.16e WiMAX Mobile Subscribers. *Computing, Networking and Communications (ICNC), 2012 International Conference on*. 7-13. Feb 2012.
75. Salem, M. B. "Security Challenges and Requirements For Industrial Control Systems in the Semiconductor Manufacturing Sector. *NIST Workshop on*

Cyber-Security for Cyber-Physical Devices, http://csrc.nist.gov/news_events/cps-workshop/slides/presentation-3_salem.pdf, 2012. (accessed February 24 2016).

76. Scarfone, K., and P. Mell, "Guide to Intrusion Detection and Prevention Systems, Feb 2007.
77. Schollmeyer, M. F. and W. H. Tranter, "Noise generators for the simulation of digital communication systems," *Simulation Symposium, 1991., Proceedings of the 24th Annual, New Orleans, LA*, 264-275, 1991.
78. Sklar, B. *Digital Communications: Fundamentals and Applications*. Prentice Hall, 2009.
79. Stone, B. and S. Stone. "Radio Frequency Based Reverse Engineering of Micro-controller Program Execution." *Proceedings of the 2015 IEEE National Aerospace and Electronics Conference (NAECON)*. 159-164. Sep 2015
80. Stone, B. and S. Stone. "Comparison of Radio Frequency Based Techniques for Device Discrimination and Operation Identification." *International Conference on Cyber Warfare and Security*. (2016, UNDER REVIEW)
81. Stone, S. J. *Radio Frequency Based Programmable Logic Controller Anomaly Detection*. Dissertation, Air Force Institute of Technology, 2013.
82. Stone, S., and M. Temple. "RF-Based Anomaly Detection for PLCs. *Sixth Annual IFIP Working Group 11.10 International Conference on Critical Infrastructure Protection*. 2012.
83. Stone, S., and M. Temple. "RF-Based Anomaly Detection For PLCs in Critical Infrastructure Applications, *International Journal of Critical Infrastructure Protection*, 5(2):11-33 (Jul 2012).

84. Stone, S., and M. Temple. Detecting Anomalous SCADA Operation Using RF-Based Hilbert Transforms, *International Journal of Critical Infrastructure Protection*, 5(2):11-33 (Jul 2013).
85. Stone, S., and M. Temple, and R. Baldwin. "RF-Based PLC IC Design Verification. *2012 DMSMS & Stand Conf. (DMSMS12)*. Nov 2012.
86. Strobel, D., F. Bache, D. Oswald, F. Schellenberg, and C. Paar. "SCANDALee: A Side-ChANnel-based DisAssembLer Using Local Electromagnetic Emanations," *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015, Grenoble*, 139-144, 2015.
87. Suski, W., M. Temple, M. Mendenhall, and R. Mills. "Radio Frequency Fingerprinting Commercial Communication Devices to Enhance Electronic Security, *Int. J. Electron. Secur. Digit. Forensic*, 1:301-322 (Oct 2008).
88. Suski, W., M. Temple, M. Mendenhall, and R. Mills. "Using Spectral Fingerprints to Improve Wireless Network Security. *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*. 1-5. Dec 2008.
89. Swierczynski, P., M. Fyrbiak, P. Koppe and C. Paar. "FPGA Trojans Through Detecting and Weakening of Cryptographic Primitives." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(8), pp.1236-1249. 2014.
90. Texas Instruments. MSP430x5xx and MSP430x6xx Family User's Guide. 2014
91. Texas Instruments, <http://www.ti.com/tool/AES-128>
92. Wawro, A. "A Hacker Speaks: How Malware Might Blow Up Your Laptop", http://www.pcworld.com/article/236875/batteries_go_boom.html, 2011.

93. Weiss, J. *Protecting Industrial Control Systems from Electronic Threats*. Momentum Press, LLC, 2010.
94. Williams, M., M. Temple, and D. Reising. "Augmenting Bit-Level Network Security Using Physical Layer RF-DNA Fingerprinting." *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*. 1-6. Dec 2010.
95. Williams, M., S. Munns, M. Temple, and M. Mendenhall. "RF-DNA Fingerprinting for Airport WiMax Communications Security." *Network and System Security (NSS), 2010 4th International Conference on*. 32-39. Sep 2010.
96. Wright, B. *PLC Hardware Discrimination Using RF-DNA Fingerprinting*. Thesis, Air Force Institute of Technology, 2014.
97. Wylie, J., Stone, S., and Mullins, B. "Detecting a Weakened Encryption Algorithm in Microcontrollers Using CorrelationBased Anomaly Detection." *Proceedings of the 2016 International Conference on Cyber Warfare and Security (ICCWS) Conference*, (2016, UNDER REVIEW)
98. Wylie, J. and Stone, S. "Detecting Anomalous Behavior in Microcontrollers Using Unintentional Radio Frequency Emissions." *Proceedings of the 2015 IEEE National Aerospace and Electronics Conference (NAECON)*, 151-154, Sep 2015.
99. Yudken, J. *Manufacturing Insecurity: Americas Manufacturing Crisis and the Erosion of the U.S. Defense Industrial Base*. High Road Strategies, <http://www.nist.gov/mep/data/upload/1-Manufacturing-Insecurity-ES-V2-1.pdf>, 2010. (accessed February 24 2016).
100. Zetter, K. *How Digital Detectives Deciphered Stuxnet, the Most Menacing Malware in History*. Technical Report, Jul 2011.

REPORT DOCUMENTATION PAGE					<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.						
1. REPORT DATE <i>(DD-MM-YYYY)</i>		2. REPORT TYPE		3. DATES COVERED <i>(From — To)</i>		
24-03-2016		Master's Thesis		Sept 2014 — Mar 2016		
4. TITLE AND SUBTITLE				5a. CONTRACT NUMBER		
Radio Frequency-Based Microcontroller Anomaly Detection				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)				5d. PROJECT NUMBER		
Wylie, Justin, Captain, USAF				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)				8. PERFORMING ORGANIZATION REPORT NUMBER		
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765				AFIT-ENG-MS-16-M-053		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)		
Air Force Research Laboratory 2241 Avionics Circle WPAFB OH 45433-7322 DSN 798-8062, COMM 937-938-8062 Email: yong.kim@us.af.mil				AFRL/Rywa		
11. SPONSOR/MONITOR'S REPORT NUMBER(S)						
12. DISTRIBUTION / AVAILABILITY STATEMENT						
DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.						
13. SUPPLEMENTARY NOTES						
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.						
14. ABSTRACT						
The research presented here focuses on applying the Correlation-Based Anomaly Detection (CBAD) process to a Microcontroller Unit (MCU) as a means of detecting an anomalous behavior by verifying that the device is behaving normally. This research utilizes a previously developed method to collect Unintentional Radio Frequency (RF) Emissions (UREs) Texas Instruments MSP430F5529 microcontrollers. Once captured, the time domain signals are processed and used as one of two inputs into the CBAD algorithm, which generates a statistic to be compared to a threshold. Like-device performance met the arbitrary benchmark of Equal Error Rate (EER) $\leq 10\%$. Cross-device performance met the benchmark when the reference signal originated from the same manufacturing lot, but did not meet the benchmark when the reference signal originated from a different manufacturing lot.						
15. SUBJECT TERMS						
correlation, unintentional emissions, anomaly detection, microcontroller, Advanced Encryption Standard (AES)						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT		18. NUMBER OF PAGES	
a. REPORT	b. ABSTRACT	c. THIS PAGE			19a. NAME OF RESPONSIBLE PERSON	
U	U	U	U		Major Samuel J. Stone, AFIT/ENG	
					19b. TELEPHONE NUMBER <i>(include area code)</i>	
					(937) 255-3636, x4555; samuel.stone@afit.edu	