

Novelty Detection for **SilGeo** Hardware Assurance

by

Lukas Schmidt

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2023

© Lukas Schmidt 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In today’s world, electronic hardware-level threats have become increasingly common. These threats can range from the infiltration of counterfeit or malicious hardware in the supply chain to the use of electronic attack tools. Due to the ever-evolving nature and customizability of inauthentic electronics, it is difficult to validate the authenticity of hardware. While measurement devices have been created to detect these threats, their successful deployment requires a high level of expertise.

This thesis addresses these challenges by proposing a novelty detection method for the [SILGEO](#) hardware validation platform with low deployment barriers. It has been shown that this method can be trained on as few as three valid devices, and the entire training and application process is fully automated, requiring no expertise. The method incorporates Bayesian statistical models that rely on carefully selected assumptions and domain knowledge. Furthermore, maximum false positive rates can be estimated and adjusted without additional data.

The presented method is tested in several case studies on devices ranging from surface-mount integrated circuits to Wi-Fi-enabled disguised attack tools. In each case, the estimated maximum false positive rate exceeded the observed false positive rate, and most counterfeit and malicious devices were identified. This thesis presents a practical solution to detecting and validating hardware in a rapidly changing threat landscape.

Acknowledgements

I would like to thank my supervisor Professor Sebastian Fischmeister for his guidance and support throughout my time at the University of Waterloo. In particular, thank you for your encouragement and the opportunities you have presented me throughout my Undergraduate and Graduate journey. Thank you, Dr. Carlos Moreno, for our weekly discussions and for providing deep and detailed feedback. This work would not exist as it is without the feedback of Dr. Oleg Iegorov, whose rigorous methodology has inspired much of my work. Special thanks to Phillipe Vibien, Goksen Umut Guler, and Srijan Pabbi, whose work and help made this thesis possible.

Thank you to Arthur Grisel-Davy, Clemence Blandnet, and Ridham Dave for making the Lab a joy to work in. To Dr. Apurva Narayan and Dr. Jean-Christophe Petkovich, thank you for introducing me to academia and for your generous guidance.

To Julia, Kai, Derek, Brittany, Dillon, Johnson, Inah, and Ben, thank you for your help and encouragement with this thesis.

Dedication

I dedicate this thesis to Susi, Joachim, Tim, and Benno Schmidt.

Table of Contents

List of Figures	viii
List of Tables	x
List of Abbreviations	xi
1 Introduction	1
2 Background	4
2.1 Hardware Level Threats And Risks	4
2.1.1 The Risks Induced By Modern Electronic Supply Chains	4
2.1.2 Hardware Level Attack Devices	5
2.2 SILGEO As A Platform For Hardware Validation	6
2.3 SILGEO Fingerprint Analysis	8
2.4 Novelty detection	8
2.4.1 Data Modeling vs Algorithmic Modeling	10
2.4.2 Parametric vs Non-Parametric	12
2.4.3 Estimating Parameters: Frequentist Point Estimates versus Bayesian Distribution Estimates	13
2.4.4 Probability Distribution Estimating vs Conjugate Priors	15

3	Proposed Method	16
3.1	Feature Engineering	17
3.1.1	Feature Extraction	19
3.1.2	Dimensionality Reduction	21
3.2	Statistical Model	23
3.2.1	Generating Distribution	24
3.2.2	Prior Covariance Matrix Knowledge	26
3.2.3	Bayesian Inference	26
3.2.4	Test Statistic	29
3.2.5	Determining Hyper Parameters	30
3.2.6	Model Aggregation	32
3.3	Summary	34
4	Experimental Evaluation	35
4.1	FTDI FT232RL Case Study	35
4.2	Zilog Z80 Case Study	41
4.3	Hardware Attack Tools Case Study	42
5	Future Work	47
6	Conclusions	48
	References	49
	APPENDICES	54
A	Hotelling’s T Test Derivation	55
	Glossary	58

List of Figures

2.1	A photo of the SILGEO measurement device with a USB adapter board. Note that SILGEO was rebranded to Anvil, hence the label seen in the photo. (<i>This photo was taken by Srijan Pabbi</i>)	7
2.2	An overview of how we categorize novelty detection methods. The Section references to the right correspond to sections in which each division is discussed.	9
3.1	Visualization of a dataset containing measurments of 200 z80 devices. . . .	18
3.2	Visualization of the features in a single measurement of a pin pair by SILGEO. . . .	18
3.3	An example of a single Scattering parameter (S-parameter) measurement of a single port combination.	20
3.4	An example of a transmission S-parameter plotted in red against the mean of nominal measurements plotted in black. A vertical grey line denotes the difference between the two at each feature.	21
3.5	This figure contains the difference between the sample and mean depicted in Figure 3.4. The vertical black lines indicate how the features are partitioned. . . .	22
3.6	This figure contains the averages of the differences shown in Figure 3.5. . . .	23
3.7	A two dimensional multivariate normal distribution with a mean at $[0, 0]$ and and the identity matrix as the covariance matrix. The x and y axes denote the space on which the distribution is defined, and the z axis shows the probability density.	25
4.1	A visualization of the distribution of FT232RL experiment test set with respect to the test statistic.	40

4.2	A visualization of the distribution of Z80 experiment test set with respect to the test statistic.	43
4.3	A visualization of the distribution of the Rubber Ducky experiment test set with respect to the test statistic.	45

List of Tables

3.1	Example phase delay and group delay at various frequencies in response to a frequency-invariant delay of 0.1 seconds.	19
4.1	Confusion matrix showing the classification of the novelty detection method applied to the FT232RL dataset when trained on four devices. A maximum expected false positive rate of 5% was selected for this classification.	36
4.2	Confusion matrix showing the classification of the novelty detection method applied to the FT232RL dataset when trained on five devices. A maximum expected false positive rate of 5% was selected for this classification.	37
4.3	Confusion matrix showing the classification of the novelty detection method applied to the FT232RL dataset when trained on ten devices. A maximum expected false positive rate of 5% was selected for this classification.	38
4.4	Sum of Anomalous Labels by the Novelty Detector on the FT232RL dataset for all electrical ports.	38
4.5	Number of anomalous electrical port combinations found by the Novelty Detector on the FT232RL dataset for all electrical port combinations. Note that the first row and column indicate the electrical port combination.	39
4.6	Confusion matrix showing the classification of the novelty detection method applied to the Z80 dataset. For the results in this table, a maximum expected false positive rate was set to 5% and 40 devices were used for training.	42
4.7	Confusion matrix showing the classification of the novelty detection method applied to the Z80 dataset. For the results in this table, a maximum expected false positive rate was set to 5% and 40 devices were used for training.	44

List of Abbreviations

DUT Device Under Test [1](#), [2](#), [6](#), [8](#), [16](#), [20](#), [25](#), [32](#), [47](#), [48](#)

IC Integrated circuit [1](#), [2](#), [5](#), [6](#)

S-parameter Scattering parameter [viii](#), [8](#), [17](#), [19–22](#), [26](#), [32](#), [33](#), [36](#), [39](#), [41](#)

Chapter 1

Introduction

Electronic components play a critical role in many aspects of modern life. They are used in defense systems, secure communication networks, public utilities and food supply chains. However, modern supply chains’ global and complex nature has led to the widespread issue of counterfeit [Integrated circuits \(ICs\)](#) [23]. Fake [ICs](#) can be found in a range of industries, including defense [43], aerospace, automotive, and essential utilities [45], and they pose a significant threat to safety-critical sectors.

The increasing reliance on electronics in critical infrastructure has also increased the threat of disguised electronic attack tools. These tools, such as keyloggers disguised as USB keys, radio-active USB cable sniffers, and “USB kill sticks,” are widely available at low costs and do not require advanced expertise to use effectively [24]. This ease of access and concealment makes them a significant concern for organizations and individuals.

Various hardware validation techniques have been developed to address hardware threats and risks, which can be broadly classified as physical or electrical inspections [23]. Physical inspections include techniques such as x-ray material analysis and internal inspection via optical analysis after decapsulating or delidding an [IC](#). Electrical inspections involve testing the functionality of the [IC](#) and their durability via burn-in tests. However, these methods are often costly, time-consuming manual procedures that require trained specialists and may destroy the [Device Under Tests \(DUTs\)](#).

[SILGEO](#), proposed by Monero et al. [34] and presented Vibien’s thesis [47], is a promising platform for hardware validation. [SILGEO](#) enables the collection of sensitive fingerprints from electronic components without powering the [DUT](#). Passive measurement allows for the testing of devices without specific power requirements and prevents any activity

or changes on the DUT. The collected fingerprints are created by measuring the propagation of electromagnetic waves within the DUT. This non-invasive approach could be a cost-effective and efficient method for hardware validation.

For the measurements collected by SILGEO to be useful for electronic manufacturers or security teams, a robust model is needed to differentiate between valid and invalid devices. In an electronics production environment, various types of ICs are often used in the same product. Each IC is given a part number and goes through a part approval process; for example, consider the DO-160 standard for environmental testing in the aerospace industry [1]. Note that when changes are made to electronic components, they are typically assigned a new part number and require a new round of testing. The supply chain for these ICs is often untrusted and may not remain consistent throughout the production run of a product. This loss of suppliers for an existing item is called the diminishing manufacturing sources and material shortages problem [2]. Obtaining and testing counterfeit ICs ahead of time can be challenging, time-consuming, and expensive. A testing model used in this setting must be able to verify that new ICs are the same type as the approved devices, using a small sample of verified devices for training. In a production line setting, false positives can be costly, requiring manual intervention, and high false positive rates can lead to alarm fatigue. The model must consider false positive considerations despite the lack of invalid ICs for training or testing.

In previous work, simple distance-based metrics have been applied to SILGEO measurements to successfully detect counterfeit FT232RL microchips [47]. This thesis aims to address the problem of detecting counterfeit ICs using the SILGEO platform by developing a novelty detection model. Given a small sample batch of verified ICs, we develop a fast, automatic method for assuring that unseen ICs belong to the same class as the verified ICs. We expand on the work by Monero et al. [34] to include the following:

1. fully automated model training
2. maximum false positive rate estimates
3. granular information on anomalies

One constraint of our model is that it is only to be trained on a limited number of nominal devices. The introduced model uses Bayesian inference to incorporate domain knowledge to overcome the small sample size and account for epistemic uncertainty.

This thesis is organized as follows: In Chapter 2, we give examples of hardware-level threats and place SILGEO in the context of other hardware validation techniques. We then supply some background in electronic network analysis and novelty detection. In Chapter 3,

we introduce our novelty detection method, followed by applications in Chapter 4. Finally, we concluded with our final thoughts and recommended the next steps in Chapter 6.

Chapter 2

Background

This chapter covers the following background information. Section [2.1](#) and Section [2.2](#) give examples of impactful hardware-level threats and current hardware validation techniques, respectively. We cover some relevant systems analysis topics required to understand our data processing choices in Section [2.3](#). Finally, we provide a breakdown of the field of novelty detection to provide context for the decisions we make in Chapter [3](#).

2.1 Hardware Level Threats And Risks

Two main threats create a need for fast, scalable hardware validation. The first threat comes from the nature of modern supply chains; we discuss this first. The second threat comes from the increased availability of disguised hardware attack tools.

2.1.1 The Risks Induced By Modern Electronic Supply Chains

The complex, opaque, global, and vulnerable nature of modern supply chains for electronics has become a matter of significant concern for many organizations, including defense departments, electronics manufacturers, and healthcare systems [\[23\]](#). The widespread dependence on these supply chains highlights the critical importance of addressing their vulnerabilities, which can have far-reaching and significant impacts on society. To fully comprehend the extent of the risks and consequences associated with modern electronics supply chains, comprehensive analyses are given by Vibien [\[47\]](#) and Halak [\[25\]](#). For this

thesis, we provide the following examples to illustrate the impactful nature of supply chain attacks.

Counterfeit ICs and malicious hardware modifications have been a persistent concern for the defense sector of the United States of America. With the increasing reliance on electronic components in military equipment, counterfeit ICs in defense supply chains have become a significant issue [28]. The pervasiveness of counterfeit electronics prompted an inquiry by the Committee on Armed Services to assess the threat posed by counterfeit parts and the current state of counterfeits in the defense industry [43]. The inquiry uncovered approximately 1 800 suspect counterfeit parts cases, including cases in various military aircraft and an anti-ballistic missile defense system. One of these cases involved 84 000 parts suspected to be counterfeit.

The proliferation of counterfeit devices in modern electronic supply chains is having a significant impact on electronic manufacturers. Research has shown that the economic impact of these counterfeit devices amounts to approximately \$100 billion globally for legitimate electronics companies [39]. The detrimental effects of these counterfeit chips involve safety concerns, lost sales of legitimate chips, and product failure caused by subpar counterfeit parts [23].

2.1.2 Hardware Level Attack Devices

Cybersecurity efforts are facing a growing threat from hardware attack tools, which can compromise the security and integrity of electronic devices. These tools, often disguised as commonplace electronics such as USB keys, can destroy, spy on, or gain control of their targets. Attack tools have become cheap, easily accessible, and simple to operate, making them a significant concern for organizations and individuals alike [24].

One such type of attack tool is the rubber ducky. The rubber ducky is an implementation of a BadUSB attack [38]. Disguised as a USB key, these devices pretend to be human interface devices such as a keyboard or mouse. They can be used to install backdoors, capture credentials, and exfiltrate sensitive documents. At the time of writing, disguised rubber duckies can be purchased for \$79.99 [24].

Another type of attack tool is the O.MG cable, which is disguised as a USB cable. In addition to the capabilities of rubber duckies, O.MG cables can be remotely controlled through their built-in antennas. These cables can perform a wide range of attacks, making them a significant threat to the security of electronic devices. At the time of writing, O.MG cables can be purchased for \$119.99 [24].

2.2 SilGeo As A Platform For Hardware Validation

Guin et al. [23] organizes hardware validation techniques into physical and electrical inspections. Physical inspections involve methods that measure the physical characteristics of a device, such as material makeup and form. Examples of physical inspections include x-ray imaging, interior visual analysis after chip disassembly, and external optical inspection. On the other hand, electrical inspections are concerned with the functional aspects of the electronic device. Examples of electrical inspections include burn-in tests, power consumption tests, and functionality tests. The main advantage of electrical inspections includes reduced test noise as only properties pertaining directly to a DUT's functionality are measured. The main advantage of physical inspections is that they do not necessarily need to be custom-created for each type of device.

A measurement captured with the SILGEO platform would be considered a physical inspection in the classification scheme discussed above. SILGEO creates fingerprints of electronic devices based on measurements of electromagnetic wave propagation and impedance discontinuities within the DUT. These measurements are taken by connecting exposed pins of a device through an adapter board to the SILGEO measurement system. These exposed pins may be in electrical connectors such as pin headers, cable connectors, and IC sockets. For example, SILGEO might gather a fingerprint on any device with a USB port via the USB adapter board. The set of adapter boards, as of writing, can measure surface mount FT232RL chips, Multilin 850 feeder protection relays, USB devices, HDMI devices, and through-hole Z80 microprocessors, but new adapter boards can be created to interface with virtually any electronic device.

The fingerprints collected by SILGEO should be sensitive to any functional differences between different DUTs. In Section 2.3, we discuss the properties of the SILGEO fingerprints in detail. The sensitivity of SILGEO fingerprints to functional changes in the DUT makes SILGEO a prime candidate for hardware validation. Unlike interior visual analysis, SILGEO is not a destructive test, and unlike most electrical inspections, SILGEO does not power on the DUT. Leaving the DUT unpowered prevents rubber duckies and OM.G cables from operating as they require an external power source. Additionally, some explosive USB attack devices also require external power to activate [26], power which the SILGEO platform does not supply.



Figure 2.1: A photo of the [SILGEO](#) measurement device with a USB adapter board. Note that [SILGEO](#) was rebranded to Anvil, hence the label seen in the photo. *(This photo was taken by Srijan Pabbi)*

2.3 SilGeo Fingerprint Analysis

SILGEO measures certain electrical behaviours of electronic devices. This section gives some background on what electrical properties **SILGEO** measures. A **SILGEO** fingerprint consists of a set of measurements taken while connected to two electrical ports of the **DUT** at a time. In total, **SILGEO** can connect to 16 different electrical ports of a **DUT** and take measurements between up to 64 electrical port pairs.

When measuring an electrical port pair, **SILGEO** uses a Vector Network Analyzer to capture the **S-parameters** of the electrical port pair. An ordering is assigned to the electrical ports in the port pairs, with one assigned to 0 and the other to 1. **SILGEO** characterizes the **DUT** through the given electrical port pair with four **S-parameters**: S_{00} , S_{10} , S_{01} , and S_{11} . A single **S-parameter** is created by first injecting an electromagnetic signal at a chosen frequency into a chosen electrical port and reading the response at a chosen electrical port. The first subscript of an **S-parameter** indicates the electrical port into which the signal was injected, and the second subscript indicates the port from which the response was read. So S_{11} is the response read from the electrical port assigned to one from the electrical signal injected into that same port.

A single **S-parameter** at a given frequency is a complex number with a meaningful interpretation when viewed in polar form. The absolute value $|S_{ij}|$ of an **S-parameter** at a given frequency gives the linear gain magnitude, which is the ratio of the output signal amplitude to the input signal amplitude. The angle of the value $\tan^{-1}(\frac{\text{Im}(S_{ij})}{\text{Re}(S_{ij})})$ gives the phase shift, which is the delay of the response in comparison to the input wave with respect to the wavelength of the input wave. S_{00} and S_{11} represent reflection coefficients, which indicate impedance discontinuities in the **DUT**. S_{10} and S_{01} represent transmission coefficients, which characterize how the electromagnetic wave travels through the **DUT**. For each of the 64 electrical port pairs that **SILGEO** has access to, **SILGEO** measures all four S-parameters at a series of different frequencies, typically between 300kHz to 4.5 GHz.

2.4 Novelty detection

Novelty detection is the task of identifying if some previously unseen sample $\mathbf{x}$ “differs” from a set of previously seen samples $X_{\text{train}} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$. We call samples that do not “differ” nominal and the samples that “differ” anomalous. For each sample $\mathbf{x}$, we define a label $l \in \{0, 1\}$ that is 0 for nominal and 1 for anomalous. Note that we follow the convention and consider correct anomalous labelling as a true positive and correct nominal

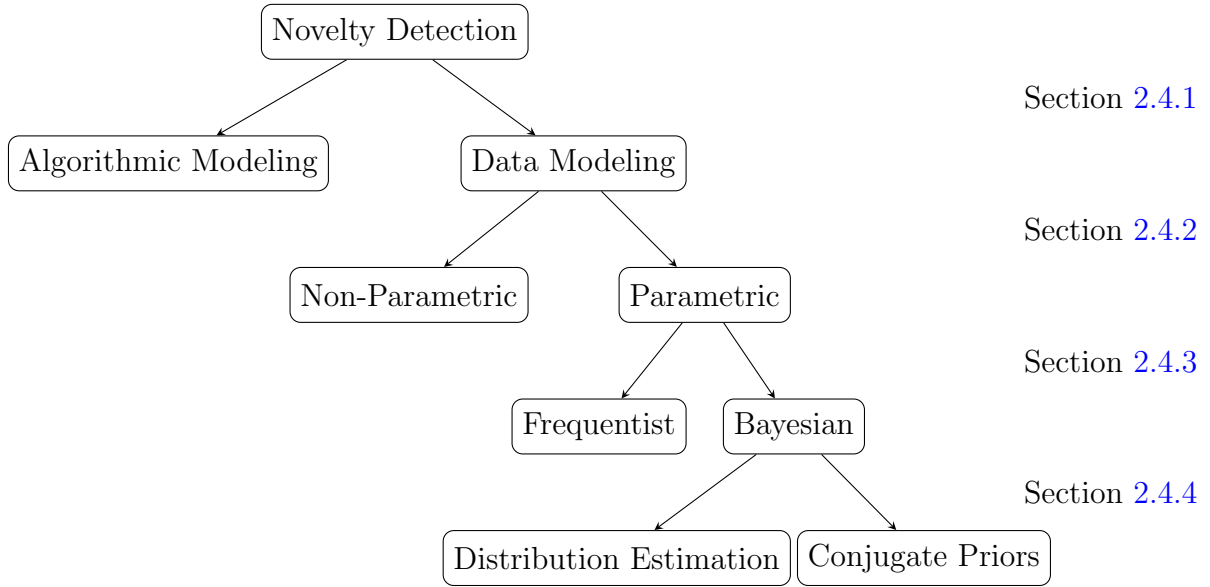


Figure 2.2: An overview of how we categorize novelty detection methods. The Section references to the right correspond to sections in which each division is discussed.

labelling as a true negative. So in the novelty detection problem statement, we are first given a dataset of only nominal samples X_{train} . We then must determine if the label l for any new sample $\mathbf{x}$ is nominal (0) or anomalous (1). Unfortunately, there is no single definition for “differ” in novelty detection; it changes depending on how the problem is approached. Note that novelty detection is often referred to as discordancy testing in the statistical literature [40].

The challenge of anomaly detection is to capture the patterns exclusive to nominal samples without examples of abnormal samples. Most novelty detection algorithms have the same basic structure. Some kind of model is fit to the training dataset of nominal samples, and then the fit model is used to give a novelty score to new samples. These novelty scores are then used to make a decision, typically through comparison to some threshold. In this section, we hierarchically categorize various novelty detection methods. Figure 2.2 provides an overview of how we break down novelty detection methods in this section.

2.4.1 Data Modeling vs Algorithmic Modeling

Data analysis (and novelty detection) methods can be categorized into methods that assume samples come from a stochastic generating model and those that assume the type of pattern which can distinguish between nominal and abnormal samples [14]. We follow Breiman’s terminology [14] by calling the approach that assumes a stochastic generating model the Data Modeling approach and the assumption-free approach the Algorithmic Modeling approach.

In the Data Modeling approach to novelty detection, a data science practitioner assumes that samples are generated by a set of possible stochastic generating models. The practitioner then uses the training dataset to select the most appropriate stochastic generating model according to some statistically meaningful metric. This approach can validate the model based on how well it matches the training data using goodness-of-fit tests. Goodness-of-fit tests allow for some model validation despite the lack of abnormal testing samples. A test statistic derived from the stochastic generating model is then used to determine a novelty score for novel samples. Since the novelty score is derived from the stochastic generating model, a threshold with statistical meaning can be set by practitioners. For example, a practitioner might set a threshold that results in a target expected false positive rate.

For example, a practitioner may assume a set of one-dimensional samples come from a normal distribution with known variance and unknown mean. In this case, the set of possible stochastic generating models is the set of all normal distributions with the known variance and any possible mean. A practitioner may select the normal distribution with the mean found using a Maximum likelihood estimate. The test statistic for this distribution may be the standard score (the number of standard deviations a new sample is from the mean) of the generating distribution. The threshold could be set based on the maximal acceptable false positive rate for the percentage of nominal observations expected to have a lower standard score to give a desired false positive rate.

In the Algorithmic Modeling approach, no assumptions are directly made on the distribution from which the data is drawn. Instead, various models are fit to the training data, and a test dataset is used to evaluate the model’s performance. This approach relies entirely on test and validation datasets to determine model performance. The advantage of this approach is the relative ease with which a model can learn complex patterns. Instead of learning the entire generative distribution, the model has to estimate a decision boundary [42]. Most examples of models successfully applied to highly complex problems follow the Algorithmic Modeling approach, such as deep neural networks [3] and gradient boosting methods [20].

For example, a practitioner may use a One-Class SVM [42] for a novelty detection problem. During training, the One-Class SVM attempts to learn a function that is positive for anomalous samples and negative for nominal samples from a training dataset containing only nominal samples. The One-Class SVM does not attempt to learn the probability density of novel nominal samples over the feature space. To determine the model’s efficacy, it is then tested on a test dataset containing both nominal and abnormal samples.

When applied to novelty detection, there are two practical differences between Data Modeling and Algorithmic Modelling approaches. The first difference is in how models are validated, and the second is in how well the approaches scale to more complex models.

Algorithmic Modeling approaches validate their models through predictive accuracy. Given a dataset, the first step for an Algorithmic Modeling approach is to consider how to split the dataset into training and test sets. The size and diversity of the test sets determine how representative the predictive accuracy is of model performance. This dependency on the size and quality of the test set can make it very challenging or impossible to create novelty detection models when working with a restricted or non-existent test set. On the other hand, data modelling approaches validate their models through statistics on how well the model “fits” the dataset. There are a variety of interpretations of what it means for a model to “fit” a dataset, leading to diverse selection model validation methods, including goodness-of-fit tests, residual examination, and posterior predictive checks [9]. Generally, Data Modeling approaches for novelty detection do not require a test set or even an example of an abnormal observation for model validation.

Creating complex models using the Data Modeling approach is typically a difficult process. Recent advances in computation methods and ability have made it possible to estimate more complex probability distributions via sampling methods such as the Hamiltonian Markov Chain Monte Carlo class of algorithms [11]. However, it is computationally expensive and difficult to estimate complex probability distributions numerically, and they must be hand-crafted for every application. The computational and labour requirements of complex models using the Data Modeling approach make scaling the Data Modeling approach to complex domains challenging. Algorithmic Modeling approaches, however, are much better at scaling in complexity. The main requirement for Algorithmic Modeling approaches to increase in complexity is more high-quality data. Recent Neural Networks’ performance exemplifies Algorithmic Modelling’s ability to scale to incredibly complex models [41].

The division between Data Modeling and Algorithmic Modeling can be blurred by applying Data Modeling approaches to the results of Algorithmic Modeling methods. For example, one might map the dataset into a latent space via autoencoders and then as-

sume the distribution of the dataset in the latent space to continue with a Data Modeling approach.

For our application, we do not have anomalous samples to use in our test set; we also have a limited number of samples to learn from. Despite this, we wish to estimate metrics of our model performance, such as expected false positive rates. These requirements restrict us to Data Modeling approaches.

2.4.2 Parametric vs Non-Parametric

In this section, we partition Data Modeling methods into parametric and non-parametric approaches. Parametric methods make assumptions about the shape of the distributions of the population that samples are drawn from and use the dataset to estimate the parameters of the assumed population distribution. Non-parametric methods are based on robust statistical tests that maintain their statistical properties across broad classes of stochastic processes.

In general, non-parametric methods have lower statistical power in exchange for requiring virtually no assumptions on the underlying distribution of the data [40]. Note that statistical power refers to the probability that a hypothesis test will correctly reject the null hypothesis; in other words, it is the sensitivity of the statistical method. Non-parametric methods are useful when little is known about the data being modelled and there are enough samples to inform the non-parametric model. On the other hand, parametric methods typically require some domain knowledge to make the necessary assumptions on the underlying distributions of the data. The advantage of parametric models is that they can incorporate prior knowledge; however, their performance depends on the prior knowledge’s quality and validity. The tradeoff between parametric and non-parametric models is generally an example of the bias-variance tradeoff [49]. Non-parametric models generally have low bias but high variance, while parametric models tend to have a high bias with low variance. A model with high bias tends to be unable to learn all relevant properties of the distribution. In contrast, a model with high variance tends to learn more properties than are relevant from the distribution. The bias-variance tradeoff is a commonly faced dilemma in statistics. In well-designed parametric models, the tendency towards a high bias is reduced by making well-informed assumptions on the distribution of the dataset. The degree of various types of bias can be checked using goodness-of-fit tests [16]. An example of a non-parametric method for novelty detection is kernel density estimation [18] [48] [8] and the Kolmogorov-Smirnov test [17]. The Kolmogorov-Smirnov test exemplifies the low statistical power (high variance) of non-parametric methods [13]. The test loses

power by not making any assumptions about the population distribution as the test needs to consider more sources of uncertainty. The increased uncertainty makes it harder for the test to reject the null hypothesis. In this case, the null hypothesis states that a sample is nominal. The normal distribution process outlined in the previous section gives an example of a parametric method.

As in the previous paragraph, the division between parametric and non-parametric approaches is sometimes unclear. In the literature, there is often disagreement over which group models belong to [40].

For this thesis, we proceed with a parametric approach as we have domain knowledge of the distribution of measurements. The parametric approach should allow us to extract more signal from our data and improve our true positive rate.

2.4.3 Estimating Parameters: Frequentist Point Estimates versus Bayesian Distribution Estimates

The Data Modeling approach assumes a stochastic model generates the data. Furthermore, the parametric approach assumes the stochastic model generates the data while only allowing for parameters of the assumed model to be learned. There are two common approaches to estimating the parameters of the stochastic model. Parameters can be estimated with either point estimates, interval estimates, or described by estimate distributions. Typically, statisticians using a frequentist approach use point or interval estimates and statisticians using a Bayesian approach use distribution estimates. While the distinction between the types of estimates is relatively simple, the difference between Bayesian and Frequentist approaches occurs due to significantly different philosophies and methodologies [19]. This section contrasts the Frequentist point estimates against Bayesian distribution estimates. As has been the case for our previous distinctions, there are interpretations in which the distinction between Bayesian and Frequentist methodologies loses some meaning [6].

The Frequentist approach concerns the probability of the data $P(D)$ occurring given a stochastic model θ : $P(D|\theta)$. The estimates for parameters in the Frequentist approach are typically point estimates that maximize the likelihood of the data $P(D|\theta)$ (these are found using the Maximum Likelihood Estimate.) There are distribution estimators in the frequentist approach [51] however, they are not often used. Note that this approach does not require any assumptions on the parameters of a model. The initial assumptions on what type of distribution the nominal distribution is can be validated through standard goodness of fit tests [16].

Conversely, the Bayesian approach finds the probability of hypotheses $P(\theta)$ given the data D and priors $\theta_0, \dots, \theta_n$: $P(\theta|D, \theta_0, \dots, \theta_n)$. While both Frequentist and Bayesian methodologies require prior information to determine the generating distribution, the Bayesian approach allows for additional prior information to be incorporated into model parameters. The priors $\theta_0, \dots, \theta_n$ for model parameters are encoded in probability density functions assigned to each parameter and are referred to as simply “priors.” This allows inference to consider uncertainty in the parameter specification and forces all prior information to be explicitly encoded. An advantage of using probability distributions to describe our knowledge of our parameters is that we can update these distributions in light of new data iteratively. For novelty detection, an interesting probability distribution is the posterior predictive distribution $p(\tilde{\mathbf{x}}|D, \theta_0, \dots, \theta_n)$. The posterior predictive distribution is the probability distribution of nominal samples given our priors and training data. Bayesian model validation is done in various ways, including posterior predictive checks [10].

To contrast the Frequentist and Bayesian approaches, we give an example for each on a simple novelty detection problem. We are given a set of one-dimensional samples that follow a Normal distribution. The Frequentist approach may use a slightly altered Grubbs test [22]. The Grubbs test is meant for outlier detection but can easily be altered to consider a single novel sample as a potential outlier as opposed to all available samples. In the altered Grubbs test, the working hypothesis is that a novel sample belongs to the same distribution as the training dataset. The alternative hypothesis is that the novel sample is anomalous. The probability that the data occurs given each hypothesis is compared to determine if the novel sample is anomalous. If the data is “significantly more likely” given the alternative hypothesis than the working hypothesis, the novel sample is considered anomalous.

We follow with a Bayesian approach to the same novelty detection problem. Prior distributions are set for the parameters of the Normal generating distribution. The posterior generating distribution is computed by applying Bayes’ rule to the prior distribution and the likelihood of the data given our prior. A new sample is then labelled as anomalous if it is “significantly unlikely” that it was generated by the posterior predictive distribution [7].

In our case, we have prior information, which we discuss in Section 3.2.5. This prior information is best encoded using prior distributions on model parameters, leading us to use a Bayesian approach to novelty detection.

2.4.4 Probability Distribution Estimating vs Conjugate Priors

This section discusses the types of prior distributions we may assign to our nominal generating distribution parameters. The prior distributions we assign to the parameters inform our posterior distribution’s shape. In general, this posterior distribution cannot be found analytically. There are two ways around this problem. The first is to use Conjugate Priors; Conjugate priors of a given stochastic model are probability distributions that do not change when updated. That is to say, when a Conjugate Prior under the appropriate stochastic model is updated in light of new data, the posterior is the same distribution as the Conjugate Prior with different parameters. Calculating the new parameters for the posterior is typically fast. The second approach allows a more extensive set of prior distributions; however, there are limitations on the complexity of posterior distributions that can be accurately estimated. The posterior distributions are estimated through various sampling techniques such as Hamiltonian Monte Carlo [37]. This approach has increased in popularity recently as it has been enabled by the processing speed of modern computers, statistical programming libraries [15] [50], and recent algorithm advancements [27].

The advantage of using Conjugate Priors is updating priors is fast and accurate. The downside is that we are restricted to the conjugate priors forms for our given stochastic model. These priors may be able to represent our actual priors well despite their adjustable parameters. For example, encoding value ranges into conjugate priors is often impossible. The sampling approach allows setting custom priors to a certain extent. Current sampling techniques do not work on all distributions and often require extensive and deep knowledge of how the sampling algorithms work to apply them correctly. For example, current sampling techniques have yet to apply to high-dimensional distributions successfully. Sampling techniques also tend to be computationally intensive, with each iteration often requiring significant computing time.

Chapter 3

Proposed Method

This chapter presents a method for detecting anomalous hardware through [SILGEO](#) measurements. Given a training set of measurements and a new unseen measurement, our method aims to determine if the new measurement belongs to the distribution that generated the training set. In practice, there are two intended modes of operation for this method. The first mode is to verify a class of devices. In this case, the training set measurements are measurements of different Electronic Components that belong to a “nominal class,” and the goal of the detector is to determine if a novel device belongs to the “nominal class” based on its measurement. In the second mode of operation, the training set contains measurements of a single device; the method must then determine whether a new measurement comes from the same device.

The intended use cases of our method impose the following requirements on our model:

1. The training dataset contains only examples of nominal measurements.
2. The method should be able to estimate false positive rates without consulting a test dataset, as we may not have access to anomalous examples.
3. The method should be explainable; specifically, anomalies should be detected with respect to pins on the [DUT](#).
4. The method should work with training datasets containing few samples, where few means less than 50.

These requirements introduce the following challenges that inform the design of our method:

1. The first challenge is the high number of features that a single **SILGEO** measurement creates in contrast to the number of training samples. Our datasets often have two orders of magnitude more dimensions than samples, even after dimensionality reduction. Novelty detection in a high dimensional setting is challenging [53], and the “curse of dimensionality” takes many forms that are discussed in this section.
2. The second challenge is the lack of anomalous examples for testing. Without a test set including anomalous samples, we have to rely on our assumptions on the generating distribution of our measurements to derive expected false positive rates for our method.

To allow for granular novelty detection, we partition **SILGEO** features into feature subsets and apply a separate novelty detector on each feature subset. After some feature engineering, our novelty detectors model each feature subset as a multivariate normal distribution and derive a test statistic for a new measurement using Mahalanobis distance. To address the challenge presented by the curse of dimensionality, we apply some dimensionality reduction and use Bayesian updating to allow our model to incorporate prior domain knowledge.

3.1 Feature Engineering

This section describes the process that starts with a **SILGEO** measurement and ends with a set of feature sets to be modelled. We start by describing the structure of **SILGEO** measurements. Next, in Section 3.1.1, we discuss the representations we use to encode our features and which features we deem redundant. Finally, in Section 3.1.2, we cover how we reduce the dimensionality of our feature space in preparation for statistical modelling.

For the purposes of our model, a **SILGEO** fingerprint of a device is a sample. As discussed in Section 2.3, a **SILGEO** fingerprint is composed of up to 64 sets of measurements. We name the measurements for a single port combination a network. Figure 3.1 visualizes the set of network objects we have in a dataset with 201 devices.

Each network is composed of four **S-parameters**, and each **S-parameter** is measured at 2500 frequencies. A **S-parameter** at a single frequency is a complex number which can be represented in many ways. Figure 3.2 visualizes the makeup of a network in which **S-parameters** are represented by their magnitude and group delay. Since one of the requirements for our method is to provide some granularity in the results of our novelty detector, we consider each network separately. We discuss the magnitude and group delay in the next section.

<u>Target Device</u>	<u>Pin Pair 1-2</u>	<u>Pin Pair 1-3</u>	<u>...</u>	<u>Pin Pair 7-16</u>
Z80-001	Network	Network	...	Network
Z80-002	Network	Network	...	Network
Z80-003	Network	Network	...	Network
⋮				
Z80-199	Network	Network	...	Network
Z80-200	Network	Network	...	Network
Z80-201	Network	Network	...	Network

Figure 3.1: Visualization of a dataset containing measurments of 200 z80 devices.

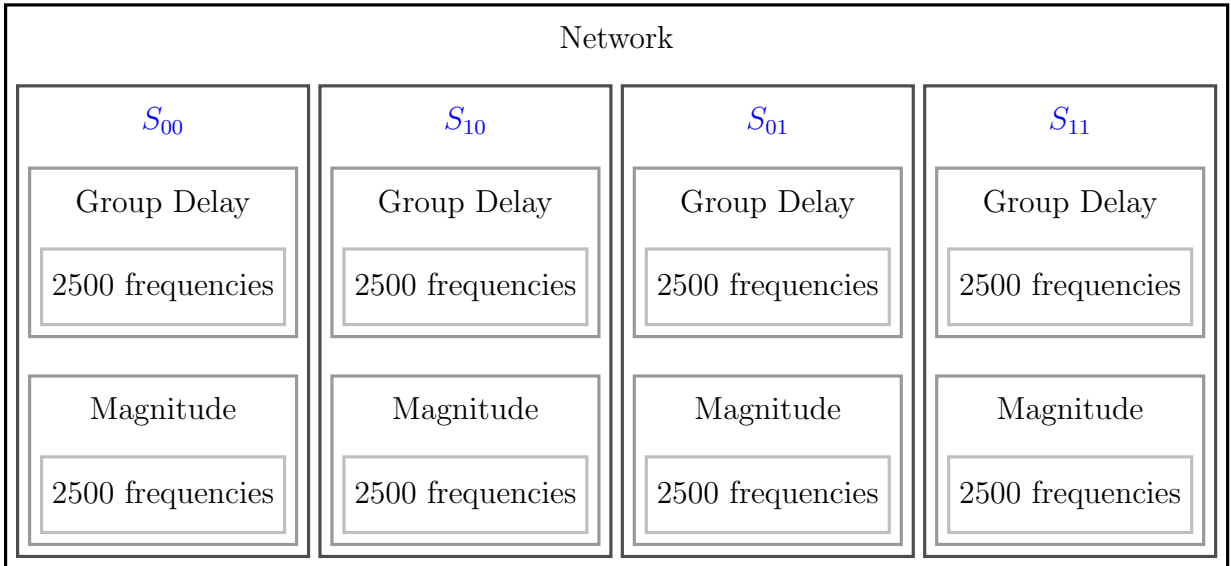


Figure 3.2: Visualization of the features in a single measurement of a pin pair by [SILGEO](#).

3.1.1 Feature Extraction

This section discusses how we partition and extract features from the [SILGEO](#) measurements described in the last section. The first task is to find a good representation for the complex values of the [S-parameters](#). A naive approach would be to use the linear magnitude and phase discussed in [Section 2.3](#). However, we want our model’s representations to be as simple as possible, as we have limited training data. To this end, we represent the gain with logarithmic magnitude instead of linear magnitude as gain values scale logarithmically. Representing the gain logarithmically allows the model to incorporate the difference between small changes at low gain levels without being overshadowed by large changes at high gain levels.

Table 3.1: Example phase delay and group delay at various frequencies in response to a frequency-invariant delay of 0.1 seconds.

Frequency	Phase Delay (Degrees)	Group Delay (Seconds)
1 Hz	36	0.1
1.1 Hz	39.6	0.1
1.2 Hz	43.2	0.1
1.3 Hz	46.8	0.1
1.4 Hz	50.4	0.1
1.5 Hz	54	0.1

Finding a good representation for the delay of the signal is more involved. Phase delay does not exhibit any scaling issues like gain did; however, phase delay can exhibit additive properties across frequencies. This is because phase delay is measured as an angle with respect to the frequency being observed. So, if the frequency increases, then so too must the angle for the same temporal delay. For example, consider a frequency-invariant delay of 0.1 seconds, then [Table 3.1](#) contains the phase delays at various frequencies. This is an issue as we really want to measure the new delays incurred at each frequency. To capture this information, we use the group delay instead. Group delay τ_g is defined in [Equation 3.1](#) with respect to unwrapped phase delay ϕ and the angular frequency ω . Note that angular frequency is defined as $\omega = 2\pi f$ where f is the frequency in Hertz. The unwrapped phase delay is the usual phase delay except that it is not bounded or wrapped between $-\pi$ and π . Since group delay is a partial derivative with respect to the frequency, we eliminate the

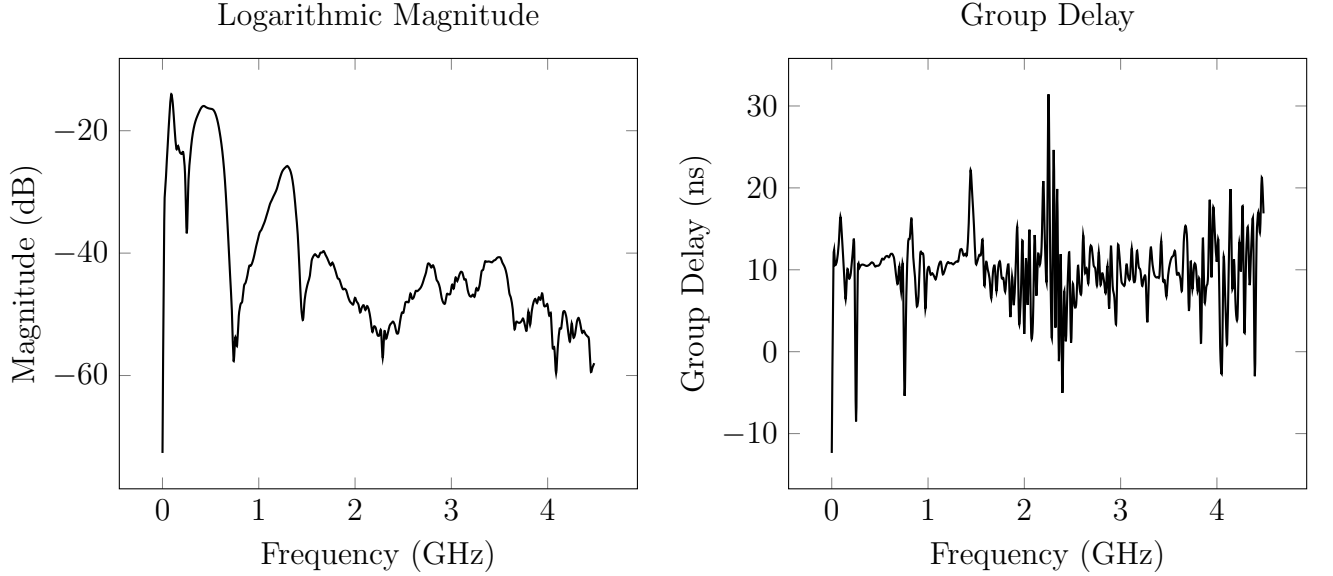


Figure 3.3: An example of a single [S-parameter](#) measurement of a single port combination.

additive properties across frequencies and only investigate the rate of change.

$$\tau_g(\omega) = -\frac{d\phi(\omega)}{d\omega} \quad (3.1)$$

Table 3.1 shows how the additive property of our example does not affect the group delay.

With the representation of our measurement fixed, we next remove redundant features. Removing redundant features is an important step, as we currently have millions of features and only expect to have dozens of samples in our training dataset. Two insights reveal that over half of the features are redundant. The first is that for any electrical port pair, the transmission [S-parameters](#) (S_{10} and S_{01}) should be practically identical; we consider any difference to be noise. Second, for every electrical port, all reflection [S-parameters](#) associated with that electrical port contains the same information about the [DUT](#). Any difference in the electrical port's reflection [S-parameters](#) is to the physical connections being made by the other electrical ports in the electrical port combinations and by noise. So, the entire fingerprint can be represented by a single reflection [S-parameter](#) for each electrical port and a single transmission [S-parameter](#) for each electrical port pair.

Going forwards, we create a model for each of the 80 mentioned [S-parameters](#) per

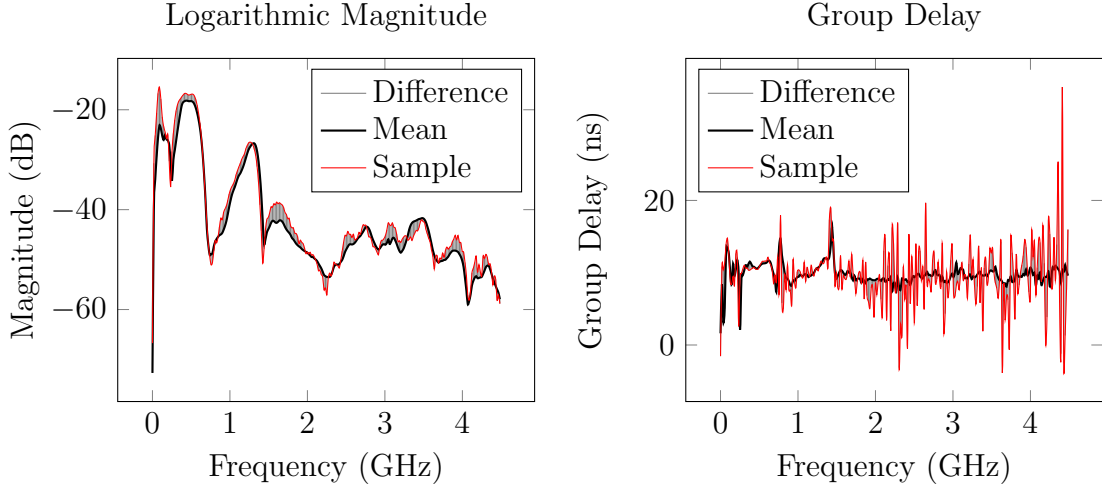


Figure 3.4: An example of a transmission [S-parameter](#) plotted in red against the mean of nominal measurements plotted in black. A vertical grey line denotes the difference between the two at each feature.

fingerprint. This enables the model results to indicate on which electric port or port pair an anomaly has been detected. Each of our [S-parameters](#) has 5 000 real-valued features after feature extraction, a group delay and a magnitude value for 2 500 frequencies. Figure 3.3 visualizes our representation of a single [S-parameter](#) in terms of magnitude and group delay, as discussed above.

3.1.2 Dimensionality Reduction

In this section, we describe how we reduce the dimensionality of the 5 000-dimension feature space created by our feature extraction technique into a feature space that statistical models can handle. Recall that one of the requirements of our method is to work with few (less than 50) samples in our training dataset; meanwhile, we currently have 5 000 features per [S-parameter](#). We aim to reduce dimensionality while preserving as much information as possible. Our dimensionality reduction method can reduce to different numbers of dimensions, which we denote as n in this section.

The following observations drive our dimensionality reduction method choices.

1. There is some correlation between the responses of frequencies near each other. An

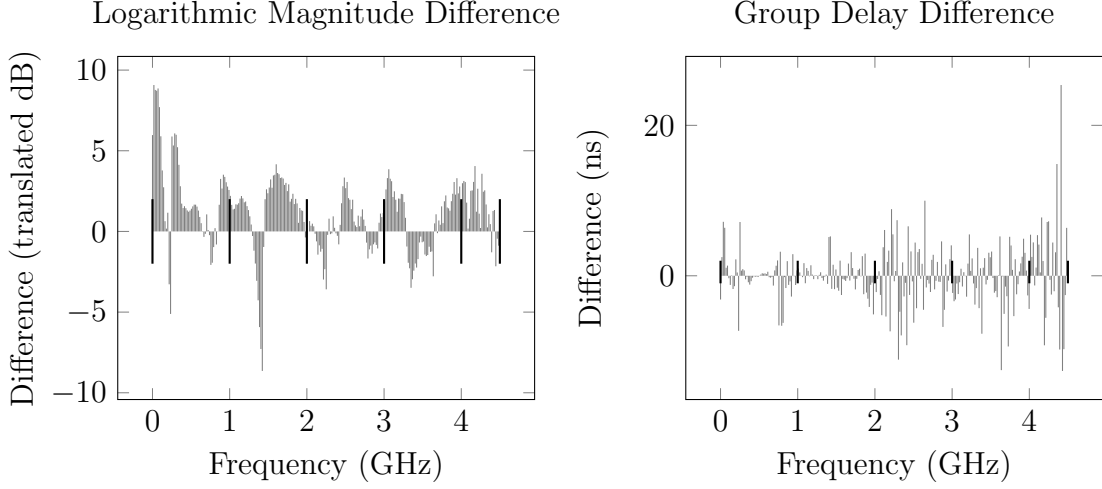


Figure 3.5: This figure contains the difference between the sample and mean depicted in Figure 3.4. The vertical black lines indicate how the features are partitioned.

example of this correlation is seen in the Logarithmic Magnitude response shown in Figure 3.4.

2. A massive deviation in a small frequency range may indicate an anomalous device.
3. Different frequency ranges exhibit different variations and noise properties. An example can be seen in Figure 3.4, where the group delay differences between the mean and the unseen sample are sporadic for high frequencies.

The difference between an unseen sample and the training dataset is the information we aim to preserve in our dimensionality reduction transformation. We are especially interested in preserving significant differences over small frequency ranges. Vibien’s approach [47] to dimensionality reduction was to measure the similarity between an unseen chip and the training dataset using dynamic time warping. We follow a similar method and base our dimensionality reduction method on the average pointwise difference from the mean of our nominal training [S-parameters](#). We chose the signed difference instead of the distance as it is easier to model statistically. Figure 3.4 visualizes the training dataset mean in black, the difference in gray, and an unseen sample in red for a transmission [S-parameter](#). Figure 3.5 shows the difference in grey directly of the same [S-parameter](#) as Figure 3.4 by subtracting the training mean. This method allows us to preserve significant differences over small frequency ranges while maintaining a transformation we can model statistically.

The next step is to account for the difference in the variation between different frequency

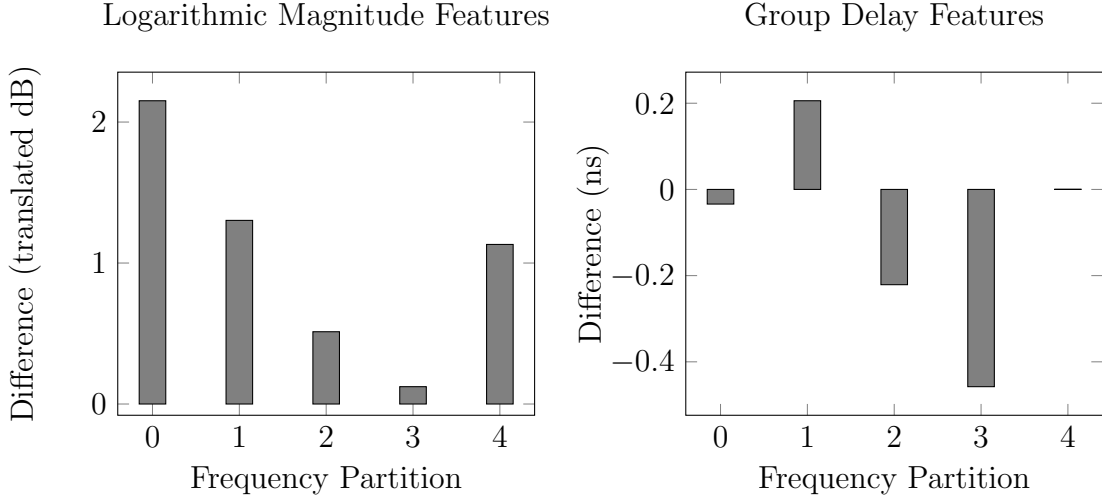


Figure 3.6: This figure contains the averages of the differences shown in Figure 3.5.

ranges. For example, high frequencies tend to vary significantly more than low-frequency ranges. High frequencies tend to have a lower signal-to-noise ratio due to the Free Space Path Loss Function [30], which describes how signal attenuation is proportional to the square of the frequency of a signal. This causes the high frequency ranges to disproportionately impact the DTW measure. To allow for the different variations of different frequency ranges, we partition the frequency space of each representation into $\lfloor n/2 \rfloor$ segments. Since we expect the response of similar frequencies to correlate, we partition our frequency space into intervals such that similar frequencies end up in the same partition. The partitions are indicated by the vertical black bars in Figure 3.5.

The resulting $\lfloor n/2 \rfloor * 2$ features are depicted by a bar graph in Figure 3.6. The main weakness of this dimensionality reduction method is that if a sample deviates above and below the mean inside of a single partition, then taking the signed mean of the deviations can cancel each other out. As the number of partitions n increases, the information lost is reduced, especially for frequency ranges that exhibit high autocorrelation, such as most logarithmic magnitude responses.

3.2 Statistical Model

This section describes our approach to creating a statistical model for novelty detection. We start by selecting a distribution we use to model the generating distribution of our

nominal training dataset. Then we discuss our prior information on the parameters for our generating distribution. In the next section, we show how to incorporate domain knowledge into our generating distribution. Finally, we discuss how we create a test statistic for a novel sample for our distribution based on a generalization of a Student’s T-test.

We first frame the novelty detection problem as a hypothesis test to guide our statistical analysis. Our null hypothesis is: the mean of the distribution generating a novel sample in question is the same as that of the distribution that generated the training samples.

3.2.1 Generating Distribution

When choosing a distribution to model our generating distribution, there are a lot of considerations to make. Non-parametric models may have limited differentiating power given the limitation on our sample size and the high dimensional nature of the data. They may also make it difficult to encode domain knowledge into the model. For these reasons, we model our distribution directly.

Our dataset consists of a single fingerprint per device on a set of devices made to be as similar as economically possible. Since the set of devices are manufactured to be as similar as economically possible, we expect the fingerprints of the devices to also be similar. For this reason, we assume that our data has a single mode to restrict our model complexity. However, there are several sources of variability in our measurements:

- Variation inherent to the devices being measured. This variation is expected to come from variations in the manufacturing process.
- Variation due to reconnection. When we put our devices in the [SILGEO](#) sockets, the fit is not exact, and there is room for the device to sit in slightly different positions. We expect this to cause some variation between the measurements of the device.
- Variation in the state of the devices under test. We expect the devices’ natural aging to slightly change their characteristics.
- Variation in the measurement setup. We expect variations between measurements from different instances of the [SILGEO](#) measurement setup and device. This is due to slight variations in the manufacturing of the multiplexor and adapter boards through which the rf signal needs to travel to reach the device.
- Variation in the measurement environment. We expect the properties of the background electromagnetic field in which the measurements are being taken to have some effect on the measurements, however slight.

Visualization of a Two Dimensional Multivariate Normal Distribution

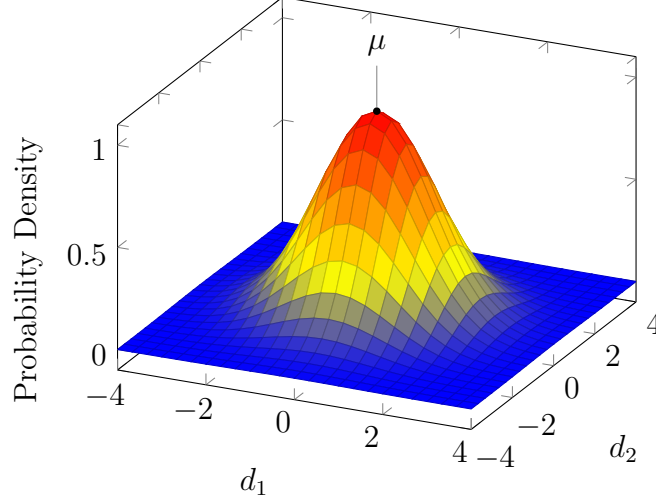


Figure 3.7: A two dimensional multivariate normal distribution with a mean at $[0, 0]$ and the identity matrix as the covariance matrix. The x and y axes denote the space on which the distribution is defined, and the z axis shows the probability density.

The measurement variability is the effect of countless interactions within the [DUT](#) and with the electromagnetic properties of the environment during the test. The central limit theorem suggests that we can model a feature with a Gaussian distribution if the culmination of the mentioned interactions is additive and if the interactions are independent. We assume that this is the case and use the Gaussian distribution for each feature. The independent interactions assumption can be slightly relaxed to weak dependence using extended versions of the central limit theorem; however, this work was not done for this thesis.

We only consider linear correlations between features for simplicity, so we model our entire reduced-dimensional feature vector as a multivariate normal distribution. A multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ has two parameters, the mean $\boldsymbol{\mu}$, and the covariance matrix $\boldsymbol{\Sigma}$. Figure [3.7](#) shows an example of a two-dimensional multivariate normal distribution.

3.2.2 Prior Covariance Matrix Knowledge

This section discusses our prior knowledge of the covariance matrix of the generating distribution. Our domain knowledge comes from the observation that all fingerprints contain multiple observations of **S-parameters**. There is structure in the distribution of **S-parameters** in general. Specifically, we can estimate the autocorrelation between our features for **S-parameters** in general. We can also estimate the variance of each of our features for **S-parameters** in general.

We do this by computing the mean correlation matrix for all other **S-parameters** in our training dataset (after removing redundant **S-parameters**). We then take the mean $\mathbf{C}$ of these correlation matrices as the initial estimate for the correlations between features of the generating distribution for the specific **S-parameter** in question. For our initial estimate of the variance of a feature i , we are more conservative, and we take the maximum σ_i of the sample variance of the feature for all other **S-parameters** in our training dataset (again after removing redundant **S-parameters**).

To construct our initial covariance matrix estimate, we use the relation $\text{Corr}(x, y) = \frac{\text{Cov}(x, y)}{\sigma_x \sigma_y}$. Using this relationship, we construct our covariance matrix by scaling our correlation matrix estimate $\mathbf{C}$ by our variance estimates $\sigma_1^2, \dots, \sigma_n^2$. Note that covariance matrices are positive semi-definite, and each sample correlation matrix is positive semi-definite. Since the space of positive semi-definite matrices is closed under addition and scaling by positive real numbers, we have that $\mathbf{C}$ is semi-positive definite. Our covariance estimate is defined as follows:

$$\begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix} \mathbf{C} \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix} \quad (3.2)$$

Note that our covariance estimate is also positive semi-definite as diagonal matrices with non-negative entries are positive semi-definite, and positive semi-definite matrices are closed under matrix multiplication.

3.2.3 Bayesian Inference

With a multivariate normal distribution as our generating distribution and a mean-based hypothesis test, the typical test statistic is the multivariate two-sample Students T-test.

This test, also known as the two-sample Hotelling’s T-test, is a direct generalization of the univariate two-sample Students T-test [31]. This test uses the Mahalanobis distance as a test statistic. Unfortunately, we can not incorporate our prior covariance matrix estimate into this test.

To incorporate our knowledge into our model, we use Bayesian inference. More specifically, we use Bayesian inference to create a robust estimation of the generating distribution of our nominal training dataset that takes into account prior information and uncertainties which we discuss in this section. We then use posterior predictive p -values [33] on a generalized version of Hotelling T-test to determine if we should reject our null hypothesis. Posterior predictive p -values are a way to do p -value based hypothesis tests on Bayesian models by computing p -values of a test statistic on the posterior predictive distribution.

Bayesian methods allow us to incorporate prior knowledge into our model by using prior distributions to express our beliefs about the models’ parameters. The training samples then update these beliefs, resulting in a final model used for inference. Incorporating prior knowledge into our model allows us to apply our statistical model in higher dimensions.

To use a prior estimate on our covariance matrix, we need to use Bayesian methods to learn from our training dataset. Since we may be working with high-dimensional spaces, we want a closed-form solution for updating our parameters since distribution estimation techniques for high-dimensional spaces are limited. To get a closed-form solution, we must use conjugate priors to represent our prior beliefs. The conjugate prior for a multivariate normal distribution parameterized by an unknown mean and covariance matrix is the inverse Wishart distribution $\mathcal{IW}(\Psi, v)$ [12]. Note that the covariance matrix prior already restricts us to symmetric positive definite matrices resolving our singular maximum likelihood estimate of the covariance matrix issue. This distribution is parameterized by our “best guess” for the sample covariance matrix Ψ_0 , and a parameter representing our belief level in our “guess” v_0 . The parameter v_0 of the inverse Wishart distribution is called the degrees of freedom and dictates how heavy the distribution’s tails are. The parameter Ψ is called the scale matrix, and $\frac{\Psi}{v-d-1}$ is the mean of the inverse Wishart distribution $\mathcal{IW}(\Psi, v)$ in d dimensions.

With our priors and training dataset, we can then update the distributions that represent our knowledge of the parameters of the multivariate normal distribution. We denote our initial prior parameters as below:

1. v_0 with $d - 1 < v_0$ - degrees of freedom of the inverse Wishart Distribution
2. Ψ_0 with Ψ_0 as a positive definite matrix - scale matrix
3. μ_0 - prior mean

4. κ_0 with $0 < \kappa_0$ - number of measurements used to obtain prior mean (these measurements must have been taken when the covariance follows Σ)

With these initial prior parameters, our initial prior distributions are as follows:

$$\Sigma \sim \mathcal{IW}(\Psi_0, v_0) \quad (3.3)$$

$$\mu|\Sigma \sim \mathcal{N}(\mu_0, \frac{\Sigma}{\kappa_0}) \quad (3.4)$$

The hierarchical prior probability distribution is comprised of our prior distributions, and the multivariate normal distribution is the following multivariate t distribution:

$$p(\tilde{y}|v_0, \Psi_0, \kappa_0, \mu_0) = t_{v_0-d+1} \left(\mu_0, \Psi_0 \frac{\kappa_0 + 1}{\kappa_0(v_0 - d + 1)} \right) \quad (3.5)$$

Given n new samples $D = y_1, \dots, y_n$, we want to update our beliefs. The magic of conjugate priors is that the posterior distributions $p(\Sigma|D)$ and $p(\mu|D)$ belong to the same respective probability distribution families as the priors $p(\Sigma)$ and $p(\mu)$. The following equations show how to calculate the parameters μ_n, κ_n, v_n , and Ψ_n of $p(\tilde{y}|v_0, \Psi_0, \kappa_0, \mu_0)$ [35] [21] [5].

$$\mu_n = \frac{\kappa_0}{\kappa_0 + n} \mu_0 + \frac{n}{\kappa_0 + n} \bar{y} \quad (3.6)$$

$$\kappa_n = \kappa_0 + n \quad (3.7)$$

$$v_n = v_0 + n \quad (3.8)$$

$$\Psi_n = \Psi_0 + \sum_{i=1}^n (y_i - \bar{y})(y_i - \bar{y})^T + \frac{\kappa_0 n}{\kappa_0 + n} (\bar{y} - \mu_0)(\bar{y} - \mu_0)^T \quad (3.9)$$

So, the posterior distribution is:

$$p(\tilde{y}|D, v_0, \Psi_0, \kappa_0, \mu_0) = t_{v_n-d+1} \left(\mu_n, \Psi_n \frac{\kappa_n + 1}{\kappa_n(v_n - d + 1)} \right) \quad (3.10)$$

3.2.4 Test Statistic

Given a posterior distribution, we must derive a test statistic for our hypothesis test. We model our generating distribution as a normal distribution. With a normal distribution, we would use the established Hotelling's T-test. However, incorporating our prior knowledge and uncertainties results in a multivariate T distribution. We still need a test statistic to use posterior predictive p -values. Fortunately, the Hotelling's T-test can be generalized to multivariate T distributions as the multivariate T distribution is an elliptical distribution. In this section, we argue why our Mahalanobis distance-based test statistic works for our elliptical posterior distribution and derive the posterior distribution on our test statistic. For background on how a similar argument can be used to derive the Hotelling's T-test, see Appendix A.

First, we define the squared Mahalanobis distance on the multivariate t distribution as $d_{t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(x) = (x - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu})$. Note that the probability density function of the multivariate t distribution is defined as

$$f_{t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(x) = \frac{\Gamma[(v+d)/2]}{\Gamma(v/2)v^{d/2}\pi^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \left[1 + \frac{1}{v}(x - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu}) \right] \quad (3.11)$$

As with the multivariate normal distribution, the above equations show that the probability density function of the multivariate t distribution decreases monotonically as the squared Mahalanobis distance increases. Consider $\boldsymbol{x}, \boldsymbol{y} \in S$ with $d_{t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\boldsymbol{x}) \leq d_{t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\boldsymbol{y})$, then

$$f_{t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\boldsymbol{x}) = \frac{\Gamma[(v+d)/2]}{\Gamma(v/2)v^{d/2}\pi^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \left[1 + \frac{1}{v}(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \right] \quad (3.12)$$

$$= \frac{\Gamma[(v+d)/2]}{\Gamma(v/2)v^{d/2}\pi^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \left[1 + \frac{1}{v}d_{t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\boldsymbol{x}) \right] \quad (3.13)$$

$$\leq \frac{\Gamma[(v+d)/2]}{\Gamma(v/2)v^{d/2}\pi^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \left[1 + \frac{1}{v}d_{t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\boldsymbol{y}) \right] \quad (3.14)$$

$$= \frac{\Gamma[(v+d)/2]}{\Gamma(v/2)v^{d/2}\pi^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \left[1 + \frac{1}{v}(\boldsymbol{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) \right] \quad (3.15)$$

$$= f_{t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\boldsymbol{y}) \quad (3.16)$$

Since $f_{t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})}$ decreases monotonically as $d_{t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})}$ increases, the sets W_t defined by the squared Mahalanobis distance $W_t = \{x \in S : d_{t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(x) < t\}$ have minimal size $|W_t|$ for the portions of the multivariate t distribution they contain.

We have that if X is a random variable in S that follows $t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $d_{t_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(X)$ follows an F distribution, specifically $\frac{1}{d}\mathcal{F}(d, v_n - d + 1)$ [36]. Note that the squared Mahalanobis distance on our posterior distribution can be rewritten as below:

$$d_{t_{v_n-d+1}(\boldsymbol{\mu}_n, \boldsymbol{\Psi}_n \frac{\kappa_n+1}{\kappa_n(v_n-d+1)})}(x) = (x - \boldsymbol{\mu}_n)^T \left(\boldsymbol{\Psi}_n \frac{\kappa_n+1}{\kappa_n(v_n-d+1)} \right)^{-1} (x - \boldsymbol{\mu}_n) \quad (3.17)$$

$$= (x - \boldsymbol{\mu}_n)^T \left(\boldsymbol{\Psi}_n^{-1} \frac{\kappa_n(v_n-d+1)}{\kappa_n+1} \right) (x - \boldsymbol{\mu}_n) \quad (3.18)$$

$$= \frac{\kappa_n(v_n-d+1)}{\kappa_n+1} (x - \boldsymbol{\mu}_n)^T \boldsymbol{\Psi}_n^{-1} (x - \boldsymbol{\mu}_n) \quad (3.19)$$

And the Mahalanobis distance of a random variable X following our posterior distribution has the following distribution:

$$\frac{1}{d} d_{t_{v_n-d+1}(\boldsymbol{\mu}_n, \boldsymbol{\Psi}_n \frac{\kappa_n+1}{\kappa_n(v_n-d+1)})}(X) \sim \mathcal{F}(d, v_n - d + 1) \quad (3.20)$$

So, similar to Appendix A, we can find a threshold t based on a target false positive rate p as follows:

$$1 - p = F_{\mathcal{F}(d, v_n-d+1)}(t) \quad (3.21)$$

$$t = F_{\mathcal{F}(d, v_n-d+1)}(1 - p)^{-1} \quad (3.22)$$

3.2.5 Determining Hyper Parameters

In this section, we discuss the hyperparameters for our model. We describe how we set our hyperparameters based on the training dataset and explain the reasoning behind our choices. In this section, we face our model's main challenge: the curse of dimensionality. Preventing overfitting while allowing as complex a model as possible motivates our hyperparameter selection.

We begin by describing the curse of dimensionality and how it comes about in our application. The curse of dimensionality with respect to machine learning refers to the interaction between the following three variables:

1. The number of features.

2. The number of samples from which we can learn.
3. The complexity of our model.

Specifically, the curse of dimensionality refers to the fact that with fixed model complexity, exponentially more data is required to generalize accurately as the number of features increases. This pattern is also referred to as the peaking phenomenon [46]. Novelty and outlier detection in high dimensional spaces is particularly challenging [4] [32]. Since we anticipate several orders of magnitude more features than samples, the curse of dimensionality applies to our problem. The main effect of the peaking effect is the risk of overfitting. We can reduce our risk of overfitting by reducing the number of features we consider, reducing the complexity of our model, or increasing our sample size. Unfortunately, there is a trade-off to these methods of reducing overfitting. Reducing the complexity of our model and the number of features we consider typically also reduces the model’s ability to differentiate between nominal and abnormal samples. We aim to select hyperparameters of our model that maximize the model’s ability to differentiate between nominal and abnormal samples while minimizing the risk of overfitting.

The hyperparameters of our model are:

- d the number of features we reduce to using the dimensionality reduction technique discussed in Section 3.1.2.
- p the target false positive rate.
- v_0 the degrees of freedom in the covariance matrix prior distribution.
- κ_0 the weight of the prior mean.
- Ψ_0 the shape matrix of our covariance matrix prior distribution.
- μ_0 the mean of the prior distribution for the mean.

The estimate of the mean of a multivariate normal distribution is not at risk of overfitting as the error only depends on the variance of the true multivariate normal distribution as opposed to the dimensionality of the space we consider. This means that the mean acts more like a single parameter than d parameters, and so does not contribute significantly to the complexity of our model. The covariance matrix, on the other hand, has $d(d+1)/2$ parameters and is the main source of complexity in our model.

The parameters that describe the prior for the mean, namely κ_0 and μ_0 , have no significant effect on the complexity of our model. We pick an uninformative prior for the mean such that the mean of our posterior distribution is solely determined by our training samples D . We do this by taking the limit of the conjugate prior density as $\kappa_0 \rightarrow 0$, which

makes the μ_0 parameter irrelevant as it no longer influences our posterior distribution. Setting an uninformative prior on the mean will cause our posterior distribution to have heavier tails which will reduce the statistical power of our posterior predictive test. Note that when $\kappa_0 \rightarrow 0$ we are approaching a limit on the value of κ_0 we need to be careful that our posterior distribution is proper, Gelman et al. [21] gives the precedent for this choice.

The parameters that describe the prior for the covariance matrix, namely Ψ_0 and v_0 , significantly affect our model's complexity. Ψ_0 determines the shape of the mode $\frac{\Psi_0}{v+d+1}$ of the prior distribution for the covariance matrix. So Ψ_0 represents our 'best guess' for the covariance matrix. v_0 is the degrees of freedom of the prior distribution of the covariance matrix. If we set $\Psi_0 = v_0 S_0$, then this prior is equivalent to v_0 samples (with sample covariance matrix S_0) added to the estimate of the covariance matrix [52]. We set $\Psi_0 = v_0 S$ where S is the matrix outlined in Section 3.2.2.

Note that Ψ_0 is not invertible for any dimensionality $d > (k-1)(n-1)$, where k is the number of considered [S-parameters](#) and n is the number of devices in our dataset. Since Ψ_0 must be invertible to be a valid scaling matrix, we have an upper limit on our dimensionality. Since our prior doesn't accurately determine the covariance structure of any individual feature set, we want to give it as small of a weight as possible. So we set $v_0 = d$; this effectively adds $(k-1)(n-1)$ samples to our covariance matrix estimate but with the weight of only d samples.

The parameter d is already limited to be less than $(k-1)(n-1)$. Since $v_0 = d$ and we wish the samples to have more influence over the covariance matrix than our prior estimate, we set the maximum of d to be n . There are no other limitations on d , and we want to maximize our differentiation ability, so we set $d = n$ where n is the number of samples in our training dataset. It is suggested that further work be done to determine the optimal value of d based on the Vapnik-Chervonenkis Dimension of the model, which measures its complexity [46] [29].

The final parameter p represents the target false positive rate of a model on a single feature vector. To set this parameter, we need to consider the desired false positive rate of the entire model, which depends on the method used to combine results from different [S-parameters](#). The following section discusses combining sub-model results and setting p .

3.2.6 Model Aggregation

To get a final result on a [DUT](#), we need to label the [DUT](#) based on the classification of each measured electrical port and electrical port pair. To do this, we simply classify the [DUT](#) as anomalous if any electrical port or electrical port pair is classified as anomalous.

Ideally, we would like to set the expected false positive rate of each sub-model such that the model as a whole has a desired expected false positive rate. However, we do not expect the **S-parameters** to behave independently, even among nominal devices. For example, we expect anomalies on transmission **S-parameter** from electrical port 1 to electrical port 7 to be positively correlated with anomalies on the reflection **S-parameters** on each port in the pair.

This causes an issue in setting p for sub-models as the probability of the transmission **S-parameter** A or the electrical port 1 reflection **S-parameter** B being a false positive is:

$$P(A \vee B) = P(A) + P(B) - P(A)P(B|A) \quad (3.23)$$

So to determine the probability of A or B happening, we need to know the correlation between A and B . We do not know the correlation between A and B ahead of time, however we assume that it will be positive in this case. Furthermore, the correlation also changes between different types of devices and so has to be learned per device. We avoid this issue by computing the false positive rate for each sub-model such that the expected false positive rate for the whole model is lower than the selected one. To do this, we compute the p-value for each sub-model as though all sub-models are independent. In the case of our probability example, we have the following:

$$P(A \vee B) = P(A) + P(B) - P(A)P(B|A) \quad (3.24)$$

$$\geq P(A) + P(B) - P(A)P(B) \quad (3.25)$$

$$(3.26)$$

So if we set the values of $P(A)$ and $P(B)$ such that $P(A) + P(B) - P(A)P(B)$ falls below a given limit, then we have that $P(A \vee B)$ must also fall below the said limit.

To scale this up to all **S-parameters**, we must determine the probability that at least one of n equally likely independent A_i events occur.

$$P(A_1 \vee A_2 \vee \dots \vee A_n) = 1 - P(\neg A_1 \wedge \neg A_2 \wedge \dots \wedge \neg A_n) \quad (3.27)$$

$$= 1 - P(\neg A_1)P(\neg A_2)\dots P(\neg A_n) \quad (3.28)$$

$$= 1 - (1 - p)^n \quad (3.29)$$

So, given a maximum false positive rate m for the whole model, we compute the desired false positive rate for each sub-model as follows $1 - (1 - m)^{\frac{1}{n}}$ where n is the number of **S-parameters** in consideration.

So, for example, consider the case that we have 80 modelled **S-parameters**, and we have a target maximum false positive rate of 5% for the whole model. Then we should set the expected false positive rate of each sub-model to be $1 - (1 - 0.05)^{\frac{1}{80}} \approx 0.00064$

3.3 Summary

This chapter introduced our model for novelty detection. The model provides expected false positive rates despite train on datasets containing only examples of nominal measurements through statistical arguments resting on a minimal set of assumptions on the generating distribution of our samples. The model can detect anomalies for specific pins and pin combinations through its segmented design. The model works with training datasets containing few samples through its simplicity and exploiting domain knowledge of the feature set.

Chapter 4

Experimental Evaluation

In this section, we assess the performance of our method. Specifically, we are interested in how our method does the following:

1. Is the observed false positive rate lower than the maximum expected false positive rate throughout our test datasets?
2. To what extent can our method detect counterfeit or malicious devices in different types of hardware? In other words, what is the model’s sensitivity or true positive rate in different device classes?
3. How does our model perform when given as few as three samples and as many as 42 samples during training?

To answer these questions, we present three case studies. The first case study was conducted on the same FT232RL devices as the case study presented by Vibien and is used to contrast the new method with existing work. The second case study was conducted on Z80 chips, a microprocessor that is a step up in complexity from the FT232RL chips. The third case study concerns the detection of seven different USB attack tools. The USB attack tools are fully assembled devices as opposed to isolated integrated circuits.

4.1 FTDI FT232RL Case Study

FT232RL devices are USB 2.0 to universal asynchronous receiver-transmitter converter microchips. Vibien conducted a case study on FT232RL devices because they are common,

inexpensive, and a known target for counterfeiting [47]. We expand on the study done by Vibien using our novelty detector to demonstrate the additions our technique offers.

For our study, we use the same dataset that Vibien used. This dataset consists of 100 suspected authentic devices, 119 suspected counterfeit devices, and 41 suspected recycled devices. The set of devices were collected to maximize their diversity. Devices were obtained from nine vendors, including Digikey, Mouser, eBay vendors, and AliExpress vendors. Devices were ordered in three batches for each Digikey and Mouser vendor. The status of each device was determined by visual inspection, functional tests, and X-ray imaging of the internal die and lead frame [47].

Vibien developed an anomaly detection method which he applied to counterfeit and authentic devices. Based on four authentic devices, Vibien developed a similarity score. For each [S-parameters](#), including those that we deem redundant, Vibien computes a DTW distance from a new device to the four reference devices. The average of the DTW distances for all [S-parameters](#) is then taken to compute the similarity score. Finally, to allow for classification, a threshold was manually determined. Using this method, Vibien correctly classified all devices in the dataset based on only his four reference devices.

In comparison, the method presented in this paper allows for the following additional features:

1. The method attempts to remain under a given false positive rate.
2. When anomalies devices are detected, we identify which electrical port and port combinations resulted in the anomalous classification.
3. The threshold is automatically determined based on the properties of the given dataset and a given maximum expected false positive rate.

Table 4.1: Confusion matrix showing the classification of the novelty detection method applied to the FT232RL dataset when trained on four devices. A maximum expected false positive rate of 5% was selected for this classification.

	Predicted Nominal	Predicted Anomalous
Suspected Authentic	95	1
Suspected Counterfeit	42	76

	Predicted Nominal	Predicted Anomalous
Suspected Recycled	26	15

We also trained on four randomly selected suspected authentic FT232RL devices to compare our model directly to Vibien’s approach. We set the maximum expected false positive rate to 5%, and we obtain the results summarized in Table 4.1. The expected false positive rate is about 1% which is well below our model’s maximum expected false positive rate. However, our sensitivity for suspected counterfeit chips is about 64%, well below Vibien’s 100%. This is due to our model’s conservative automatic threshold placement. To achieve similar performance to Vibien, we need more training samples to help characterize the nominal generating distribution.

Table 4.2: Confusion matrix showing the classification of the novelty detection method applied to the FT232RL dataset when trained on five devices. A maximum expected false positive rate of 5% was selected for this classification.

	Predicted Nominal	Predicted Anomalous
Suspected Authentic	92	3
Suspected Counterfeit	8	110
Suspected Recycled	18	23

We run our method on the same dataset using five and ten randomly selected training samples to demonstrate how our model performs as we increase the number of training samples. We selected the same maximum expected false positive rate of 5% for these experiments. Table 4.2 shows the results for five training samples. By adding only one additional training sample, our sensitivity on suspected counterfeit increased from 64% to 93% while the false positive rate devices raised to only 3%, staying under our maximum expected false positive rate of 5%. Table 4.3 gives the results for ten training samples. We get 100% sensitivity on counterfeit and recycled devices with ten training samples, while the false positive rate remains below 5%.

Table 4.3: Confusion matrix showing the classification of the novelty detection method applied to the FT232RL dataset when trained on ten devices. A maximum expected false positive rate of 5% was selected for this classification.

	Predicted Nominal	Predicted Anomalous
Suspected Authentic	88	2
Suspected Counterfeit	0	118
Suspected Recycled	0	41

While we don’t achieve the 100% correct classification that Vibien’s method achieves, we train on a similar number of samples and automatically select the threshold. Our false positive rate remains under the predicted limit, and we also achieve 100% sensitivity on the suspected recycled devices, which Vibien does not report.

Table 4.4: Sum of Anomalous Labels by the Novelty Detector on the FT232RL dataset for all electrical ports.

Electrical Port	Number of Anomalous Ports Found
1	2
2	6
3	52
4	43
5	13
6	11
15	49
16	6
7	45
8	44
9	44
10	78
11	3
12	78
13	8
14	5

In addition to the device level classification, we have the results on an electrical port and port combination basis. These results are displayed in Table 4.4 and Table 4.5, respectively, for the previous application in which we trained on ten devices. In the tables, we sum the number of times each electrical port and port combination is labelled as anomalous by the novelty detector. For example, anomalies were only found twice in the reflections on electrical port 1, whereas 122 anomalies were found in the transmissions from electrical port 6 to electrical port 8. This information may be helpful for investigating the nature of each anomaly detected. It also indicates which electrical ports are most important to inspect when looking for counterfeit or recycled devices.

Table 4.5: Number of anomalous electrical port combinations found by the Novelty Detector on the FT232RL dataset for all electrical port combinations. Note that the first row and column indicate the electrical port combination.

	1	2	3	4	5	6	15	16
7	81	49	38	62	77	77	53	2
8	85	52	51	68	55	122	85	6
9	87	80	84	85	85	83	84	82
10	45	73	81	88	74	54	83	82
11	56	84	87	82	83	86	87	84
12	55	81	49	13	54	54	80	55
13	85	82	56	60	59	82	93	56
14	84	73	79	53	48	81	80	37

We can also visualize the expected distribution of nominal devices against the actual distribution of devices in the test set on our test statistic. Furthermore, we can normalize the results of each [S-parameters](#) to overlay their results on the same plot. This information is visualized in Figure 4.1 for the previous model trained on ten devices. The x-axis is the test statistic, so it is the Mahalanobis distance from the mean of the training samples. Note that the x-axis is logarithmic, so the entire distribution of the authentic and counterfeit chips can be seen in the same graph. The black line is the posterior predictive distribution of nominal samples on the test statistic. The y-axis on the left shows the probability density of the expected distribution. To visualize the distribution of the test dataset, there is a histogram in green for suspected authentic FT232RL [S-parameters](#) and a histogram in red for counterfeit and recycled FT232RL [S-parameters](#). The y-axis on the right shows the

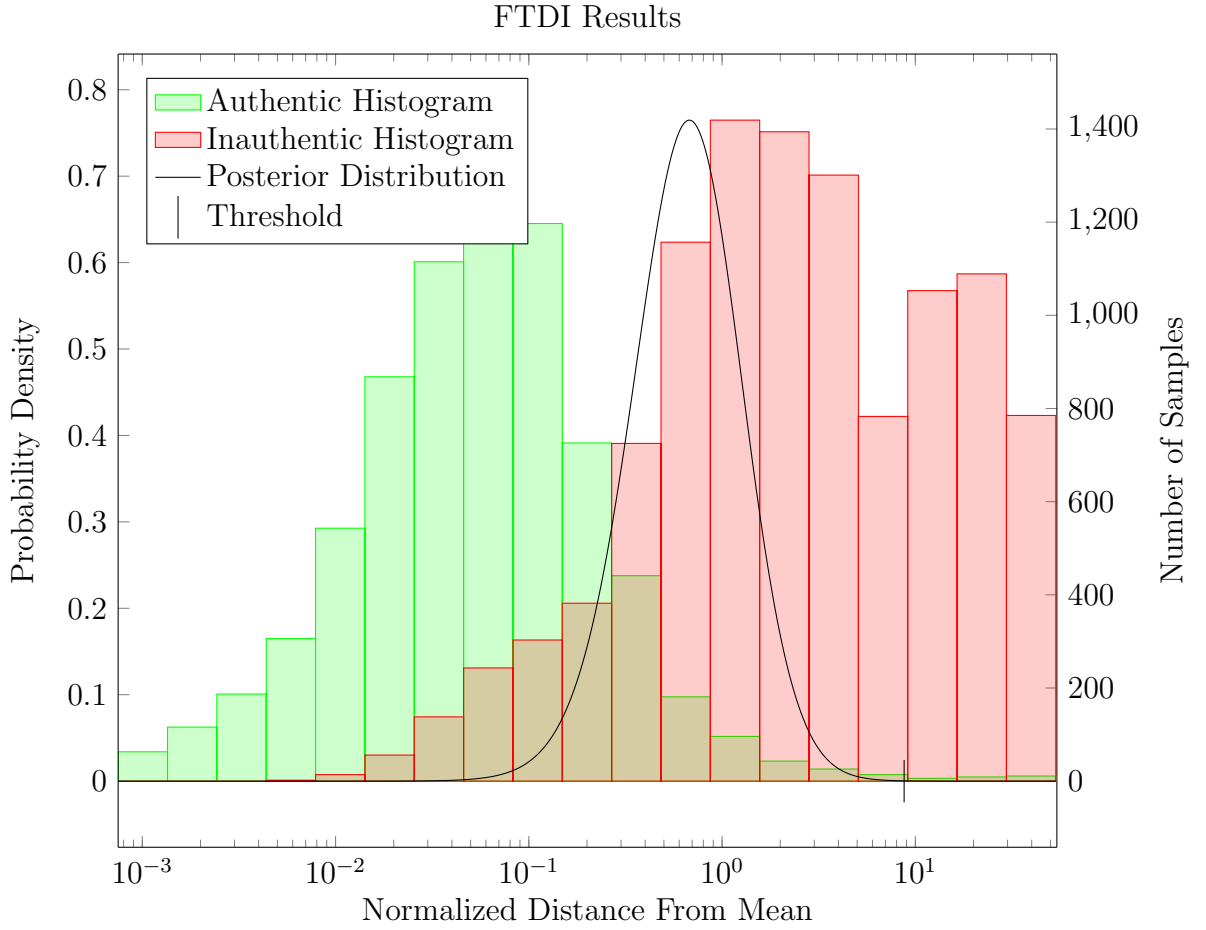


Figure 4.1: A visualization of the distribution of FT232RL experiment test set with respect to the test statistic.

number of samples in each histogram bin. The vertical black mark indicates the threshold for a maximum expected false positive rate of 5%.

Figure 4.1 shows that our posterior predictive distribution does not match that of the suspected authentic devices in the test dataset. This can be seen by the posterior predictive distribution (black line) to the right of the histogram of the suspected authentic devices in green. Since the posterior predictive distribution is to the right of the authentic measurements, the posterior distribution expects higher distances from the mean than is seen in the test dataset. This is expected as we overestimate the variance in our covariance matrix prior.

Figure 4.1 also reveals the extent to which the model separates measurements from suspected authentic devices from suspected counterfeit and recycled devices. This can be seen by the extent to which the red and green histograms are separated. Note that since we are overlaying the measurements on all considered [S-parameters](#), and since only one [S-parameter](#) needs to be classified as anomalous for the entire chip to be classified as anomalous, we have that the counterfeit chip associated with a measurement that has a low distance from the mean may still be classified as anomalous on some other [S-parameter](#).

In conclusion, this case study has shown that our method provides comparable results to the manual method created by Vibien. Furthermore, we indicate how the model’s sensitivity increases as the number of training samples increases while maintaining false positive rates on the test dataset below the maximum expected false positive rate.

4.2 Zilog Z80 Case Study

In the second case study, we studied the Z80 microprocessor created by Zilog. The Z80 is a popular single-chip microcontroller from 1976. Unlike the FT232RL integrated circuits, which were surface-mount, the Z80 integrated circuits in this case study were in the through-hole DIP-40 packaging. The different packaging meant the Z80 devices were much larger than the FT232RL devices.

For this case study, 200 Z80 integrated circuits were used. 60 Z80 devices were ordered from Mouser and Digikey, and 140 were collected from various Ali-Express vendors. The devices were labelled by visual inspection and X-ray analysis into 61 suspected authentic and 139 suspected inauthentic devices.

Table 4.6: Confusion matrix showing the classification of the novelty detection method applied to the Z80 dataset. For the results in this table, a maximum expected false positive rate was set to 5% and 40 devices were used for training.

	Predicted Nominal	Predicted Anomalous
Suspected Authentic	21	0
Suspected Counterfeit	14	128

For this section, we trained our novelty detector on 40 randomly selected suspected authentic devices. We then applied the trained model with a maximum expected false positive rate of 5%. Table 4.6 shows the resulting confusion matrix. The false positive rate is 0%, and the sensitivity is 90%. Note that with a low enough maximum expected false positive rate, we can achieve 100% correct classification.

As in the previous section, we finish with a visual investigation into the estimated and test set distributions. Figure 4.2 displays the expected and usual distribution of the model trained on 40 training samples in the same format as Figure 4.1. We can see the result of increasing the number of training samples in the shape of the posterior predictive distribution. Since more training samples were used, the model’s estimate for the maximum variance of each feature increased in accuracy and deviated further from the actual variance of each feature. The impact of the conservative estimate of the nominal samples’ variance on the model’s sensitivity is offset by the increased differentiation ability of the model due to the increased dimensionality. In other words, as the dimensionality of our model increases, we scale up the confidence in the prior covariance matrix estimate linearly. This results in the narrow posterior predictive probability density function in Figure 4.2. The median of the posterior predictive density on the test statistic should always lie at the edge or on the outer edge of the distribution of nominal samples as you increase the number of training samples. This is how we obtain a sensitivity of 90% while maintaining confidence in the maximum expected false positive rate.

4.3 Hardware Attack Tools Case Study

This case study concerns itself with hardware attack tools. This case study aims to differentiate between a set of approved devices and a hardware attack tool disguised to look like

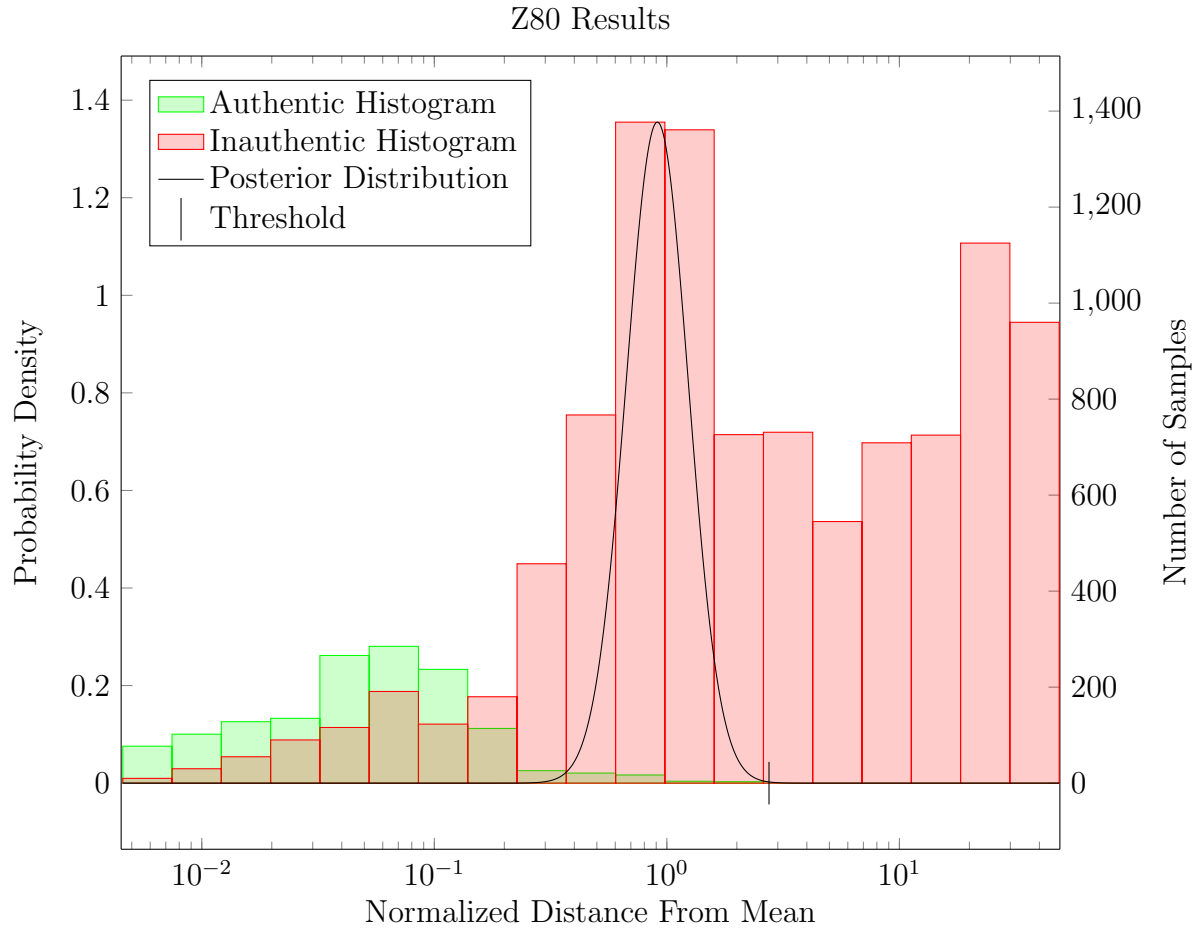


Figure 4.2: A visualization of the distribution of Z80 experiment test set with respect to the test statistic.

the approved devices. For this study, we have several different types of O.MG Cables and a Rubber Ducky. We also have a set of approved devices for each attack tool, after which the attack tools are disguised.

Table 4.7: Confusion matrix showing the classification of the novelty detection method applied to the Z80 dataset. For the results in this table, a maximum expected false positive rate was set to 5% and 40 devices were used for training.

	Predicted Nominal	Predicted Anomalous
USB Key	7	0
Rubber Ducky	0	1

For the first dataset, we have a set of 10 USB keys. These USB keys are of the same type from the same manufacturer, bought in the same batch. In addition to the USB keys, we have a Rubber Ducky attack tool disguised to look like one of the USB keys. This case study aims to find the Rubber Ducky attack tool against the set of approved USB keys.

Table 4.7 displays the confusion matrix of our method applied to the USB key/Rubber Ducky dataset. For this experiment, a maximum expected false positive rate of 0.5% was set, and three USB Keys were used for training. All devices in the test set were correctly identified.

Figure 4.2 displays the expected and usual distribution in the same format as Figure 4.1. The posterior distribution on the test statistic has a significantly different shape than in previous results, as only three devices were used during training. Furthermore, a USB connector only has seven electrical ports connected as opposed to the 16 electrical ports we connect to for the FT232RL and Z80 case studies. This causes the prior estimate for the covariance matrix to be less informed.

For the O.MG dataset, we have six different O.MG cables. Each O.MG cable is a different type of USB cable, as shown below:

- USB Type A to USB Type C
- USB Type C to Lightning
- USB Type A to Lightning
- USB TYPE C to USB Micro

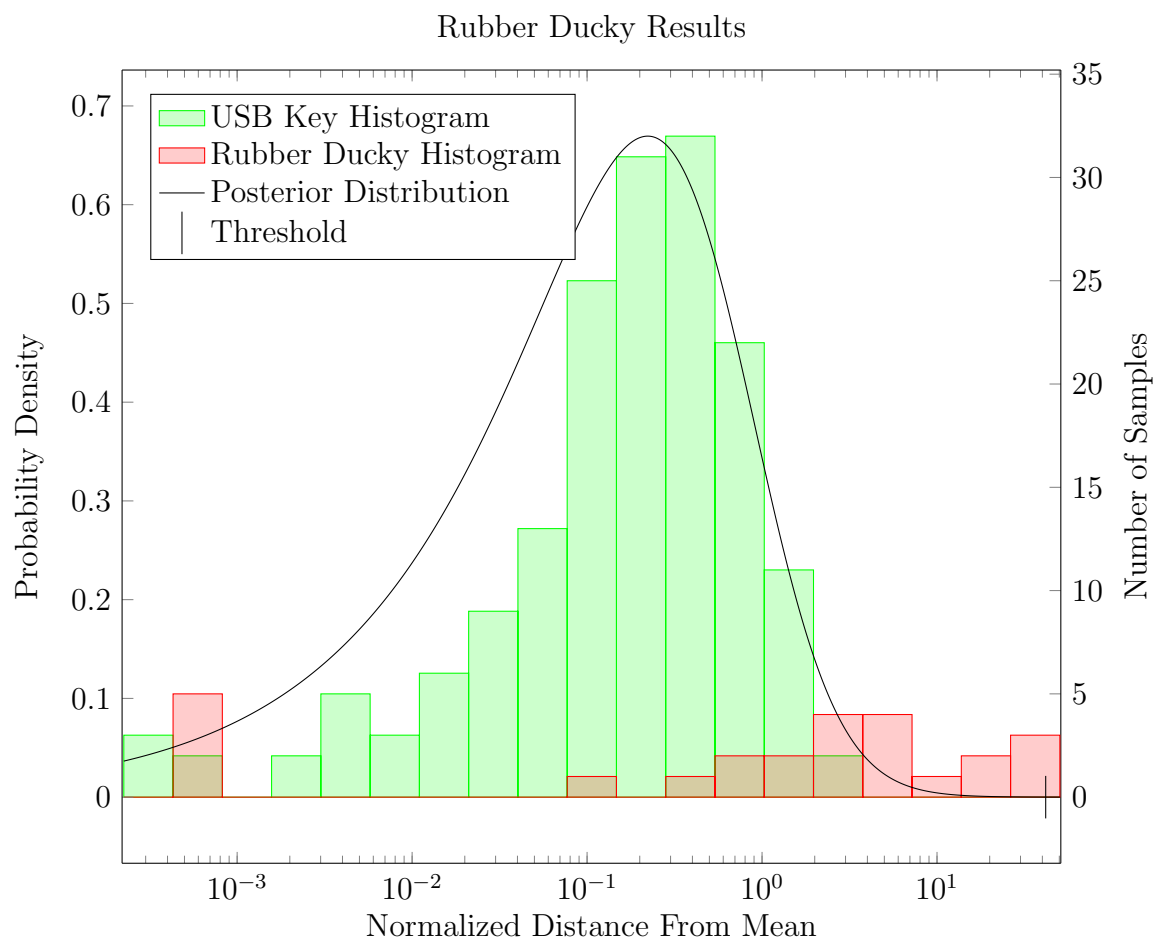


Figure 4.3: A visualization of the distribution of the Rubber Ducky experiment test set with respect to the test statistic.

- USB Type C to USB Type C

For each O.MG cable, we have ten visually identical USB cables which do not contain attack tools. For each cable class, we train the novelty detector on three randomly selected nominal cables to use the O.MG cable along with the other seven nominal cables as our test dataset. With the maximum expected false positive rate set to 10%, we 100% correct classification for four out of the six experiments. For the remaining two, all cables are predicted to be nominal. Note that a maximum expected false positive rate could be selected for all six experiments, resulting in 100% correct classification. However, the method’s conservative assumptions lead to conservative estimates to limit false positive rates.

Chapter 5

Future Work

There are many areas in which the method presented in this thesis would benefit from future work. We will discuss three key areas, including model validation, maximum expected false negative rates, and dimensionality reduction alternatives.

The method presented in this thesis uses expected false positive rates to set novelty detection thresholds in the absence of abnormal training samples. We suggest goodness-of-fit tests are used for model validation. Goodness-of-fit tests could provide model validation and automatically verify the assumptions on which the model is built during deployment.

The second area of improvement is to investigate if assumptions can be made that allow us to place a limit on the false negative rate as opposed to the false positive rate. In order to achieve this, we suggest domain knowledge on electromagnetic signal propagation be applied to determine the minimum change between an inauthentic and an authentic [DUT](#). This lower bound on the difference between authentic and inauthentic fingerprints might then be used to derive a maximum expected false negative rates similar to how the model currently finds a maximum expected false positive rate. An maximum expected false negative rate would allow the method to be used in fields in which safety concerns outweigh economic concerns.

We suggest that the dimensionality reduction method is further developed. While this thesis used simple dimensionality reduction methods to allow for statistical modelling, we suggest exploring how other dimensionality reduction methods, such as dynamic time warping, might be statistically modelled. Well-suited dimensionality reduction techniques could increase the separation between authentic and inauthentic measurements to allow for a more sensitive test. The challenge with more complex dimensionality reduction methods is that they can be challenging to model statistically.

Chapter 6

Conclusions

This thesis introduces an automatic novelty detection method for hardware validation with [SILGEO](#). The method is designed to operate under constraints faced by electronic manufacturers. Specifically, the novelty detection method is designed for training on a constrained set of verified devices. A maximum expected false positive rate can be estimated and adjusted without retraining despite the constrained training data. Furthermore, results are granular with respect to the electrical ports in the [DUT](#) and contain useful information for quality assurance technicians. The method incorporates Bayesian inference and posterior predictive p -values to achieve these outcomes.

To evaluate the novelty detection method introduced, case studies on two common integrated circuits and several types of attack tools are presented. In each case study, we obtained results in which false positive rates remained under the set maximum expected false positive rate while the over 90% of inauthentic or suspected inauthentic devices were correctly identified. An overview of novelty detection methods is given to motivate the choices made in the design of the method presented. Furthermore, an argument for a generalized version of a student's T-test is given for multivariate T distribution.

References

- [1] DO-160, Environmental Conditions and Test Procedures for Airborne Equipment. Radio Technical Commission for Aeronautics.
- [2] DSP :: DMSMS. <https://www.dsp.dla.mil/Programs/DMSMS/>.
- [3] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. Heliyon, 4(11):e00938, November 2018.
- [4] Charu C Aggarwal and Philip S Yu. Outlier Detection for High Dimensional Data.
- [5] Ignacio Alvarez, Jarad Niemi, and Matt Simpson. Bayesian inference for a covariance matrix, July 2016.
- [6] M. J. Bayarri and J. O. Berger. The Interplay of Bayesian and Frequentist Analysis. Statistical Science, 19(1):58–80, February 2004.
- [7] M. J Bayarri and J Morales. Bayesian measures of surprise for outlier detection. Journal of Statistical Planning and Inference, 111(1):3–22, February 2003.
- [8] Yoshua Bengio and Martin Monperrus. Non-Local Manifold Tangent Learning. In Advances in Neural Information Processing Systems, volume 17. MIT Press, 2004.
- [9] P. M. Bentler and Douglas G. Bonett. Significance tests and goodness of fit in the analysis of covariance structures. Psychological Bulletin, 88:588–606, 1980.
- [10] Johannes Berkhof, Iven van Mechelen, and Herbert Hoijtink. Posterior predictive checks: Principles and discussion. Computational Statistics, 15(3):337–354, September 2000.

- [11] Michael Betancourt. The Convergence of Markov Chain Monte Carlo Methods: From the Metropolis Method to Hamiltonian Monte Carlo. Annalen der Physik, 531(3):1700214, 2019.
- [12] Christopher M. Bishop. Pattern Recognition and Machine Learning. Information Science and Statistics. Springer, New York, 2006.
- [13] Brad M Boyerinas. Determining the Statistical Power of the Kolmogorov-Smirnov and Anderson-Darling Goodness-of-Fit Tests via Monte Carlo Simulation.
- [14] Leo Breiman. Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). Statistical Science, 16(3):199–231, August 2001.
- [15] Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan : A Probabilistic Programming Language. Journal of Statistical Software, 76(1), 2017.
- [16] RalphB D’Agostino. Goodness-of-Fit-Techniques. Routledge, October 2017.
- [17] Wayne W. Daniel. Applied Nonparametric Statistics. PWS-KENT Pub., 1990.
- [18] Richard O. Duda, Peter E. Hart, Peter Elliot Hart, and David G. Stork. Pattern Classification. Wiley, 2001.
- [19] Isabella Fornacon-Wood, Hitesh Mistry, Corinne Johnson-Hart, Corinne Faivre-Finn, James P. B. O’Connor, and Gareth J. Price. Understanding the Differences Between Bayesian and Frequentist Statistics. International Journal of Radiation Oncology, Biology, Physics, 112(5):1076–1082, April 2022.
- [20] Jerome H. Friedman. Stochastic gradient boosting. Computational Statistics & Data Analysis, 38(4):367–378, February 2002.
- [21] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. Bayesian Data Analysis Third edition (with errors fixed as of 15 February 2021). page 677.
- [22] Frank E. Grubbs. Procedures for Detecting Outlying Observations in Samples. Technometrics, 11(1):1–21, February 1969.
- [23] Ujjwal Guin, Ke Huang, Daniel DiMase, John M. Carulli, Mohammad Tehranipoor, and Yiorgos Makris. Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain. Proceedings of the IEEE, 102(8):1207–1228, August 2014.

- [24] Hak5. Hacking Tools & Media — Hak5 Official Site. <https://shop.hak5.org/>.
- [25] Basel Halak. CIST: A Threat Modelling Approach for Hardware Supply Chain Security. In Basel Halak, editor, Hardware Supply Chain Security: Threat Modelling, Emerging Attacks and Countermeasures, pages 3–65. Springer International Publishing, Cham, 2021.
- [26] Scharon Harding. Journalist plugs in unknown USB drive mailed to him—it exploded in his face. <https://arstechnica.com/gadgets/2023/03/journalist-plugs-in-unknown-usb-drive-mailed-to-him-it-exploded-in-his-face/>, March 2023.
- [27] Matthew D Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo.
- [28] Bryan T Horvath. NOT ALL PARTS ARE CREATED EQUAL. THE IMPACT OF COUNTERFEIT PARTS IN THE AIR FORCE SUPPLY CHAIN. page 30.
- [29] D.R. Hush and B.G. Horne. Progress in supervised neural networks. IEEE Signal Processing Magazine, 10(1):8–39, January 1993.
- [30] Syed Kamrul Islam and Mohammad Rafiqul Haider. Sensors and Low Power Signal Processing. Springer, New York, 2010th edition edition, December 2009.
- [31] Richard A. Johnson and Dean W. Wichern. Applied Multivariate Statistical Analysis. Pearson Prentice Hall, Upper Saddle River, N.J, 6th ed edition, 2007.
- [32] Firuz Kamalov and Ho Hon Leung. Outlier Detection in High Dimensional Data. Journal of Information & Knowledge Management, 19(01):2040013, March 2020.
- [33] Xiao-Li Meng. Posterior Predictive p -Values. The Annals of Statistics, 22(3):1142–1160, September 1994.
- [34] Carlos Moreno, Sebastian Fischmeister, and Philippe Vibien. A method and apparatus for detection of counterfeit parts, compromised or tampered components or devices, tampered systems such as local communication networks, and for secure identification of component, April 2021.
- [35] Kevin P Murphy. Conjugate Bayesian analysis of the Gaussian distribution. page 29.
- [36] Saralees Nadarajah and Samuel Kotz. Mathematical Properties of the Multivariate t Distribution. Acta Applicandae Mathematicae, 89(1-3):53–84, December 2005.

- [37] Radford M. Neal. MCMC Using Hamiltonian Dynamics. May 2011.
- [38] Karsten Nohl and Jakob Lell. BadUSB - On Accessories that Turn Evil, August 2014.
- [39] M. Pecht and S. Tiku. Bogus: Electronic manufacturing and consumers confront a rising tide of counterfeit electronics. IEEE Spectrum, 43(5):37–46, May 2006.
- [40] Marco A. F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. Signal Processing, 99:215–249, June 2014.
- [41] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation, February 2021.
- [42] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support Vector Method for Novelty Detection. In Advances in Neural Information Processing Systems, volume 12. MIT Press, 1999.
- [43] Committee on Armed Services United States Senate. Inquiry into Counterfeit Electronic Parts in the Department of Defense Supply Chain: Report of the Committee on Armed Services, United States Senate, May 21, 2012. CreateSpace Independent Publishing Platform, May 2012.
- [44] Mlnoru Slotani. Tolerance regions for a multivariate normal population. Annals of the Institute of Statistical Mathematics, 16(1):135–153, December 1964.
- [45] Mark (Mohammad) Tehranipoor, Ujjwal Guin, and Domenic Forte. Counterfeit Integrated Circuits. In Mark (Mohammad) Tehranipoor, Ujjwal Guin, and Domenic Forte, editors, Counterfeit Integrated Circuits: Detection and Avoidance, pages 15–36. Springer International Publishing, Cham, 2015.
- [46] S. Theodoridis and Konstantinos Koutroumbas. Pattern Recognition, Fourth Edition. November 2008.
- [47] Philippe Vibien. SilGeo: A Method for the Detection of Counterfeit, Compromised, or Tampered Electronic Devices. Master’s thesis, University of Waterloo, August 2022.
- [48] Pascal Vincent and Yoshua Bengio. Manifold Parzen Windows. In Advances in Neural Information Processing Systems, volume 15. MIT Press, 2002.
- [49] Ulrike von Luxburg and Bernhard Schoelkopf. Statistical Learning Theory: Models, Concepts, and Results, October 2008.

- [50] Thomas Wiecki, John Salvatier, Ricardo Vieira, Maxim Kochurov, Anand Patil, Brandon T. Willard, Michael Osthege, Bill Engels, Colin Carroll, Osvaldo A. Martin, Adrian Seyboldt, Austin Rochford, Luciano Paz, rpgoldman, Kyle Meyer, Peadar Coyle, Marco Edward Gorelli, Ravin Kumar, Junpeng Lao, Oriol Abril-Pla, Virgile Andreani, Taku Yoshioka, George Ho, Thomas Kluyver, Kyle Beauchamp, Alexandre Andorra, Demetri Pananos, Eelke Spaak, Benjamin Edwards, and Eric Ma. `Pymc-devs/pymc: V5.0.1`. Zenodo, December 2022.
- [51] Min-ge Xie and Kesar Singh. Confidence Distribution, the Frequentist Distribution Estimator of a Parameter: A Review: Confidence Distribution, the Frequentist Distribution Estimator. International Statistical Review, 81(1):3–39, April 2013.
- [52] Zhiyong Zhang. A Note on Wishart and Inverse Wishart Priors for Covariance Matrix. Journal of Behavioral Data Science, 1(2), 2021.
- [53] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining: The ASA Data Science Journal, 5(5):363–387, 2012.

APPENDICES

Appendix A

Hotelling's T Test Derivation

We aim to identify the smallest subset of the feature space that contains a certain percentage of the nominal measurement distribution. We consider a new measurement nominal if it falls within this subset. While this approach does not allow us to make any claims on the ability of our model to identify anomalous measurements, it does allow us to make claims on the false positive rate for nominal measurements.

We start by defining some terms.

- We define d as the dimension of our feature vector after dimensionality reduction.
- We define S to be the d dimensional feature space.
- We define the size $|W|$ of some subset of our feature space S to be the Lebesgue measure of W . The Lebesgue measure generalizes length, area, or volume for n dimensions.
- We denote the probability density function of a distribution $\mathcal{D}$ by $f_{\mathcal{D}}$.
- We denote the cumulative probability density function of a univariate distribution $\mathcal{D}$ by $F_{\mathcal{D}}$.
- We say a set W contains p of a distribution $\mathcal{D}$ when $\int_{\mathbf{x} \in W} f_{\mathcal{D}}(\mathbf{x}) d\mathbf{x} = p$.

We can rephrase the goal. Let $0 \leq p \leq 1$ represent our acceptable false positive rate. Let $\mathcal{D}$ be the distribution of nominal samples in our feature space S . Then we wish to find some subset $W \subset S$ of our feature space S such that has minimal size $|W|$ under the constraint that $\int_{\mathbf{x} \in W} f_{\mathcal{D}}(\mathbf{x}) d\mathbf{x} = p$.

In our case, the distribution of our nominal samples is a multivariate normal distribution $\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. To find a subset W of our feature space S with minimal size $|W|$

under the constraint that $\int_{\mathbf{x} \in W} f_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\mathbf{x}) d\mathbf{x} = p$ for some $0 \leq p \leq 1$ we use the Mahalanobis distance. The squared Mahalanobis distance with respect to the multivariate normal distribution is defined as

$$d_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \quad (\text{A.1})$$

and the probability density function of our nominal sample distribution is

$$f_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\mathbf{x}) = \left(\frac{1}{2\pi} \right)^{p/2} |\boldsymbol{\Sigma}|^{-1/2} e^{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})} \quad (\text{A.2})$$

Our first observation is that the probability density function of the multivariate normal distribution decreases monotonically as the squared Mahalanobis distance increases. Consider $\mathbf{x}, \mathbf{y} \in S$ with $d_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\mathbf{x}) \leq d_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\mathbf{y})$, then

$$f_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\mathbf{x}) = \left(\frac{1}{2\pi} \right)^{p/2} |\boldsymbol{\Sigma}|^{-1/2} e^{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})} \quad (\text{A.3})$$

$$= \left(\frac{1}{2\pi} \right)^{p/2} |\boldsymbol{\Sigma}|^{-1/2} e^{-\frac{1}{2}d_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\mathbf{x})} \quad (\text{A.4})$$

$$\leq \left(\frac{1}{2\pi} \right)^{p/2} |\boldsymbol{\Sigma}|^{-1/2} e^{-\frac{1}{2}d_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\mathbf{y})} \quad (\text{A.5})$$

$$= \left(\frac{1}{2\pi} \right)^{p/2} |\boldsymbol{\Sigma}|^{-1/2} e^{-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu})} \quad (\text{A.6})$$

$$= f_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\mathbf{y}) \quad (\text{A.7})$$

Since $f_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}$ decreases monotonically as $d_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}$ increases, the sets W_t defined by the squared Mahalanobis distance $W_t = \{\mathbf{x} \in S : d_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\mathbf{x}) < t\}$ have minimal size $|W_t|$ for the portions of the multivariate normal distribution they contain.

The last step is to find t for a given desired false positive rate p such that

$$\int_{\mathbf{x} \in W_t} f_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(\mathbf{x}) d\mathbf{x} = p \quad (\text{A.8})$$

We have that if X is a random variable in S that follows $\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $d_{\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}(X)$ follows a Chi-Squared distribution, specifically $\chi^2(d)$ [44]. So,

$$1 - p = F_{\chi^2(d)}(t) \tag{A.9}$$

$$t = F_{\chi^2(d)}^{-1}(1 - p) \tag{A.10}$$

In summary, we have a multivariate normal distribution representing nominal sample distribution. Given a desired false positive rate, we use the Mahalanobis distance to select a subset of our feature space. A new sample is labelled nominal if it falls within our selected subset and abnormal if it does not. Given our assumptions on the distribution of nominal samples, the expected false positive rate of the method matches the desired false positive rate.

Glossary

S_{00} the input port voltage reflection coefficient [8](#)

S_{01} the reverse voltage gain [8](#), [20](#)

S_{10} the forward voltage gain [8](#), [20](#)

S_{11} the output port voltage reflection coefficient [8](#)

SilGeo A system for creating hardware fingerprints based on impedance discontinuities and wave propagation characteristics. [i](#), [iii](#), [vi](#), [viii](#), [1](#), [2](#), [6–8](#), [16–19](#), [24](#), [48](#)