2017

Anomaly Detection in RFID Networks

Alaa Alkadi

ANOMALY DETECTION IN RFID NETWORKS

by

Alaa Alkadi

A thesis submitted to the
School of Computing
in partial fulfilment of the requirements of the degree of

Master of Science in Computer and Information Sciences

UNIVERSITY OF NORTH FLORIDA
SCHOOL OF COMPUTING

December, 2017

The thesis “Anomaly Detection in RFID Networks” submitted by Alaa Alkadi in partial fulfilment of the requirements for the degree of Master of Science in Computer and Information Sciences has been

Approved by the thesis committee — Date

_____ _____

Dr. Zornitza Prodanoff
Thesis Advisor and Committee Chairperson

_____ _____

Dr. Patrick Kreidl

_____ _____

Dr. Roger Eggen

Accepted for the school of Computing:

_____ _____

Dr. Sherif Elfayoumy
Director of the School

Accepted for the College of Computing, Engineering, and Construction:

_____ _____

Dr. Mark A. Tumeo
Dean of the College

Accepted for the University:

_____ _____

Dr. John Kantner
Dean of the Graduate School

ACKNOWLEDGEMENT

I would first like to thank my thesis advisor Dr. Zornitza Prodanoff of the School of Computing at the University of North Florida. The door to Dr. Prodanoff's office was always open whenever I ran into trouble or had a question about my research or writing. She consistently allowed this paper to be my own work, but steered it in the right direction upon any deviation.

I would also like to especially recognize and thank Dr. Patrick Kreidl of the School of Engineering at the University of North Florida for his absolute dedication and sacrifice in assisting with this research, finalizing findings and writing complex code. Dr. Kreidl did not hesitate to cut into his personal and private time to help the research stay on schedule and when it was crunch-time, he worked and directed the research over the phone even at hours beyond midnight. His understanding and flexibility went well beyond anything I've ever hoped for in a thesis project.

I would also like to acknowledge Dr. Roger Eggen of the School of Computing at the University of North Florida as the main facilitator, organizer and guidance throughout the thesis production process. Dr. Eggen made himself available always and was of great help from start to finish. I must also acknowledge and thank Mr. Jim Littleton of the School of Computing at the University of North Florida for his valuable time in assisting with finalizing, formatting and preparing this thesis document prior to publishing.

Finally, I must express my very profound gratitude to my employer Mr. Gianni Arcaini, Chairman and CEO of Duos Technologies Inc. in Jacksonville Florida for fully sponsoring and financially supporting my graduate education at the University of North Florida. Mr. Arcaini understands the importance of continuing education and has made every effort possible to accommodate the degree program at the work place.

I must also extend my gratitude to my superiors Mr. Charles Hoeppner, Director of Software Development at Duos Technologies Inc. and Mr. David Ponevac, Vice President and CTO at Duos Technologies Inc. Both Mr. David and Mr. Charles supported my degree program by providing flexible hours, remote days and time off when needed. This accomplishment would not have been possible without their unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. Thank you.

CONTENTS

FIGURES

TABLES

ABSTRACT

Available security standards for RFID networks (e.g. ISO/IEC 29167) are designed to secure individual tag-reader sessions and do not protect against active attacks that could also compromise the system as a whole (e.g. tag cloning or replay attacks). Proper traffic characterization models of the communication within an RFID network can lead to better understanding of operation under "normal" system state conditions and can consequently help identify security breaches not addressed by current standards. This study of RFID traffic characterization considers two piecewise-constant data smoothing techniques, namely Bayesian blocks and Knuth's algorithms, over time-tagged events and compares them in the context of rate-based anomaly detection.

This was accomplished using data from experimental RFID readings and comparing (1) the event counts versus time if using the smoothed curves versus empirical histograms of the raw data and (2) the threshold-dependent alert-rates based on inter-arrival times obtained if using the smoothed curves versus that of the raw data itself. Results indicate that both algorithms adequately model RFID traffic in which inter-event time statistics are stationary but that Bayesian blocks become superior for traffic in which such statistics experience abrupt changes.

Chapter 1

INTRODUCTION

1.1 Motivation

The development of Radio Frequency Identification (RFID) technology has changed the process of electronic identification in previously unimagined ways. Although RFID was initially developed over 50 years ago, its spread was inhibited until now by high cost and non-uniform industry standards. RFID technology is used in more products than ever today and is relied on to identify numerous items in our consumer intensive activities. While many individuals may not realize the direct impact of RFID technology on their lives, it is of immense value and is utilized by hospitals, airports, department stores, schools, warehouses and many others. With thousands of RFID transactions taking place on a daily basis an important question arises. Are RFID transactions immune to malicious attacks? Is the platform secure enough and does it provide a method to detect and reject anomalies? This study is motivated by the need to provide answers to these questions.

1.2 Background of the Study

The concept of detecting anomalies in network traffic is not entirely new as considerable research has already been done in that area. Typically, network anomaly detection is done through real-time classification of trends using various algorithms [Haselsteiner06].

These trends are analyzed and based on these algorithms a conclusion is drawn on whether an anomaly is present. A forecasting algorithm is often utilized to draw an expected baseline. Examples of such include the Holt-Winters method [Beach04]. Since not all anomalies are attacks or intrusions, a typical network traffic characterization method may rely on historical events and statistics to draw differences between detections. Behavior-based anomaly detection methods such as the maximum entropy estimation method have also been researched with relative success [Gu05]. This study, however, introduces new approaches to detecting such anomalies. These approaches are the Bayesian Blocks method and the Knuth Rule method. These algorithms will be briefly discussed.

1.3 Bayesian Blocks

The Bayesian blocks technique was developed to correct shortfalls in previous statistical methods. Originally developed for the analysis of photon arrival behavior, the Bayesian blocks method was developed to capture local variability which previously was near impossible in classical statistical methods. Initially proposed by [Scargle98], the aim of Bayesian Blocks modeling is to select, per some *fitness function*, the model that best fits patterns in the raw data. The Bayesian Blocks method is applied using an iterative procedure that finds approximate multiple change points: start with the whole observation interval, decide between single segment vs. two segments via an analytical method (i.e. via evaluating the closed form expression for the fitness function), pick the "winner" model and recursively apply the same procedure to its resulting subintervals [Alkadi16].

A halting condition is defined as no change points being detected by the analytical method in the "winner" subinterval of the data, where the threshold parameter ρ is used to put a lower limit on the subinterval length:

$$\rho \approx \frac{length\ of\ data\ interval}{desired\ time\ resolution}.$$

1.4 Knuth's Rule

Knuth's rule algorithm is similar to the Bayesian framework. Its aim is to determine an optimal number of bins (blocks) given the data, where bin lengths are equal in size (unlike variable length blocks in Bayesian Blocks) [Knuth13]. The algorithm works over multimodal data as it is designed to accept data of any distribution types [Alkadi16]. An advantage of Knuth's Rule over other methods is that it is not based on an optimization criterion that relies on the error between the density model and the actual density. The output of the Knuth's rule algorithm is a similar plot of a piecewise-constant function as the one produced by Bayesian Blocks. The significant difference between the two algorithms is that bin sizes are variable in Bayesian blocks and are fixed in Knuth's algorithm, hence the condition that bin lengths need to be of fixed size for the output model produced in an iteration. The execution starts by computing a probability mass for each bin in the raw data histogram through the integration of the height of the bin (block) over its width. Hence, the input is empirical, and no prior knowledge of the sampled probability density function is used. Therefore, the algorithm is designed to accept data of any distribution type [Knuth13]. In each iteration, the goal is to produce the posterior

probability for the model with a given number of bins. The optimal model over all iterations is selected by maximizing the logarithm of the posterior probability.

1.5 Problem Statement

Most of the proposed models for RFID networks use channel modeling techniques with the goal of quantifying network performance in terms of lower layers' channel properties: antenna diversity gain, signal-to-noise ratio, bit error rate, path loss and others [Arnitz10, Chen11, Choudhury14, Li11, Malison08, Smietanka12]. Unfortunately, such metrics are not very useful for the purpose of detecting patterns and anomalies as user-level traffic behavior cannot be assessed through their use. One example of more useful quantification would be to consider the application-layer traffic patterns in terms of the statistical frame arrival process at the RFID reader. Protecting the air interface between ultra-high frequency RFID tags and a reader has been adequately solved by standardization efforts through the development of latest lightweight cryptography standards [Alkadi16]. The International Standards Organization (ISO) / International Electrotechnical Commission (IEC) 29167 defines several crypto suites: AES 128, PRESENT80, ECC-DH, Grain 128-A, AES-OFB, XOR, ECDSA-ECDH, cryptoGPS, RAMON (Part 10 through Part 19) [ISO11].

These available crypto suite choices include a range of widely adopted techniques including block and stream ciphers with 128-bit key length, elliptic curve cryptography based on Diffie-Hellman key agreement method, low-cost public key cryptography, and

Rabin-Montgomery cryptography. In addition, NIST SP 800-57 recommendations on the security strength of lightweight crypto suites require that after 2030 symmetric cipher encryptions with key lengths shorter than 128-bits be disallowed. Yet, commercial implementations of RFID systems compliant with various ISO/IEC air-interface standards do not often offer security through ISO/IEC 29167 [Alkadi16].

In addition, the ISO/IEC 29167 standard does not address protection for High Frequency (HF) RFID networks using standards such as ISO/IEC 14443 and ISO/IEC 15693, or international standards, e.g. FeliCa. The main reason for this void is the fact that HF RFID networks are short range since power is transferred to and from the tag by means of the inductive coupling of reactive near-field energy. As inductive coupling takes place in the antenna's near-field region, no power is being radiated outside the near-field. Ultra-High Frequency (UHF) RFID tags have to be better secured because of their longer read range as power transfer to and from the tag is realized through capacitive coupling and involves the reader emitting a propagating electromagnetic wave. Nevertheless, known Near-Field Communication (NFC) attacks exist and have been documented. Some of them include eavesdropping, data corruption, man-in-the-middle attack, and data insertion and modification [Alkadi16].

1.6 Objective of the Study

The aim of this study is to evaluate the performance of Bayesian Blocks and Knuth's Rule algorithms for the purposes of modeling HF RFID traffic and their use in anomaly

detection, where an anomaly is a sudden change of the flow of the network traffic. This would allow traffic characterization of the RFID transaction to take place. It is believed that this characterization would allow for the better understanding of systems under "normal" conditions, and consequentially help in identifying security breaches not addressed by current standards.

Chapter 2

RFID BACKGROUND

2.1 What are RFID Devices?

RFID devices are wireless communication devices used in RFID networks for tagging and identification. Typically, RFID devices can be categorized into readers and tags.

2.2 RFID Tags

RFID tags are devices capable of holding identification information in their memory and transmitting this information over a distance in response to a query from a reading apparatus. RFID tags are usually made up of integrated circuits coupled to an antenna that is typically printed, etched, stamped or vapor-deposited on a paper substrate or Polyethylene Terephthalate (PET) surface. The integrated circuit (IC) and antenna combo, also referred to as an inlay, is placed on a printed label and inserted into a more durable structure. RFID IC's perform the functions of storing and processing information as well as modulating and demodulating radio frequency signals [ISO11].

By design, RFID tags can be passive, active or battery assisted passive (BAP). Passive tags are smaller and do not require a battery to supply their energy needs. However passive tags derive their energy from the radio energy transmitted by the reader. Active

tags usually require a battery to supply their energy needs. Since they have an active source of energy, they may be programmed to transmit their ID signal periodically [ISO11]. Tags may be single channel (read only) or dual channel accessible (read and write). The manufacturer frequently assigns tags a serial number. It is also noteworthy to recognize that tags may contain more than a serial number and may include stock numbers, production dates or other specific information. The serial number, also known as an electronic product code (EPC), is written to the tag by an RFID printer and is usually a 96-bit string of data [ISO11]. The first eight bits represent the header section which identifies the version of the protocol. The next 28 bits represent the organization that manages the data for the tag. The next 24 bits indicate its object class which identifies the type of product, with the last 36 bits representing a serial number that is unique. The last two fields are usually determined by the manufacturer that issued the tag [ISO11].

2.3 RFID Readers

A radio frequency identification reader (RFID reader), also known as an interrogator, is a device used to interface and communicate with RFID tags. This is made possible by radio waves that facilitate the transfer of data between tags and the reader through an antenna that captures tag information. Readers perform the tasks of continuous inventorying, filtering (searching for compatible tags) and writing and encoding tags. Various kinds of RFID readers are available and can be classified by the type of tags they can interact with [ISO11].

Passive Reader Active Tags (PRAT) systems are capable of receiving signals from active tags, and they are operable within a radial distance of up to 600m from the tag. This allows for flexibility in an application such as asset protection [ISO11].

Active Reader Passive Tag (ARPT) systems utilize an active reader component that sends interrogative signals as well as responses from a passive tag. Active Reader Active Tag (ARAT) systems include an active tag which can be awoken by signals sent from the active reader. These are also fixed readers that are set up within a tightly controlled zone [ISO11].

In order to communicate with tags, readers require the use of an antenna. RFID antennas serve as a link between readers and tags by effectively converting electric current into radiated waves. The choice of reader and tag antenna is influenced by the factors such as the application and environment as there are different types of antennas [ISO11].

Like reader systems, there are various types of RFID antennas which are best suited for various applications and environments. The most popular types of antenna systems are the linear and circular polarized antennas. Linear antennas are best deployed in instances where long ranges are desirable, and high powers are applicable. This enables signals to possess high penetrative powers. They are however sensitive to tag orientation and may exhibit difficulty in reading tag information. Conversely, circular polarized antennas have less penetrative power as well as range but are less susceptible to tag orientation. The operational range usually influences the choice of antennas. Operational instances less

than 30cm utilize magnetic coupling as readability is usually uninfluenced by the introduction of potential dielectrics such as metal, wood, and water. In remote applications, communication between reader and tag is affected by the presence of dielectrics. Therefore, electromagnetic couplings are not recommended [ISO11].

2.4 Application of NFC in Today's World

Today, most daily activities inevitably involve the use of digital communications. With the increase in integration of Near Field Communication (NFC) in mobile devices, various opportunities abound for the application of this technology. Evolving from Radio Frequency Identification (RFID) technology, NFC is developed from short-range radio communication technology which brings two NFC compatible devices together in less than four centimeters [Hongwei13]. Developed by Sony and Philips in 2012, NFC is rapidly becoming popular and will soon be in common devices [Hongwei13]. NFC development is vigorously being pursued by big corporations such as Apple, Google, and Blackberry. With these companies announcing plans to fully deploy NFC in e-commerce and health care organizations [Hongwei13]. NFC is already being applied in various instances some of which are further discussed in this paper.

2.4.1 Financial Transactions

One of the most prominent uses of NFC today is the facilitation of financial transactions as holding significant amounts of cash could be bothersome as well as pose significant

risks. This has led to the development of secure, convenient, peer to peer as well as peer to business transactional platforms based entirely on contactless NFC technology. A recent study conducted by PayPal in Canada suggests that as much as 34% of people now prefer electronic payments over cash payments [Hongwei13].

2.4.2 Transportation

NFC technology is also applicable to the transportation sector. Transportation companies, especially in railway and train services, have begun to use NFC technology to ease user stress and increase comfort. Transportation industries in the United States of America (US) and Germany announced in 2011 that plans were underway to make available NFC-based ticketing systems and information systems on board trains [Hongwei13].

2.4.3 Entertainment and Hospitality

NFC technology is also extensively deployed in the hospitality sector where it is used in various instances such as security key cards and payment systems. Users are also able to "check in" their locations on social media using NFC technology [Hongwei13].

2.5 RFID Standards

Several organizations such as the International Standards Organization (ISO), Auto-ID Center and EPCglobal have developed various RFID standards for standardization and easy implementation across the board. RFID standards cover air interface protocols,

product compatibility, applications and data content. EPCglobal developed a tag classification system that is recognized by the World Trade Organization (WTO) and the ISO. The classification scheme is presented in Figure 1. [ISO11] is an air interface protocol describing tag and reader specifications. Figure 1 also highlights the major classifications of the ISO 18000 protocol. Anti-collision protocols such Q and the adaptive Q algorithm have been developed for use in UHF Class 1 Gen 2 tags. Without these algorithms, it would be impossible to have two tags communicate with a single reader at the same instant.

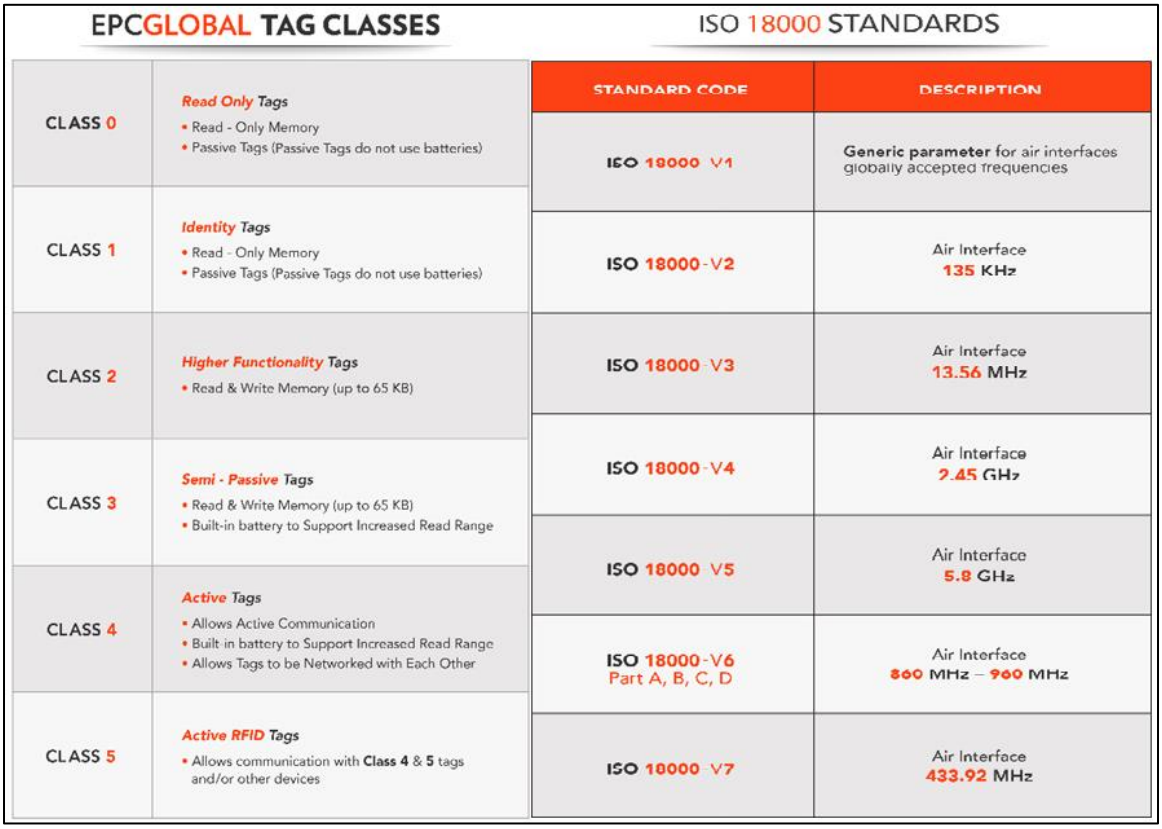

Figure 1: ISO 18000 Standards and EPCglobal Tag Classes [Farooq12]

2.6 Performance of RFID Systems

[Wang09] conducted a study investigating the performance of UHF passive RFID systems. The test was carried out with the aim of analyzing and establishing the loss of performance under varied conditions of free space, materials, and Line of Sight (LOS) environments. Performance analysis in open space was carried out using the expression,

$$P_0 = Q\left(\sqrt{2\frac{E_b}{N_o}}\right) \text{ with } Q(y) = (2\pi)^{-1/2} \int_y^{\infty} e^{-\frac{x^2}{2}} dx$$

With x and y denoting Cartesian co-ordinates. Figure 2 shows a plot of the degradation of power with distance using some assumed values.

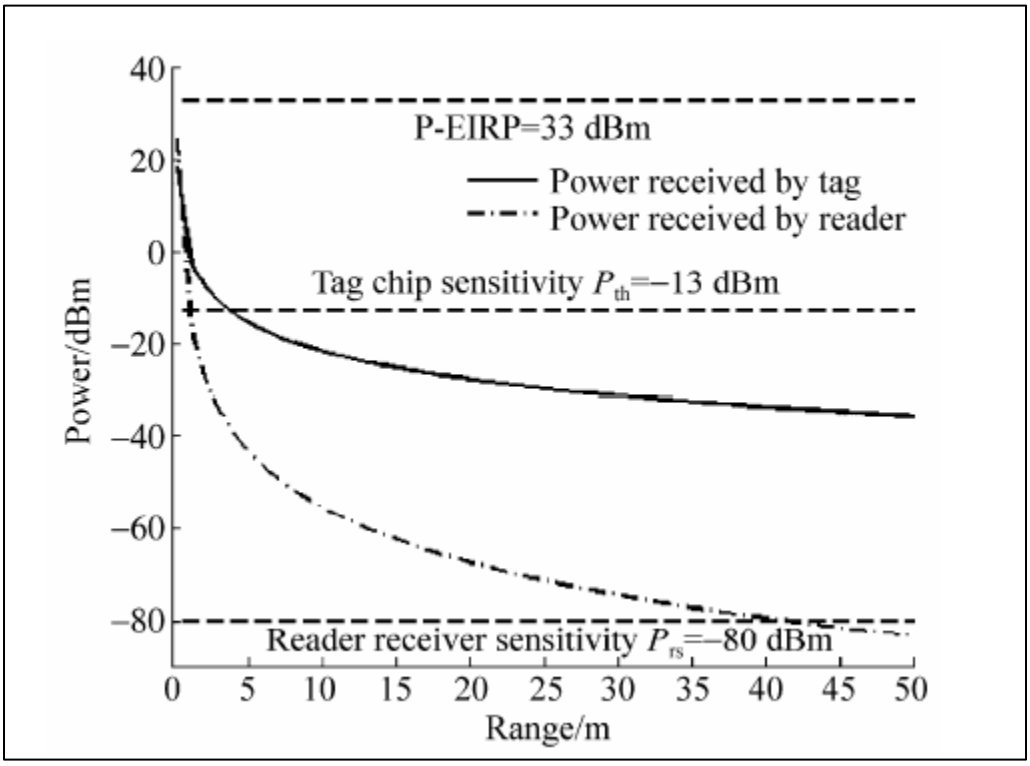

Figure 2: Degradation of Reader/Tag Power with Radial Distance [Wang09]

When [Wang09] tested for performance near various materials, the power gain L_{gain} was defined as the gain/loss of tag antenna due to materials. This was evaluated as,

$$A_0 = \frac{\lambda^2 G_{tag}}{4\pi L_{gain}}$$

Where A_0 is the effective area of the antenna, and G_{tag} is the antenna gain. The result presented in Figure 3 and Figure 4 shows the degradation of information transfer rate. Table 1 shows L_{gain} for different materials.

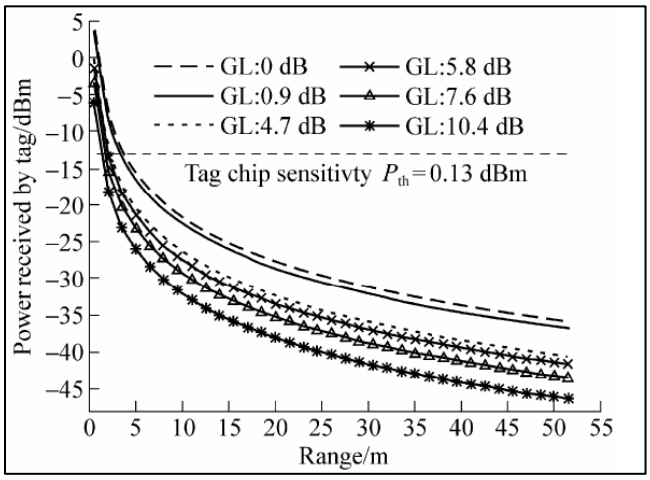

Figure 3: Degradation of Reader/Tag Power for Various Materials [Wang09]

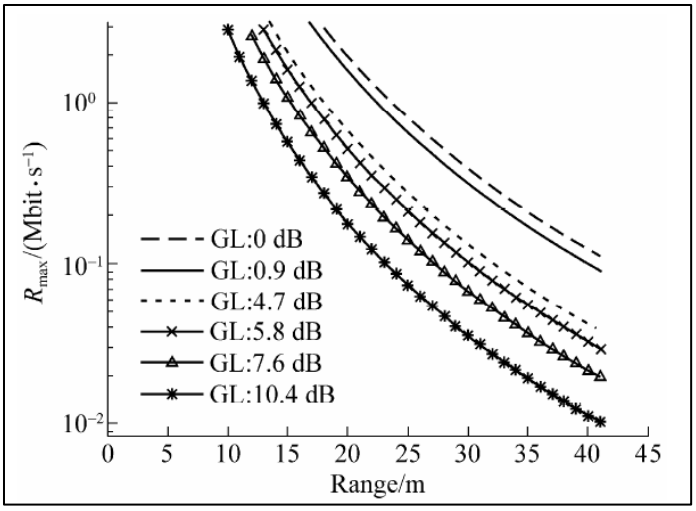

Figure 4: Degradation of Reader/Tag Information Transfer Rate for Different Materials [Wang09]

Materials	Cardboard sheet	Pine plywood	Deionized water	Ethylene glycol	Aluminum slab
L_{gain} /dB	0.9	4.7	5.8	7.6	10.4
L, W, H/(cm)	31, 65, 4	30, 57, 3	22, 10, 10	22, 10, 10	46, 58

Table 1: Lgain for Different Materials [Wang09]

2.7 Importance of Anomaly Detection in RFID Networks

The detection of anomalies in RFID networks is of enormous importance due to the financial, social, as well as, security implications of compromised RFID transactions. The widespread implementation of RFID means that very grave consequences could arise if anomaly detection is not adequately tackled. As explained earlier, current RFID standards such as the ISO 29167 protect and encrypt the individual transactions from successful attacks such as Man in the Middle (MITM). It is, however, necessary to secure and detect

network attacks as early detection of attacks could be advantageous. [Faruqui08] states that although most tags are capable of some limited symmetric key cryptography, there are still continuing threats in several areas. The use of RFID in healthcare presents an excellent example of situations where there could be profound effects of anomaly detection.

[Dunnebeil11] proposed an electronic health card system (eHC) which incorporates RFID with the aim of acquiring patient data from the card even in an unconscious state. This system with its proposed benefits could conversely have adverse effects if measures such as anomaly detection are not adequately taken. Possible adverse consequences in this instance include health insurance fraud, identity theft, and violation of medical privacy or in the worst case, death. Other applications of RFID such as supply chain management, aviation, and hospitality also pose a significant risk in the area of corporate espionage, loss of inventory, security, and privacy [Angeles05]. However, only a few RFID anomaly and intrusion detection studies have been published to date [Farooq12, Gaitan12, Yang11].

2.8 Bayesian Statistics

The world of statistics is commonly classified into two paradigms which are the frequentist and the Bayesian schools. The Bayesian paradigm believes that probability is a rational, conditional measure of uncertainty which closely mirrors the actual state of affairs of an event. Named after Thomas Bayes, Bayesian theory in contrast to the

frequentist school does not assume that probabilities are outcomes of long repeated trial but rather information which can be updated in the light of new evidence. Bayesian statistics utilizes non-negative degrees of belief also known as priors and uses these priors to predict posterior beliefs or probabilities.

Mathematically, Bayesian probability for discrete events can be represented as,

$$P(\theta|y) = P(y|\theta)\frac{P(\theta)}{\mathrm{P}(y)}$$

With $P(\theta|y)$ read as the probability that an event θ occurs given the event y has occurred and is known as the posterior probability while $P(y|\theta)$ represents the likelihood of an event. $P(\theta)$ and $\mathrm{P}(y)$ represent the probability of prior events. The expression above can also be applied to continuous events and distributions with minimal differences [David09].

2.9 Applications of Bayesian Statistics

Typically, Bayesian methods are used in three instances [David09]. These instances are described in the following three sections.

2.9.1 Lack of Accurate Prior Data

This is explained as situations where the only viable method is to include quantitative prior judgment as a result of certain inadequacies of a model or data. Instances of this include policy determination given incomplete data from a range of sources. Examples

include the use of Bayesian methods by the Food and Drug Administration (FDA) in the drug trials [David09].

2.9.2 Mid- Sized Problems with Multiple Sources of Evidence

In moderately sized problems, featuring various sources of evidence (data) such that hierarchical models can be constructed by assuming a prior shared distribution whose parameters can be estimated from the data, Bayesian methods may be employed. Instances of this include multi-center studies, meta-analysis, diseases mapping and accident mapping. Although results may be similar to results obtained using classical techniques, interpretations are different [David09].

2.9.3 Joint Huge Probability Models

Bayesian techniques are also applied to the construction of the joint probability model. This model is constructed sometimes using thousands of observations and parameters where the only feasible way of predicting and making inferences of unknown quantities is by Bayesian techniques. Popular applications include signal analysis, image processing, spam filtering, and gene expression data [David09].

Another application of Bayesian methods in a not so strict sense involves the construction of joint prior distributions where parameters of inputs are uncertain in a deterministic prediction model. By placing a joint distribution on these parameters, a predictive

probability distribution can be formulated using techniques such as Monte Carlo methods [David09]. This is commonly employed in risk analysis, economic health modeling and climate projections and is commonly referred to as probabilistic sensitivity analysis [David09].

2.9.4 Bayesian Blocks

Bayesian blocks is a statistical analysis technique developed to detect local variability as well as function as a characterization algorithm. Motivated by limitations in existing characterization techniques, Bayesian blocks seeks a novel approach to the statistical problem of binning and clustering such that the statistic is considered constant within a sub-interval which in related literature are referred to as "Blocks." Initially developed by [Scargle98] to detect and characterize photon behavior in gamma bursts, the objective of the analysis was to find the optimal binning that identified blocks such that the photon arrival rate is consistent with being constant. According to [Scargle98], classical characterization and binning fallacies such as equal bin lengths and minimum size often erase vital observational information, which is often found in gamma bursts which are smoothed over in classical methods.

Bayesian blocks as proposed by [Scargle98] are suited to analyze:

- Time-Tagged Event (TTE)
- Count of events in bins.

- The measurement of a quasi-continuous observables at a sequence of events in time.

For the first two cases, the signal of interest is the event rate which is proportional to the probability distribution regulating events and is relevant to this study. Since the Poisson distribution closely models the probabilities of photon arrival or non-arrival, it forms the core building block for the resulting formulation. Mathematically, the probability of an event P occurring according to the Poisson distribution is given as,

$$P_k = \frac{(\lambda \delta t)^k \, e^{-\lambda \delta t}}{k!}$$

Consider a time series event of length T, divided into M sub-interval of δt = T/ M. The sub-interval δt is selected such as any block is multiples of δt. From the above expression, the probability of an event not occurring (i.e. no photon arrival) reduces to:

$$P_0 = \, e^{\,-\,\lambda \delta t}$$

Therefore, the probability that an event occurs $(P \leq 1)$ equals:

$$1 - \, P_0 = 1 - \, e^{-\lambda \delta t}$$

Hence the likelihood function (i.e. the probability of obtaining the data) that N out of M blocks have photons in them is:

$$P(D|\, \theta, M) = \prod_j p_0^{M-N} \, (1 - \, p_1)^{M-N}$$

It is, however, desirable to obtain the global likelihood over the time series i.e. $P(D|\, M)$. By assuming p_1 as a parameter that is uniformly distributed between 0 and 1, the global likelihood $P(D|\, M)$ is then evaluated as:

$$P(D|\, M) = \int d\theta \; P(D|\theta, M) P(\theta|M)$$

$$P(D|\,M) = \int_0^1 dp_1\, p_0^{M-N}\,(1 - p_1)^{M-N}$$

The above expression can be re-written as:

$$P(D|\,M) = B(N+1, M-N+1)$$

Which is the beta function and can also be written as:

$$\frac{\Gamma(N+1)\Gamma(M-N+1)}{\Gamma(M+2)}$$

Therefore, from a Bayesian perspective, the global posterior probability $P(M|\,D)$ can be determined as:

$$P(M|\,D) = P(D|\,M)\;P(M)$$

Where $P(M)$ is the prior probability of the model [Scargle98].

To find blocks, the Bayesian blocks method compares two models (fitness functions) as a function of their global likelihoods. The two models are described as,

- Model 1: The dataset is contained in one block
- Model 2**:** The dataset is segmented into two blocks at some change point that maximizes the global likelihood of model 2.

The models are compared using the ratio,

$$\frac{P(M_2|\,D)}{P(M_1|\,D)} = o_{21}$$

The ratio o_{21} is also called the Bayes factor and depending on the ratio, the segmented model (model 2) or the un-segmented model is selected.

2.9.5 Scargle's Algorithm

As earlier explained, the objective of the Bayesian blocks technique is to find the optimal block segmentation of the data into a piecewise constant representation, with the case for two segments demonstrated. However, when the number of segments becomes more than N = 2, the earlier stated method become computationally burdensome and is rather impractical. It is, therefore, desirable to develop a simple iterative approach to determine the point of segmentation. This simple iterative technique is known as Scargle's Algorithm named after [Scargle98].

The Algorithm begins with the whole observation. The Bayesian blocks technique is then applied to the interval to determine whether it should be segmented or not. If segmentation is favored, the procedure is then re-applied to all the created intervals. If the computed ratio favors segmentation, the resulting sub-intervals are then subjected to the Bayesian blocks technique until the computed ratio over any interval does not favor segmentation. This might seem as a logical sequence of events and conditions however, when it is considered that the computed Bayes ratio commonly has a value slightly greater than 1, it becomes desirable to propose an alternative condition. A solution to this by [Scargle98] is to impose a prior ratio where insignificant segmenting is discouraged. This prior ratio is computed as,

$$p \approx \frac{lenght\ of\ data\ interval}{desired\ time\ resolution}$$

2.9.6 Application to Astronomy

For demonstrative purposes, this section shows the application of Bayesian blocks in the analysis of Burst and Transient Source Experiment (BATSE) (Trigger 551) gamma rays. This being an example of TTE it also serves as a good example due to its modulating structure. By applying Scargle's algorithm to four gamma ray TTEs, using a MATLAB program which is provided in the appendix section, the following plots were obtained. The plots show the characterization as performed by Scargle's algorithm on the left while the characterization using 32 evenly spaced bins is shown on the right. Figure 5 shows the resultant plot when all the 4 TTEs are superimposed onto 1 time series, while Figure 6 shows the individual TTEs [Scargle98].

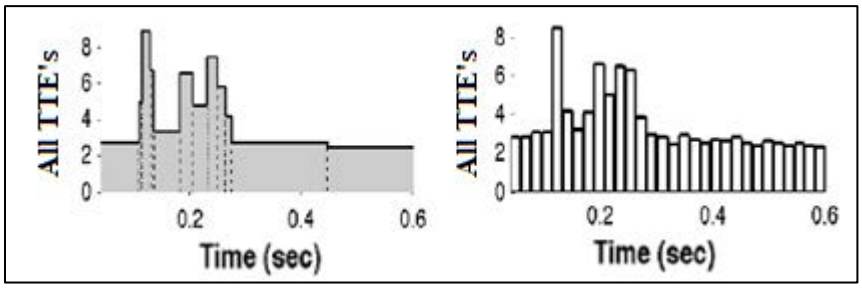

Figure 5: The Characterization of the Superimposed TTEs [Scargle98]

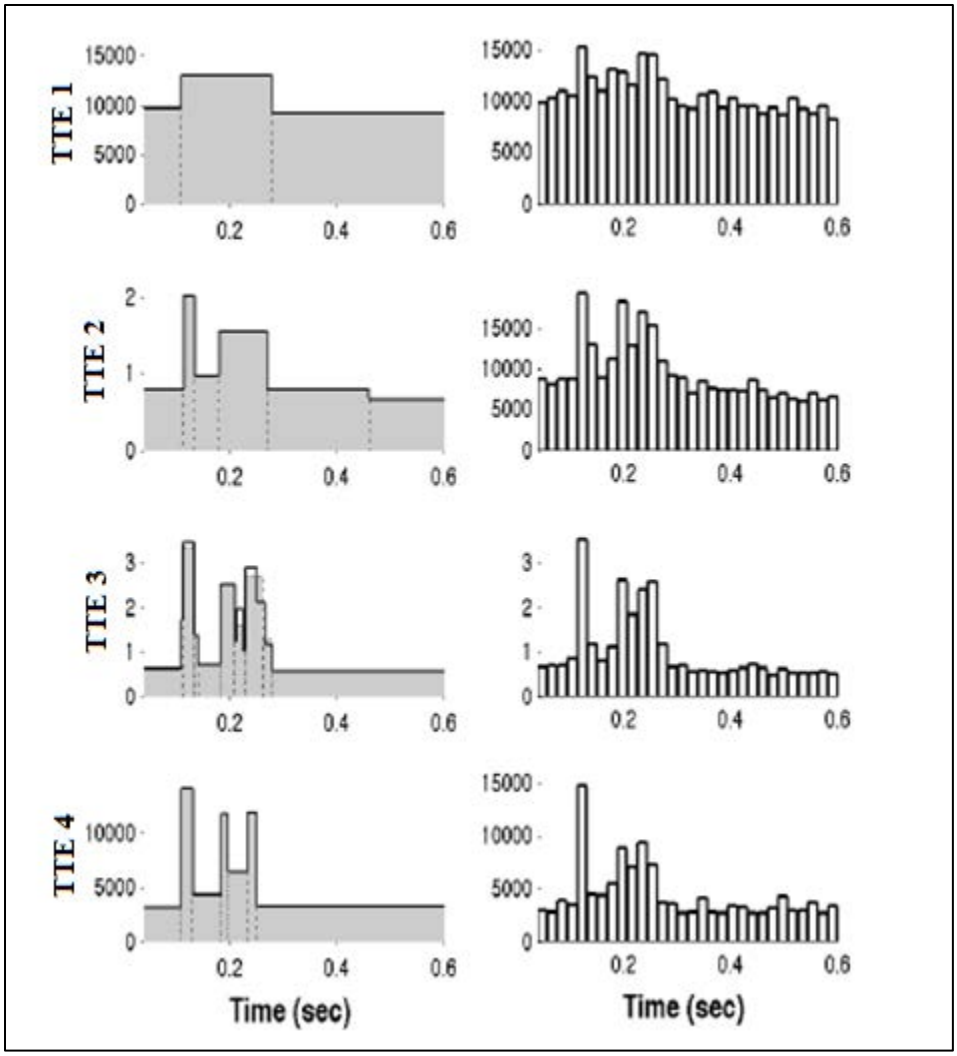

Figure 6: The Characterization of the Four Individual TTEs [Scargle98]

2.9.7 Knuth's Algorithm

The output of the Knuth's rule algorithm is a similar plot of a piecewise-constant function as the one produced by Bayesian Blocks. It is similarly based on the Bayesian statistics framework. The goal is to find the optimal number of bins (blocks) given the total observation interval length and the raw data time series. A condition is imposed that bin lengths need to be of fixed size for the output model produced in an iteration. The execution starts by computing a probability mass for each bin in the raw data histogram through the integration of the height of the bin (block) over its width. Hence, the input is empirical, and no prior knowledge of the sampled probability density function is used. This is why the algorithm is designed to accept data of any distribution type. In each iteration, the goal is to produce the posterior probability for the model with a given number of bins. The optimal model over all iterations is selected by maximizing the logarithm of the posterior. Further details on the derivations for the piecewise constant model can be found in [Knuth13].

2.9.8 Related Works

[Alkadi16] conducted an initial study in which piecewise linear models were used to characterize RFID traffic data using both Scargle's and Knuth's algorithm. The dataset consisted of 650 RFID tag transactions carried over the ISO/IEC 14443 half duplex block transmission protocol. Figure 7 shows the output plot of the characterization algorithms. A visual inspection of the plots reveals a good fit by both models suggesting that both

Bayesian blocks and Knuth's algorithm can be potentially used in the design of network intrusion detection systems.

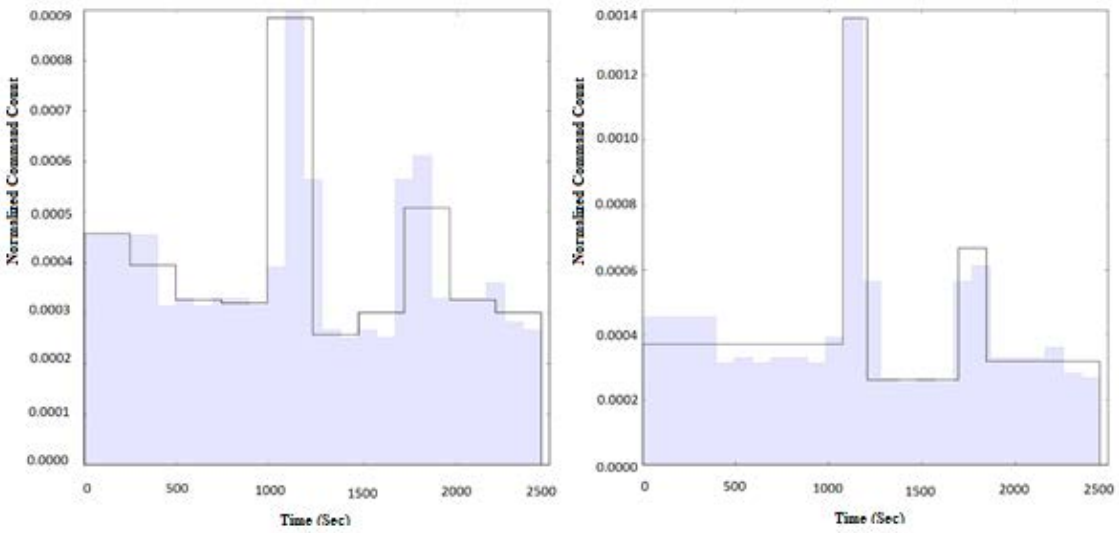

Figure 7: Normalized RFID Command Count

Figure 7 shows Raw Binned Data (Purple) and Horizontal Piecewise Models (Black): Knuth's Rule (Left), Bayesian Blocks (Right). Note: Vertical Outlines are Also Shown for Better Visualization.

Chapter 3

THE EXPERIMENT

3.1 Method of Experimentation

In the preceding chapters, Bayesian blocks, Scargle's and Knuth's Algorithm have already been described as well as their applicability to the characterization of TTE's. In this section, the Scargle and Knuth algorithms are used in the characterization of generated RFID tag data. For experimentation, several tags were read using a single reader with each session lasting long enough to transmit 200, 400, 500 and 1000 tag responses. Each response message was carried by a single frame as defined in the ISO/IEC 14443 protocol - a half-duplex block transmission protocol.

The experimentation was performed using a single Blackberry Priv RFID/NFC reader, running Android 5.1.1 and several NFC tags (IC chips model NTAG213, produced by NXP). Tag user area capacity was 144 bytes.

The test bed of our experiment included an application to scan incoming NFC tags, a simple database on the Android-based device itself storing every piece of information collected about each tag when scanned and a number of standard NFC tags. After a dataset (i.e. experimental sample) is collected which includes scans and inter-arrival

times added to the internal database, the database is then exported as a CSV file to a Windows based PC.

The CSV file's data are then parsed, normalized and plotted by our Bayesian Blocks and Knuth's Rule graph-generating program written in Python. This Python program can be found in Appendix A. The program also draws the raw data as the background of each plot. This step is useful for visually identifying the difference between the constant bins' size of Knuth's Rule and the variable bin sizes of the Bayesian Blocks algorithm. The program is also capable of keeping other metrics during execution, such as the set of change points of each Bayesian Blocks graph as well as the constant bin size rendered by Knuth's Rule graphs.

Chapter 4

THE OUTCOME

4.1 Results and Discussion

The input to the two piecewise-constant models used in this study was in the form of time series extracted from the experimentation log file, the produced CSV file. A statistical description of the samples that we generated from actual experimentation is shown in Table 2. Those five samples consisted of 200, 400, 500, and 1000 frame arrivals, respectively.

Statistic / Counts	**200**	**400**	**500**	**1000**
Mean	3.66	2.23	7.30	5.28
Standard Error	0.04	0.05	0.07	0.09
Median	3.94	2.41	7.67	5.44
Mode	3.94	3.85	6.51	2.14
Standard Deviation	0.54	1.01	1.64	2.89
Sample Variance	0.30	1.02	2.67	8.37
Kurtosis	0.22	-1.11	-0.39	-1.31
Skewness	-1.49	0.31	0.01	0.07
Range	1.36	2.8	5.56	8.83
Minimum	2.58	1.05	4.31	1.05
Maximum	3.94	3.85	9.87	9.88
Count	198	398	498	998

Table 2: Descriptive Statistics for Study Samples

The scans within each dataset were collected and gathered manually. The time between each scan was loosely kept constant during every dataset entirely. For the 200 and 400 sample sets, the time between scans was approximately 3 seconds, 10 seconds for the 500 and 1000 sample sets. Figures 8 through 11 present the output from the execution of Knuth's Rule and Bayesian Blocks algorithms (Knuth's on top, Bayesian's on the bottom) on the sets. Shown in purple, is the raw data format with total command counts recorded in bins. Each bin is a time interval of fixed length (25 seconds). The models are depicted in black, where vertical outlines are added to the piecewise-constant plots for better visualization.

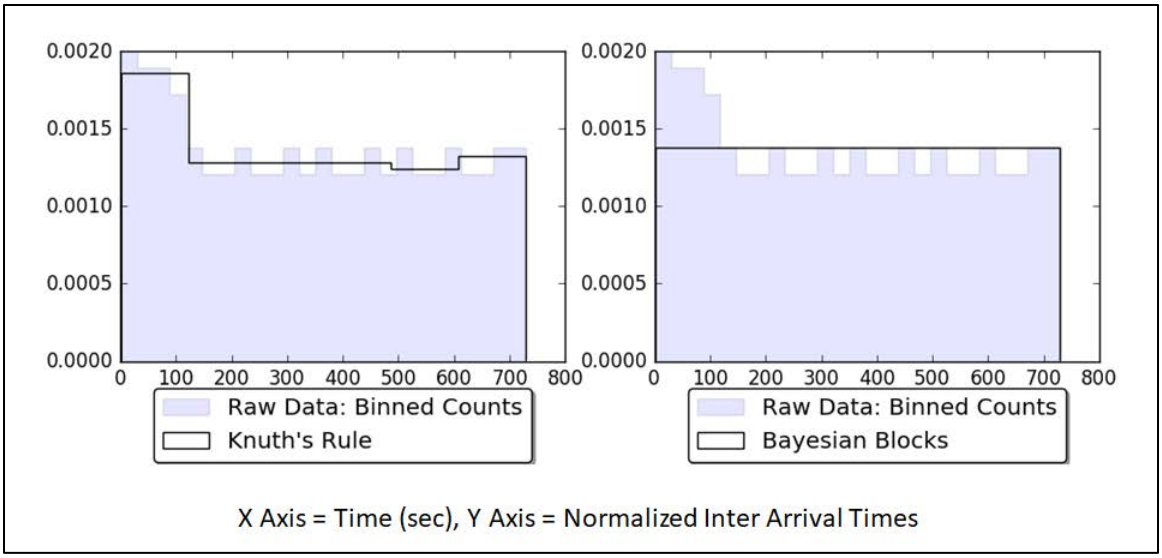

Figure 8: Output of Bayesian Blocks and Knuth's Algorithm for the 200 Sample Set

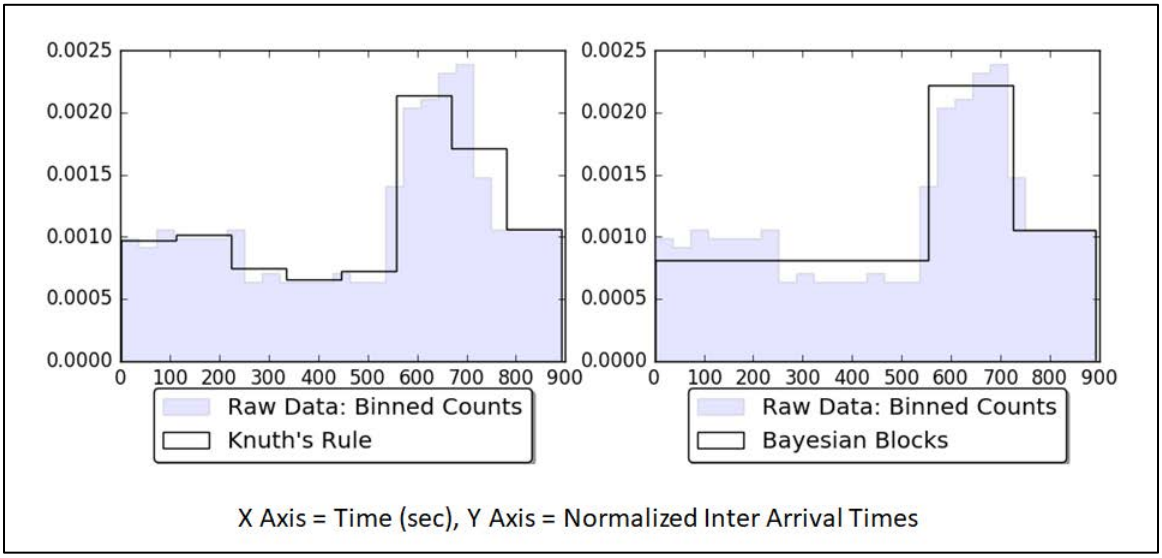

Figure 9: Output of Bayesian Blocks and Knuth's Algorithm for the 400 Sample Set

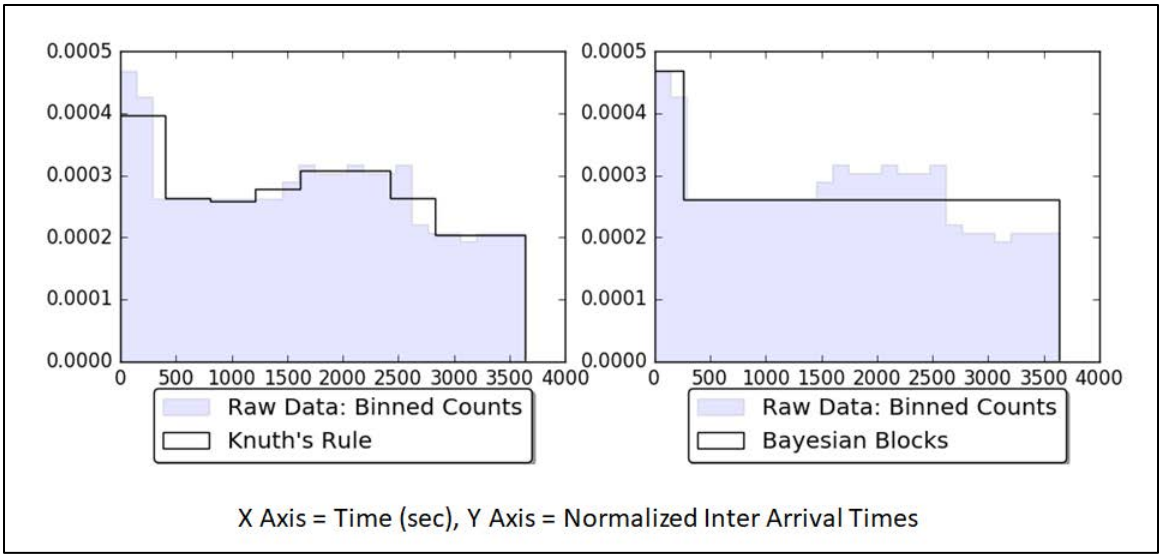

Figure 10: Output of Bayesian Blocks and Knuth's Algorithm for the 500 Sample Set

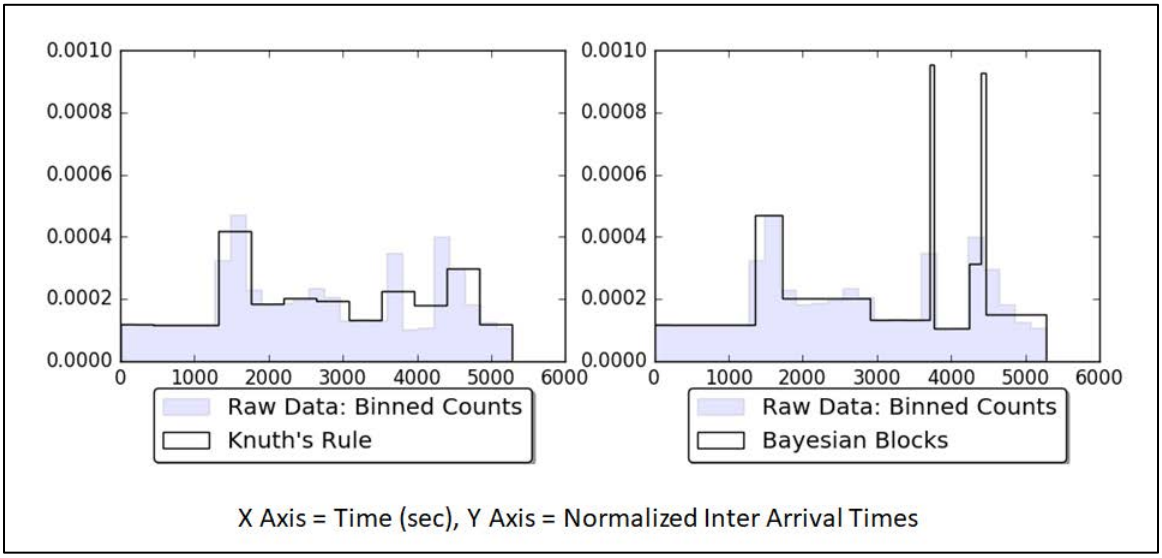

Figure 11: Blocks and Knuth's Algorithm for the 1000 Sample Set

From the visual inspection of the generated plots, we can infer that the Bayesian Blocks algorithm tends to produce little to no change-points in datasets of lesser samples. The 200 sample dataset produced a single bin when processed through Bayesian Blocks program; this is due to the inter-arrival times of scans being too similarly spaced. When larger inter-arrival time deviation occurs, as visible from the 400 and 1000 sample dataset plots, the Bayesian Blocks algorithm produces more significant change-points. This, of course, is not observable in Knuth's Rule plots as the bin size is constant and is determined over the entire dataset without utilizing a fitness function (varying bin width).

When these results are examined with the raw data plotted in the background of each graph, a clear distinction can be immediately noticed. If the data is too uniform or is without any anomalies, the Bayesian Blocks algorithm produces a single or a couple of bins with similar heights. Therefore, we can deduce that any anomaly or inconsistency in the inter-arrival times of scans would cause spikes (more bins of different heights) in our Bayesian Blocks plots, thus suggesting that our Bayesian Blocks algorithm can be used as a mechanism for anomaly detection in RFID network traffic. The histograms of inter-arrival gaps for each sample of scans we have generated are shown in Figure 12. The difference in plot shapes between samples is suggestive of differences in the traffic characteristics for each sample.

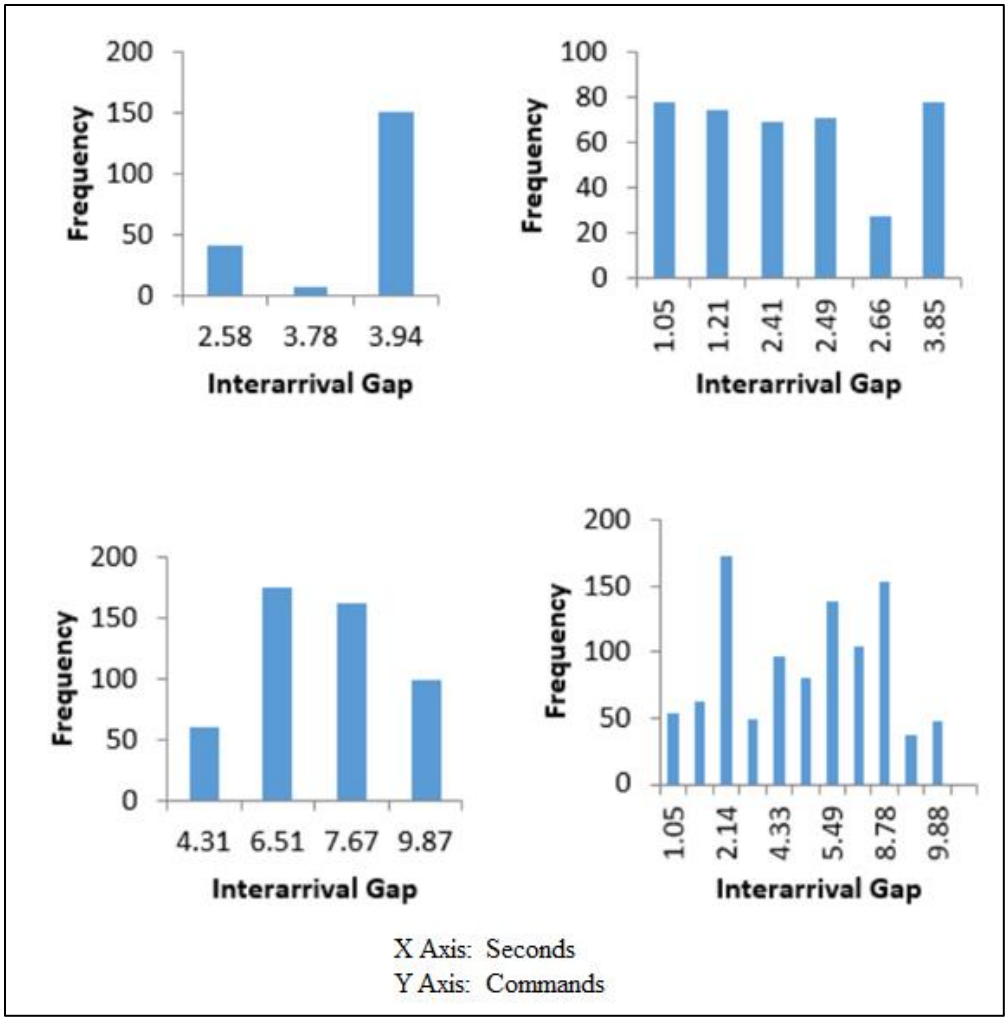

Figure 12: Inter-Arrival Gap Histograms: 200 to 1000 Frames Per Sample

4.2 Further Testing and Anomaly Detection

To further test the concept of how the classification of inter-arrival times through Bayesian Blocks and Knuth's Rule can be utilized in anomaly detection in RFID networks, a simple MATLAB program was developed to perform a graphical comparison between the two methods plotted against the raw data. The goal of the plots is to visualize the data and apply an anomaly detection filter all the way from 0% anomalous

assumption to 100%. The filter is a sliding filter, which means it sweeps the data from 0% to 100% and plots the result accordingly. When the result is plotted in Figure 13 below, it can be clearly seen that the Bayesian Blocks algorithm performs better than Knuth's Rule given the fractional assumed anomaly rate.

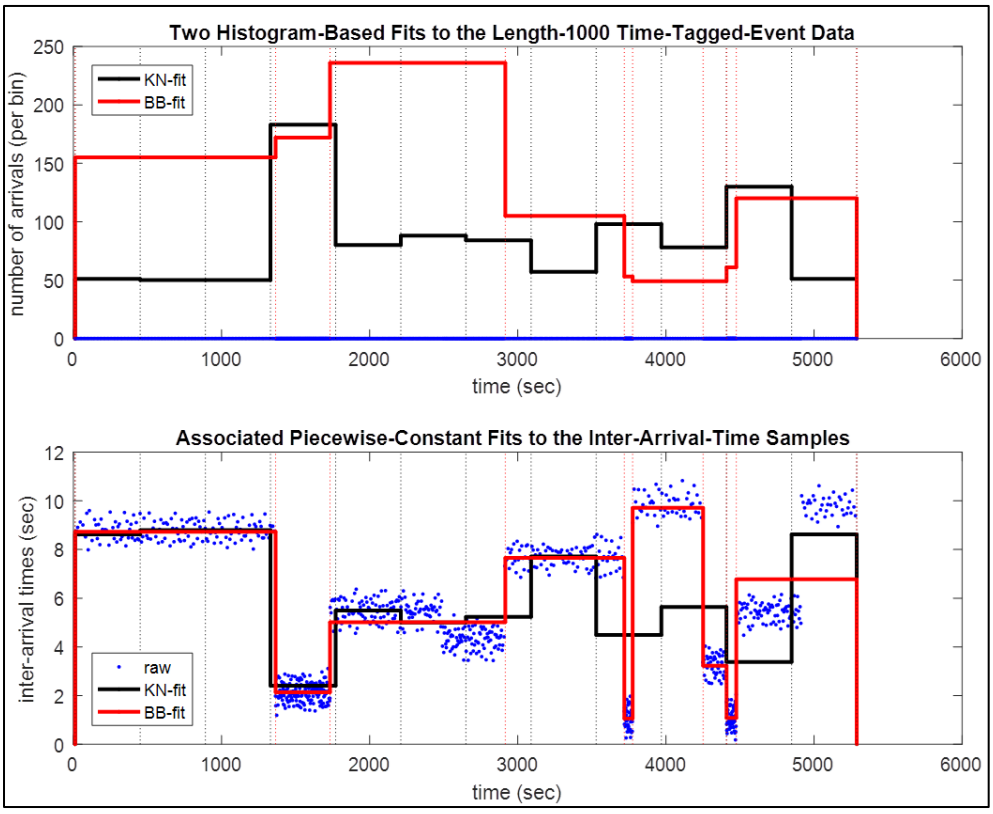

Figure 13: Histogram-based Fits to Time-Tagged Event Data and Associated Piecewise-Constant Fits to Inter-Arrival Time Samples

In Figure 13, both Knuth's Rule and Bayesian Blocks algorithms were plotted for the 1000 samples dataset over each other. The top graph in Figure 13 shows the data plotted as-is in a histogram fashion and the second graph shows the data plotted as piecewise-constant fits to the inter-arrival time samples. It is clear from the bottom graph in Figure 13 that the Bayesian Blocks algorithm is following the raw data more accurately than

Knuth's Rule. As visible, abrupt change in inter-arrival times causes change-points to occur which in turn draws a bin closest to the real representation of the raw data. Since Knuth's Rule decides on an optimal constant bin-width, the contour of the raw data cannot be followed accurately all the time; however, with Bayesian Blocks bin-width is variable which solves this problem.

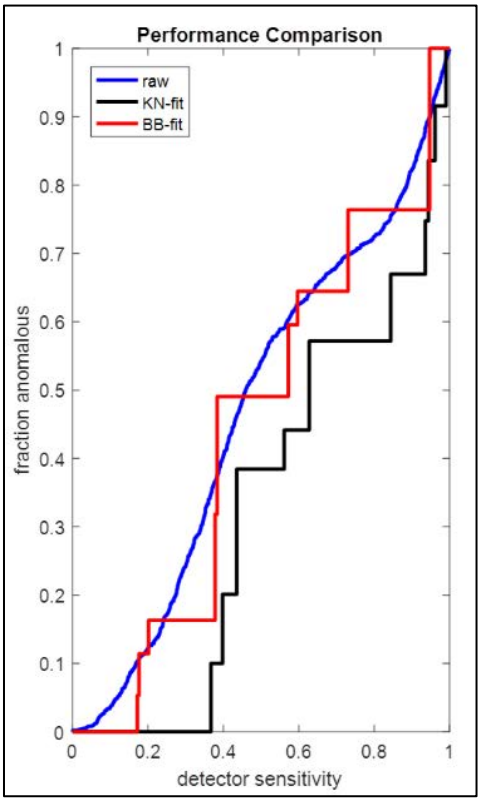

Figure 14: Performance Comparison Graph

In the third and final graph in Figure 14, a detector sensitivity-based performance comparison that sweeps the lower and upper bounds of the inter-arrival times in the previous graph was created. The sweep starts at 0% and ends at 100%. The anomaly detection analysis starts by computing (from the TTE data) the minimum, mean and maximum inter-arrival times. It then assumes a sensitivity parameter p in [0,1], whose

value corresponds to an accept region [lower bound a, upper bound b] on the observed inter-arrival times via the following scheme:

- lower bound a = average x – (1-p) * (average x – minimum x)
- upper bound b = average x + (1-p) * (maximum x – average x)

p = 0 corresponds to region [minimum x, maximum x] that accepts all samples as normal, while p = 1 corresponds to region [average x, average x] that rejects all samples. This is essentially a significance test assuming the empirical inter-arrival rate distribution. The performance comparison sweeps p from 0 to 1 and shows that the fraction of samples accepted using the Bayesian Blocks algorithm fit is "better" than the fraction accepted using the Knuth's Rule fit, where "better" is in the sense of similarity to the fraction of samples accepted using the raw samples during that same sweep of p. This performance analysis can run on all our datasets; however, since the 1000 samples dataset contains abrupt changes (anomalies), the performance analysis works best on it.

Chapter 5

FINAL OBSERVATION

5.1 Conclusion and Recommendation

This study presents an evaluation of the performance of Knuth's Rule and Bayesian Blocks algorithms when used to model RFID traffic and detect traffic anomalies. The datasets used in this study consist of time series of binned RFID command counts. Overall, both algorithms produced piecewise constant models that detected significant changes in the rate of command counts over time. Hence, the resulting plots appear to follow the contours describing the variations in the rate found in raw data.

As differences in traffic patterns are present, if the histograms of two different sets of RFID traces form visually different plot shapes, we safely conclude that both techniques could be potentially useful to model traffic produced by various RFID applications and detect traffic anomalies given a well-defined detector method as shown above. Based on our results, we believe and recommend that these algorithms be adapted and applied to the further design of anomaly detection systems.

REFERENCES

Print Publications:

[Alkadi16]
Alkadi, A., Prodanoff, Z., & Kreidl, P. (2016, November 24). Piece-Wise Constant Models for RFID Traffic. In 2016 IEEE International Conference on RFID Technology and Applications (RFID-TA). Foshan, China, 21-23 Sept. 2016. Foshan, China: IEEE. doi:10.1109/RFID-TA.2016.7750733.

[Angeles05]
Angeles, R. (2005). RFID Technologies: Supply-chain Applications and Implementation Issues. Information Systems Management [Electronic version], 22(1), 51–65. doi:10.1201/1078/44912.22.1.20051201/85739.7.

[Arnitz10]
Arnitz, D. (2010, September 30). UWB Channel Sounding for Ranging and Positioning in Passive UHF RFID [Electronic version]. ELECTRONICS LETTERS, 46(20). doi:10.1049/el.2010.1703.

[Beach04]
Beach, A., & Modaff, M. (2004). Network Traffic Anomaly Detection and Characterization. In Northwestern University. Retrieved from http://cs.northwestern.edu/~ajb200/anomaly%20detection%20paper%201.0.pdf, last accessed June 10, 2017.

[Chen11]
Chen, Y., Woo, W. L., & Wang, C. X. (2011). Channel Modeling of Information Transmission Over Cognitive Interrogator-Sensor Networks [Electronic version]. IEEE Transactions on Vehicular Technology, 60(1), 2–15. doi:0.1109/TVT.2010.2089545.

[Cheng10]
Cheng, S., Tom, K., Thomas, L., & Pecht, M. (2010). A Wireless Sensor System for Prognostics and Health Management [Electronic version]. IEEE Sensors Journal, 10(4), 856–862. doi:10.1109/JSEN.2009.2035817.

[Choudhury14]
Choudhury, S. H., & Smith, M. (2014, November 20). Quantitative Comparison of Indoor RFID Channel Models Using Bootstrap Techniques. In 2014 IEEE Wireless Communications and Networking Conference (WCNC). Istanbul, Turkey, 6-9 April 2014. Istanbul, Turkey: IEEE. doi:10.1109/WCNC.2014.6951921.

[David09]
David, S., & Rice, K. (2009). Bayesian Statistics [Electronic version]. Scholarpedia, 4(8). doi:10.4249/scholarpedia.5230.

[Dunnebeil11]
Dunnebeil, S., Kobler, F., Koene, P., Leimeister, J. M., & Krcmar, H. (2011). Encrypted NFC Emergency Tags Based on the German Telematics Infrastructure. In 2011 Third International Workshop on Near Field Communication. Hagenberg, Austria, 22-22 Feb. 2011. Hagenberg, Austria: IEEE. doi:10.1109/NFC.2011.18.

[Farooq12]
Farooq, M. U., Asif, M., Nabi, S. W., & Qureshi, M. A. (2012). Optimal Adjustment Parameters for EPC Global RFID Anti-collision Q-Algorithm in Different Traffic Scenarios. In 2012 10th International Conference on Frontiers of Information Technology. Islamabad, India, 17-19 Dec. 2012. Islamabad, India: IEEE. doi:10.1109/FIT.2012.61.

[Faruqui08]
Faruqui, R. A., & Tayab, E. (2008). RFID: Importance, Privacy/Security Issues and Implementation. In 2008 International Symposium on Biometrics and Security Technologies. Islamabad, Pakistan, 23-24 April 2008. Islamabad, Pakistan: IEEE. doi:10.1109/ISBAST.2008.4547668.

[Gaitan12]
Gaitan, A. M., Popa, V., Gaitan, V. G., Petrariu, A. I., Lavric, A., & Gherasim, S. A. (2012). RFID Network Yraffic Analysis Based on an Empirical Model. In 2012 9th International Conference on Communications (COMM). Bucharest, Romania, 21-23 June 2012. Bucharest, Romania: IEEE. doi:10.1109/ICComm.2012.6262549.

[Gu05]
Gu, Y., McCallum, A., & Towsley, D. (2005, October). Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation [Electronic version]. ACM SIGCOMM, 5(1), 32.

[Hameed14]
Hameed, S., Hameed, B., Hussain, S. A., & Khalid, W. (2014). Lightweight Security Middleware to Detect Malicious Content in NFC Tags or Smart Posters. In Lightweight Security Middleware to Detect Malicious Content in NFC Tags or Smart Posters. Beijing, China, 24-26 Sept. 2014. Bucharest, Romania: IEEE. doi:10.1109/TrustCom.2014.118.

[Haselsteiner06]
Haselsteiner, E., & Breitfuß K. (2006). Security in Near Field Communication (NFC). In Workshop on RFID Security. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=9C388C9C13A2F5BB09C7F56D1AEBAC62?doi=10.1.1.475.3812&rep=rep1&type=pdf, last accessed June 10, 2017.

[Hongwei13]
Hongwei, D. (2013, August). NFC Technology: Today and Tomorrow [Electronic version]. International Journal of Future Computer and Communication, 2(4), 351-354. doi:10.7763/IJFCC.2013.V2.183.

[ISO11]
ISO/IEC 29143. (2011). ISO/IEC 29143:2011 - Information technology -- Automatic Identification and Data Capture Techniques - Air Interface Specification for Mobile RFID Interrogators. Retrieved from https://www.iso.org/standard/45166.html, last accessed June 10, 2017.

[Knuth13]
Knuth, K. (2013). Optimal Data-Based Binning for Histograms. Albany, NY: University at Albany (SUNY). Retrieved from https://arxiv.org/pdf/physics/0605197.pdf, last accessed June 10, 2017.

[Li11]
Li, G. (2011, May 5). Bandwidth Dependence of CW Ranging to UHF RFID Tags in Severe Multipath Environments. In 2011 IEEE International Conference on RFID. Orlando, FL, USA, 12-14 April 2011. Orlando, FL, USA: IEEE. doi:10.1109/RFID.2011.5764631.

[Malison08]
Malison, P., Promwong, S., Sukutamatanti, N., & Banpotjit, T. (2008). Indoor Measurement and Modeling of RFID Transmission Loss at 5.8 GHz With Human Body. In 2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. Krabi, Thailand, 14-17 May 2008. Krabi, Thailand: IEEE. doi:10.1109/ECTICON.2008.4600461.

[Samiul11]
Samiul H. & Michael S., Quantitative Comparison of Indoor RFID Channel Models Using Bootstrap Techniques. In 2014 IEEE Wireless Communications and Networking Conference (WCNC). Istanbul, Turkey, 6-9 April 2014. Istanbul, Turkey: IEEE. doi:10.1109/WCNC.2014.6951921.

[Scargle98]
Scargle, J. (1998, September). Studies in Astronomical Time Series Analysis. V. Bayesian Blocks, a New Method to Analyze Structure in Photon Counting Data [Electronic version]. The Astrophysical Journal, 504(1), 405-418. doi:10.1086/306064.

[Smietanka12]
Smietanka, G., & Götze, J. (2012, November 20). Modeling and Simulation of MISO Diversity for UHF RFID Communication. In 2012 Federated Conference on Computer Science and Information Systems (FedCSIS). Wroclaw, Poland, 9-12 Sept. 2012. Wroclaw, Poland: IEEE.

[Wald92]
Wald, A. (1992, April 28). Sequential Tests of Statistical Hypotheses [Electronic version]. The Annals of Mathematical Statistics, 16(2), 117-186. doi:10.1214/aoms/1177731118.

[Wang09]
Wang, H., Pei, C., & Zheng, F. (2009, December). Performance Analysis and Test for Passive RFID System at UHF Band [Electronic version]. The Journal of China Universities of Posts and Telecommunications, 16(6), 49-56. doi:10.1016/S1005-8885(08)60288-5.

[Yang11]
Yang, H., Guo, J., & Deng, F. (2011). Collaborative RFID Intrusion Detection With an Artificial Immune System [Electronic version]. Journal of Intelligent Information Systems, 36(1), 1–26. doi:10.1007/s10844-010-0118-3.

APPENDIX A

Bayesian Blocks and Knuth's Rule Python Code

```
import numpy as np
from scipy import stats
from matplotlib import pyplot as plt
from astroML.plotting import hist
from astroML.density_estimation import bayesian_blocks
from astroML.density_estimation import knuth_bin_width

# Experimental TTE data collected from mobile phone NFC tag
reader
# The un-commented set will process when this script is run
# This data was manually imported from the mobile app's
database

# Set 1
# 100 samples 30 sec approx between arrivals
# data_set = [8.96, 14.72, 19.62, 26.57, 34.90, 38.15,
45.55, 50.06, 57.85, 63.89, 70.61, 74.65, 83.11, 89.22,
95.56, 102.73, 107.14, 113.06, 119.45, 127.06, 130.16,
139.52, 144.62, 151.47, 158.29, 164.83, 167.57, 174.16,
183.29, 187.27, 193.27, 200.11, 206.61, 214.12, 220.56,
225.24, 230.35, 238.28, 243.79, 250.35, 254.13, 263.56,
266.44, 272.70, 279.51, 285.96, 294.60, 299.90, 303.92,
311.91, 316.85, 322.42, 328.60, 338.13, 343.96, 349.46,
356.16, 359.12, 366.50, 371.64, 377.84, 384.33, 392.71,
398.14, 405.36, 409.49, 415.73, 422.68, 430.55, 436.31,
442.59, 446.96, 454.20, 458.32, 467.17, 473.58, 479.53,
486.42, 491.56, 498.94, 503.97, 508.35, 515.45, 522.09,
529.09, 533.46, 539.99, 545.98, 551.65, 557.11, 566.74,
573.01, 577.75, 585.81, 591.70, 596.66, 602.69, 610.21,
613.88, 621.25]
# ymax = 0.0018

# Set 2
# 200 samples 3 sec approx between arrivals
# data_set = [4.01, 7.41, 10.30, 12.17, 13.87, 17.24,
20.28, 22.74, 26.05, 29.62, 30.35, 34.72, 37.01, 39.93,
40.63, 43.40, 44.34, 48.23, 49.88, 54.97, 58.05, 57.69,
61.74, 63.58, 65.45, 69.68, 69.76, 74.93, 77.81, 79.36,
83.18, 84.30, 87.93, 88.03, 93.45, 95.67, 96.37, 99.70,
103.49, 106.72, 109.34, 109.35, 113.56, 117.38, 120.40,
125.33, 128.68, 135.75, 136.15, 141.63, 146.55, 150.94,
```

```
152.72, 155.66, 160.29, 165.70, 167.71, 174.47, 178.98,
179.85, 186.45, 190.28, 193.47, 197.00, 201.80, 203.41,
209.36, 210.82, 216.17, 219.94, 225.69, 227.46, 231.01,
235.05, 238.81, 244.67, 249.77, 251.71, 256.21, 260.76,
262.67, 269.91, 272.72, 274.30, 281.77, 285.45, 288.01,
292.23, 296.12, 300.36, 304.21, 305.45, 311.55, 313.35,
318.04, 321.53, 325.10, 330.41, 334.40, 337.06, 344.08,
347.95, 349.51, 352.80, 357.72, 361.95, 365.86, 372.25,
374.81, 376.48, 381.90, 387.11, 390.73, 392.42, 397.92,
400.76, 406.47, 408.80, 412.09, 415.87, 422.63, 427.41,
430.22, 432.92, 438.23, 439.82, 444.99, 449.20, 453.10,
457.36, 460.17, 466.40, 469.62, 471.54, 475.27, 479.87,
483.35, 489.60, 491.81, 498.13, 501.20, 505.49, 509.14,
511.08, 517.84, 520.19, 523.16, 526.38, 531.94, 535.94,
539.88, 541.85, 549.57, 549.68, 555.36, 557.98, 561.82,
566.68, 570.11, 573.32, 581.18, 581.62, 588.85, 591.47,
594.38, 597.77, 603.17, 607.13, 612.36, 613.03, 617.65,
622.44, 625.93, 629.80, 633.62, 637.40, 642.81, 647.29,
651.36, 654.89, 656.51, 662.62, 664.94, 671.60, 675.30,
679.15, 682.09, 685.46, 689.27, 691.51, 699.37, 699.48,
705.80, 709.75, 711.07, 714.78, 719.71, 726.19, 728.10,
730.91]
# ymax = 0.0020

# Set 3
# 400 samples 3 sec approx between arrivals
# data_set = [2.87, 6.13, 9.33, 13.45, 16.08, 19.57, 22.41,
23.32, 27.22, 29.56, 32.81, 34.21, 36.49, 39.45, 43.75,
46.46, 48.37, 50.68, 54.43, 54.91, 58.88, 61.20, 63.04,
64.45, 67.42, 71.46, 74.57, 75.59, 78.33, 80.28, 84.28,
87.03, 89.81, 93.59, 94.80, 97.61, 99.77, 102.64, 102.45,
105.61, 110.87, 109.59, 114.36, 116.44, 120.76, 119.56,
123.43, 124.97, 127.68, 132.02, 132.82, 135.25, 138.38,
139.49, 142.54, 147.00, 147.62, 153.06, 155.76, 157.48,
157.29, 162.82, 165.23, 165.16, 170.24, 170.60, 172.63,
176.52, 179.92, 183.11, 182.47, 184.34, 190.10, 191.46,
192.23, 196.21, 199.48, 201.12, 204.26, 206.00, 210.49,
212.64, 212.60, 216.60, 217.65, 221.88, 225.02, 227.58,
226.67, 230.69, 233.84, 234.74, 238.49, 240.51, 244.37,
244.33, 250.01, 249.25, 253.17, 254.50, 261.39, 263.73,
267.88, 272.77, 274.10, 280.99, 283.38, 287.22, 290.63,
294.58, 299.72, 300.79, 305.90, 311.15, 311.77, 319.43,
320.49, 324.96, 328.66, 334.76, 337.44, 339.47, 344.01,
349.98, 350.79, 357.63, 361.40, 362.75, 366.18, 372.22,
375.66, 377.81, 384.05, 386.27, 391.25, 395.87, 397.25,
403.18, 407.28, 411.09, 412.23, 415.75, 421.87, 425.56,
427.79, 432.56, 436.43, 442.27, 442.63, 448.62, 451.97,
```

```
455.51, 461.67, 465.11, 468.82, 473.41, 476.79, 478.42,
484.62, 485.60, 492.41, 493.07, 499.83, 502.99, 506.87,
510.18, 512.45, 517.91, 520.29, 527.08, 531.26, 531.50,
537.43, 540.43, 545.37, 547.07, 550.96, 557.63, 556.46,
560.37, 558.80, 562.87, 563.75, 564.15, 566.01, 564.89,
566.04, 569.07, 569.01, 572.56, 573.87, 573.64, 576.05,
577.17, 576.20, 577.60, 579.02, 578.60, 581.01, 583.58,
585.14, 583.38, 584.77, 586.46, 588.19, 588.64, 592.69,
590.65, 592.68, 596.88, 597.67, 596.99, 599.60, 600.80,
602.05, 604.21, 602.33, 605.41, 605.70, 606.21, 609.15,
609.32, 611.49, 613.02, 613.90, 613.39, 616.87, 616.95,
618.05, 618.51, 620.58, 620.36, 621.50, 625.91, 624.74,
628.32, 627.00, 629.27, 630.97, 629.97, 634.16, 632.62,
636.07, 637.37, 638.51, 638.19, 637.95, 642.31, 642.71,
643.67, 644.96, 647.26, 646.10, 649.85, 649.85, 650.19,
651.63, 653.47, 655.09, 654.22, 656.86, 655.66, 659.40,
658.12, 658.10, 660.66, 662.07, 661.40, 662.47, 665.46,
665.23, 665.83, 666.84, 668.65, 671.29, 672.05, 673.27,
672.07, 673.66, 674.49, 676.99, 678.75, 679.72, 680.15,
680.29, 679.97, 683.76, 681.85, 685.98, 687.51, 685.25,
687.63, 689.40, 690.53, 690.53, 691.07, 694.77, 693.89,
696.01, 698.18, 698.32, 699.06, 700.28, 701.23, 699.59,
703.69, 703.46, 705.75, 704.26, 706.36, 706.49, 707.83,
708.92, 711.70, 712.21, 714.73, 714.65, 713.75, 714.56,
717.83, 718.84, 721.06, 721.90, 723.15, 721.89, 724.24,
726.09, 727.51, 727.44, 726.45, 730.85, 731.86, 733.72,
735.81, 738.16, 739.96, 745.24, 745.22, 749.96, 752.46,
754.88, 756.26, 756.36, 761.37, 762.10, 763.17, 766.56,
769.08, 772.48, 773.27, 776.80, 778.93, 781.73, 783.61,
786.42, 788.83, 792.80, 794.78, 795.37, 796.91, 802.81,
803.89, 805.40, 808.85, 810.46, 814.89, 816.82, 816.85,
821.48, 822.77, 825.29, 827.11, 831.64, 833.03, 835.55,
838.56, 840.60, 842.50, 845.59, 846.22, 850.45, 851.96,
852.61, 858.34, 860.50, 860.05, 864.66, 867.42, 869.26,
871.98, 874.34, 874.94, 877.23, 880.46, 883.64, 886.53,
886.66, 891.22, 891.66, 896.09]
# ymax = 0.0025

# Set 4
# 500 samples 10 sec approx between arrivals
# data_set = [5.16, 11.43, 14.46, 17.57, 23.81, 27.75,
33.43, 35.58, 41.81, 46.73, 49.21, 54.99, 57.40, 62.19,
66.69, 69.17, 77.15, 78.25, 84.34, 86.50, 92.68, 97.65,
99.17, 104.16, 108.00, 114.58, 119.30, 122.52, 128.57,
130.55, 136.48, 140.94, 143.53, 147.53, 152.78, 157.04,
162.51, 164.84, 169.05, 174.94, 177.75, 184.65, 185.49,
189.97, 194.95, 199.48, 205.04, 207.42, 214.29, 219.33,
```

221.07, 224.29, 229.83, 235.12, 237.76, 242.63, 246.30,
251.16, 256.35, 260.01, 264.38, 268.58, 275.94, 285.13,
291.80, 301.72, 307.02, 313.70, 322.83, 331.42, 337.86,
345.32, 352.44, 362.46, 368.13, 374.97, 384.60, 389.96,
398.06, 408.43, 414.00, 421.02, 428.90, 436.17, 445.62,
454.53, 461.83, 467.95, 474.59, 483.66, 492.00, 501.12,
507.35, 514.53, 523.91, 531.04, 538.79, 545.23, 551.65,
562.53, 566.53, 576.72, 582.79, 593.12, 599.44, 608.04,
614.09, 623.07, 628.12, 635.97, 643.16, 651.10, 661.50,
669.36, 674.96, 682.13, 689.66, 699.59, 707.75, 713.87,
721.46, 727.42, 737.31, 745.85, 752.83, 761.74, 766.66,
773.95, 784.07, 791.84, 796.83, 807.83, 815.78, 822.48,
829.50, 838.61, 845.46, 850.38, 859.93, 868.29, 874.88,
884.70, 890.92, 896.96, 905.10, 914.85, 923.14, 928.55,
937.73, 943.31, 952.97, 958.89, 968.98, 975.15, 981.10,
988.70, 999.17, 1005.34, 1015.18, 1021.33, 1026.98,
1037.63, 1045.80, 1050.00, 1060.71, 1066.92, 1074.89,
1080.62, 1090.16, 1098.04, 1104.79, 1111.63, 1122.56,
1127.07, 1137.53, 1144.71, 1152.25, 1158.18, 1165.62,
1176.03, 1181.22, 1188.61, 1195.70, 1204.01, 1214.12,
1222.10, 1228.16, 1237.36, 1243.05, 1249.58, 1260.15,
1264.38, 1274.80, 1280.54, 1290.83, 1296.34, 1306.65,
1311.42, 1319.41, 1326.83, 1336.75, 1343.59, 1348.72,
1358.46, 1364.41, 1375.26, 1381.35, 1388.84, 1396.75,
1404.61, 1410.71, 1417.89, 1425.91, 1434.12, 1443.40,
1450.06, 1457.02, 1466.22, 1474.42, 1481.86, 1490.43,
1495.37, 1504.79, 1511.19, 1518.12, 1522.97, 1533.05,
1539.57, 1546.09, 1551.80, 1558.84, 1562.57, 1570.43,
1577.98, 1585.01, 1590.80, 1595.29, 1604.78, 1607.54,
1616.12, 1624.21, 1629.61, 1635.83, 1643.74, 1647.63,
1656.86, 1663.23, 1668.45, 1676.32, 1681.40, 1686.92,
1693.35, 1701.82, 1708.41, 1714.21, 1721.74, 1726.88,
1734.11, 1737.68, 1745.11, 1753.02, 1760.67, 1767.59,
1771.87, 1780.21, 1783.54, 1789.96, 1797.51, 1805.52,
1810.61, 1815.91, 1825.76, 1830.40, 1837.63, 1842.05,
1850.54, 1857.33, 1862.15, 1870.03, 1876.23, 1881.17,
1891.04, 1895.94, 1903.88, 1907.69, 1913.78, 1923.23,
1928.04, 1936.29, 1941.61, 1949.79, 1954.45, 1962.24,
1966.92, 1974.75, 1982.33, 1987.39, 1992.78, 1999.93,
2007.88, 2012.44, 2020.85, 2026.72, 2031.45, 2037.62,
2047.12, 2052.36, 2058.69, 2064.13, 2072.48, 2078.73,
2083.50, 2089.37, 2098.64, 2105.54, 2111.93, 2119.09,
2124.48, 2130.45, 2138.53, 2141.66, 2149.65, 2154.93,
2164.53, 2170.18, 2174.27, 2183.93, 2189.26, 2193.68,
2201.07, 2208.12, 2214.93, 2220.68, 2228.69, 2234.34,
2241.02, 2245.78, 2252.89, 2261.30, 2265.64, 2275.36,
2280.20, 2288.30, 2293.91, 2300.75, 2305.15, 2311.43,

```
2320.12, 2326.54, 2333.85, 2337.71, 2346.40, 2351.22,
2358.94, 2363.00, 2369.15, 2376.16, 2385.97, 2389.87,
2397.17, 2403.15, 2410.80, 2417.94, 2422.54, 2429.84,
2437.20, 2440.86, 2447.82, 2454.80, 2463.74, 2469.86,
2476.40, 2482.23, 2487.93, 2493.03, 2502.69, 2508.57,
2512.68, 2522.43, 2527.51, 2534.78, 2541.88, 2547.14,
2551.66, 2558.86, 2564.81, 2572.24, 2577.92, 2585.77,
2592.75, 2599.19, 2605.61, 2611.96, 2619.52, 2626.55,
2629.76, 2636.49, 2645.25, 2649.15, 2659.46, 2669.08,
2677.03, 2686.41, 2698.41, 2705.86, 2716.97, 2727.76,
2736.88, 2746.15, 2755.41, 2766.84, 2777.15, 2785.93,
2795.26, 2806.94, 2815.16, 2823.66, 2835.89, 2843.77,
2854.97, 2866.50, 2875.20, 2885.95, 2895.79, 2902.40,
2913.57, 2925.32, 2935.73, 2942.55, 2954.64, 2964.76,
2971.40, 2982.38, 2992.59, 3001.75, 3011.21, 3023.69,
3032.39, 3042.85, 3053.77, 3060.31, 3073.16, 3083.03,
3090.18, 3101.77, 3111.37, 3120.08, 3129.50, 3143.15,
3149.22, 3160.22, 3171.58, 3179.62, 3189.86, 3202.36,
3209.71, 3220.36, 3230.05, 3238.75, 3249.23, 3259.66,
3267.80, 3280.05, 3289.57, 3300.90, 3306.95, 3320.31,
3328.51, 3337.29, 3346.54, 3356.56, 3369.72, 3376.25,
3389.59, 3397.11, 3407.50, 3415.58, 3426.94, 3439.09,
3445.19, 3457.07, 3465.48, 3478.38, 3487.52, 3498.30,
3505.51, 3516.28, 3524.17, 3535.22, 3546.70, 3557.45,
3566.31, 3574.50, 3583.47, 3597.17, 3605.62, 3615.81,
3624.29, 3633.03, 3645.51]
# ymax = 0.0005

# Set 5
# 650 samples 5 sec approx between arrivals
# data_set = [5.17, 7.18, 12.42, 13.75, 17.48, 24.03,
25.46, 29.97, 34.30, 37.51, 41.18, 43.38, 46.51, 48.69,
52.56, 55.55, 58.70, 63.68, 65.77, 70.25, 73.58, 78.42,
80.60, 81.92, 88.69, 89.39, 95.15, 97.46, 101.36, 104.76,
109.12, 112.25, 114.90, 119.08, 121.20, 126.33, 125.80,
129.92, 134.41, 138.58, 142.46, 146.31, 147.44, 153.07,
156.78, 159.44, 163.21, 165.10, 168.45, 171.69, 175.19,
179.98, 183.25, 186.89, 187.61, 192.35, 197.21, 198.77,
203.38, 207.41, 208.46, 214.18, 217.59, 221.36, 224.68,
224.88, 230.14, 232.02, 236.58, 241.89, 243.99, 246.55,
251.99, 252.07, 255.48, 259.46, 262.34, 266.37, 270.89,
275.33, 279.35, 282.34, 283.29, 286.43, 289.97, 293.19,
295.98, 300.78, 304.26, 308.39, 309.65, 313.76, 317.04,
322.73, 326.28, 327.16, 333.34, 335.61, 339.00, 343.98,
347.13, 350.29, 353.60, 356.35, 359.58, 362.42, 365.98,
370.58, 373.18, 377.29, 380.56, 383.87, 387.31, 387.65,
391.67, 397.25, 401.19, 409.03, 411.03, 415.75, 422.77,
```

424.98, 431.13, 436.65, 439.08, 445.67, 448.86, 453.33,
458.95, 463.09, 468.33, 473.87, 479.95, 482.52, 486.63,
492.06, 497.06, 503.53, 508.97, 511.86, 515.58, 521.66,
524.37, 531.20, 536.94, 540.15, 546.04, 551.60, 555.45,
559.73, 562.95, 567.21, 574.84, 579.61, 583.53, 588.29,
594.33, 599.08, 602.01, 608.06, 610.41, 615.44, 619.69,
626.81, 632.41, 633.84, 639.04, 645.35, 651.20, 656.56,
658.44, 663.28, 669.58, 675.63, 679.08, 684.34, 689.65,
691.98, 697.08, 703.56, 707.63, 711.50, 717.23, 723.04,
726.83, 729.49, 737.43, 739.21, 745.60, 748.17, 754.97,
760.38, 762.57, 767.05, 772.76, 780.31, 784.56, 787.65,
793.44, 798.97, 802.08, 809.11, 813.11, 816.88, 822.78,
824.32, 829.22, 836.60, 839.36, 846.54, 851.09, 856.45,
857.79, 863.97, 869.34, 873.02, 876.57, 883.92, 888.59,
892.07, 895.86, 904.28, 906.59, 910.06, 918.23, 921.24,
926.11, 931.51, 936.88, 939.31, 946.76, 949.49, 955.56,
960.80, 964.02, 967.74, 973.79, 976.67, 984.37, 987.79,
991.39, 998.52, 1000.34, 1005.93, 1010.48, 1017.50,
1020.66, 1027.88, 1030.46, 1034.28, 1039.52, 1043.34,
1050.01, 1054.48, 1057.82, 1063.16, 1067.33, 1074.04,
1079.27, 1080.93, 1082.47, 1082.07, 1083.85, 1085.89,
1083.52, 1085.93, 1088.59, 1090.13, 1089.41, 1090.66,
1090.71, 1094.64, 1094.93, 1094.33, 1095.28, 1098.77,
1098.83, 1099.28, 1098.94, 1103.02, 1105.05, 1102.94,
1105.14, 1107.53, 1109.20, 1109.52, 1111.39, 1109.84,
1111.14, 1113.76, 1112.65, 1114.57, 1116.05, 1119.17,
1120.21, 1120.93, 1122.54, 1120.99, 1123.73, 1125.89,
1125.72, 1128.26, 1126.17, 1126.92, 1129.58, 1130.66,
1132.94, 1131.84, 1136.44, 1134.45, 1137.17, 1138.58,
1140.60, 1140.53, 1140.60, 1143.22, 1143.52, 1144.01,
1145.31, 1147.94, 1148.35, 1150.56, 1149.96, 1151.70,
1153.39, 1153.61, 1155.47, 1156.04, 1158.42, 1159.92,
1160.96, 1160.37, 1161.62, 1163.84, 1163.02, 1166.39,
1163.99, 1166.12, 1168.77, 1169.63, 1169.72, 1173.29,
1171.04, 1173.54, 1174.98, 1177.00, 1178.18, 1177.65,
1177.46, 1182.06, 1182.28, 1180.76, 1185.37, 1183.13,
1184.04, 1186.70, 1189.54, 1188.00, 1191.04, 1191.60,
1194.65, 1192.21, 1195.56, 1194.38, 1195.79, 1198.84,
1198.81, 1199.32, 1202.28, 1203.14, 1202.74, 1204.66,
1205.29, 1205.41, 1206.62, 1207.49, 1211.14, 1215.27,
1220.80, 1227.40, 1233.70, 1241.37, 1246.67, 1252.59,
1259.22, 1263.34, 1269.72, 1275.06, 1280.93, 1288.30,
1292.65, 1298.06, 1304.20, 1311.39, 1318.20, 1325.18,
1331.13, 1335.04, 1342.84, 1348.12, 1353.30, 1358.14,
1365.29, 1369.45, 1375.46, 1382.07, 1388.02, 1395.61,
1401.37, 1406.98, 1413.91, 1419.92, 1423.01, 1431.22,
1436.07, 1444.47, 1447.45, 1454.17, 1459.96, 1465.48,

```
1470.55, 1477.87, 1485.80, 1492.07, 1494.48, 1501.59,
1509.40, 1513.57, 1518.06, 1527.65, 1532.95, 1538.69,
1543.03, 1550.92, 1555.94, 1559.73, 1566.86, 1574.74,
1578.83, 1584.58, 1590.12, 1598.32, 1602.38, 1608.99,
1614.44, 1620.75, 1626.80, 1632.79, 1637.10, 1645.36,
1650.80, 1655.47, 1663.80, 1669.78, 1675.86, 1679.04,
1685.83, 1691.08, 1697.71, 1700.67, 1704.81, 1705.29,
1709.17, 1708.68, 1711.76, 1714.61, 1715.04, 1720.72,
1721.63, 1725.83, 1724.44, 1729.25, 1730.35, 1733.62,
1737.06, 1738.38, 1740.94, 1742.19, 1743.37, 1748.63,
1748.20, 1750.85, 1755.59, 1755.13, 1758.57, 1758.97,
1761.46, 1764.48, 1766.03, 1769.84, 1772.74, 1773.28,
1776.89, 1779.26, 1781.35, 1784.22, 1785.98, 1790.05,
1790.46, 1794.78, 1795.91, 1798.28, 1800.86, 1800.64,
1805.45, 1808.46, 1809.80, 1809.89, 1813.07, 1817.43,
1817.86, 1819.48, 1823.84, 1824.41, 1826.50, 1831.27,
1833.01, 1834.03, 1838.34, 1841.19, 1843.24, 1845.33,
1846.41, 1848.07, 1852.82, 1855.25, 1860.01, 1864.13,
1868.66, 1874.40, 1877.63, 1883.08, 1889.48, 1893.71,
1897.77, 1902.39, 1907.81, 1909.73, 1917.75, 1919.81,
1926.50, 1929.95, 1936.65, 1940.96, 1943.79, 1950.65,
1953.33, 1956.90, 1962.42, 1966.72, 1973.77, 1977.92,
1979.97, 1987.61, 1989.39, 1996.97, 2001.40, 2005.99,
2009.26, 2013.31, 2019.85, 2021.77, 2027.89, 2032.90,
2035.99, 2040.88, 2047.02, 2050.82, 2055.72, 2059.87,
2065.02, 2068.63, 2074.95, 2080.03, 2083.22, 2088.30,
2092.39, 2097.72, 2101.38, 2106.56, 2110.75, 2117.08,
2123.10, 2128.45, 2130.68, 2134.96, 2140.99, 2145.23,
2150.29, 2153.43, 2159.88, 2163.96, 2169.60, 2173.33,
2177.80, 2181.15, 2185.61, 2191.61, 2196.87, 2201.81,
2205.25, 2209.65, 2214.27, 2221.68, 2222.90, 2230.20,
2231.51, 2235.54, 2238.28, 2242.41, 2245.58, 2249.39,
2254.31, 2258.27, 2260.29, 2262.48, 2267.10, 2269.42,
2277.18, 2281.54, 2288.22, 2294.36, 2299.19, 2308.06,
2312.88, 2318.15, 2322.73, 2328.71, 2334.66, 2340.87,
2347.41, 2353.79, 2360.41, 2364.51, 2369.50, 2377.96,
2380.95, 2389.76, 2396.61, 2398.88, 2405.37, 2411.27,
2416.40, 2425.25, 2429.50, 2437.57, 2442.05, 2448.65,
2451.40, 2460.90]
# ymax = 0.0014

# Set 6
# 1000 samples 10 sec approx between arrivals
data_set = [11.39, 19.94, 28.70, 37.61, 46.22, 55.35,
64.02, 73.03, 81.63, 90.15, 99.64, 107.63, 117.23, 125.91,
134.28, 142.91, 152.14, 161.15, 169.60, 178.63, 186.83,
196.08, 204.91, 213.62, 222.44, 231.24, 239.89, 248.07,
```

257.61, 266.28, 275.18, 283.63, 292.77, 301.08, 310.02,
319.07, 327.48, 336.01, 344.61, 354.13, 362.28, 371.44,
379.92, 388.75, 397.79, 406.27, 415.69, 424.07, 433.18,
441.62, 450.02, 459.50, 467.81, 476.34, 485.63, 494.41,
502.98, 511.44, 520.94, 529.37, 538.16, 547.04, 556.20,
564.73, 573.48, 581.75, 590.58, 599.81, 608.13, 617.21,
625.67, 634.77, 643.81, 652.59, 660.74, 670.01, 678.88,
687.20, 696.69, 705.55, 714.20, 722.61, 730.95, 740.17,
749.11, 757.48, 766.18, 775.31, 783.86, 793.09, 801.20,
810.48, 818.71, 827.75, 837.05, 845.50, 854.66, 863.59,
872.32, 880.65, 889.84, 898.71, 907.15, 915.49, 925.03,
933.70, 942.44, 951.32, 959.58, 968.28, 976.80, 986.07,
995.14, 1003.42, 1012.32, 1021.11, 1030.41, 1038.91,
1046.99, 1056.38, 1064.80, 1073.45, 1082.23, 1091.67,
1099.93, 1108.84, 1117.98, 1126.66, 1135.49, 1143.88,
1153.20, 1161.76, 1170.32, 1179.03, 1187.81, 1196.81,
1205.18, 1214.61, 1222.63, 1231.67, 1240.61, 1249.30,
1257.86, 1267.01, 1275.96, 1284.44, 1292.92, 1302.38,
1311.01, 1319.76, 1328.24, 1337.23, 1346.18, 1355.14,
1363.90, 1365.88, 1367.46, 1370.13, 1372.59, 1373.78,
1376.54, 1378.94, 1380.89, 1382.50, 1385.29, 1387.19,
1389.13, 1391.09, 1393.82, 1396.04, 1398.06, 1400.02,
1402.06, 1404.00, 1406.16, 1408.73, 1410.80, 1412.61,
1415.11, 1417.39, 1419.03, 1421.55, 1423.55, 1425.15,
1427.35, 1430.23, 1431.71, 1434.10, 1436.31, 1438.75,
1440.65, 1442.83, 1444.64, 1447.00, 1449.29, 1451.02,
1453.84, 1455.82, 1457.37, 1459.82, 1462.31, 1464.09,
1466.11, 1468.27, 1470.93, 1472.75, 1474.66, 1476.73,
1479.53, 1481.33, 1483.51, 1485.69, 1487.76, 1489.32,
1491.72, 1493.70, 1495.84, 1498.62, 1500.67, 1502.94,
1504.88, 1506.67, 1509.13, 1510.73, 1513.39, 1515.39,
1518.09, 1519.68, 1521.66, 1524.04, 1526.65, 1528.20,
1530.49, 1532.25, 1534.98, 1536.48, 1538.93, 1540.91,
1543.57, 1545.59, 1547.30, 1549.43, 1552.31, 1554.47,
1556.15, 1558.53, 1560.14, 1562.93, 1564.34, 1566.46,
1568.73, 1570.81, 1572.80, 1575.23, 1577.79, 1579.75,
1581.68, 1584.14, 1586.55, 1588.37, 1590.66, 1592.64,
1594.97, 1596.37, 1599.20, 1600.88, 1603.40, 1605.21,
1607.39, 1609.20, 1611.75, 1613.86, 1616.13, 1618.50,
1620.47, 1622.14, 1624.40, 1626.62, 1629.11, 1631.41,
1632.78, 1635.11, 1637.24, 1639.33, 1641.84, 1644.06,
1646.11, 1648.15, 1650.47, 1652.41, 1654.78, 1657.12,
1659.16, 1661.37, 1663.11, 1665.69, 1667.26, 1669.78,
1672.01, 1673.79, 1676.06, 1677.70, 1680.67, 1682.91,
1684.43, 1686.93, 1688.61, 1690.84, 1692.79, 1695.53,
1697.52, 1699.70, 1701.25, 1704.00, 1706.12, 1708.20,
1709.76, 1711.93, 1714.47, 1716.18, 1718.33, 1721.44,

1723.04, 1725.51, 1727.39, 1729.62, 1731.40, 1733.29,
1738.91, 1744.31, 1750.21, 1756.15, 1760.79, 1766.69,
1772.65, 1777.57, 1783.11, 1788.26, 1794.22, 1800.14,
1804.82, 1810.88, 1815.84, 1821.46, 1826.71, 1832.72,
1837.95, 1843.87, 1849.35, 1854.99, 1860.47, 1865.81,
1870.66, 1876.15, 1881.60, 1887.48, 1892.84, 1898.26,
1904.46, 1909.02, 1915.39, 1920.15, 1925.63, 1931.02,
1936.82, 1942.46, 1947.79, 1953.67, 1958.93, 1964.65,
1969.70, 1974.88, 1980.91, 1986.75, 1991.78, 1997.40,
2002.97, 2007.91, 2013.88, 2019.13, 2024.93, 2029.99,
2036.11, 2041.33, 2047.05, 2051.98, 2057.42, 2063.50,
2068.33, 2073.98, 2079.18, 2085.48, 2090.26, 2095.95,
2101.19, 2106.68, 2112.99, 2117.95, 2123.65, 2129.01,
2134.05, 2139.57, 2145.38, 2150.97, 2156.81, 2161.97,
2167.61, 2172.78, 2178.07, 2183.94, 2189.86, 2194.46,
2200.35, 2206.41, 2211.19, 2216.83, 2222.81, 2227.55,
2233.85, 2238.44, 2244.48, 2250.00, 2255.72, 2261.14,
2266.80, 2272.00, 2277.47, 2282.83, 2288.24, 2293.46,
2298.83, 2304.98, 2310.33, 2315.90, 2321.67, 2326.35,
2332.22, 2337.65, 2343.12, 2348.67, 2354.55, 2359.27,
2365.44, 2370.98, 2376.53, 2381.87, 2387.52, 2392.83,
2398.40, 2403.92, 2409.20, 2414.44, 2420.15, 2425.73,
2431.24, 2436.28, 2441.92, 2447.93, 2453.38, 2458.88,
2463.49, 2469.65, 2474.97, 2480.60, 2485.43, 2491.31,
2495.34, 2500.16, 2503.93, 2508.50, 2513.31, 2517.43,
2521.40, 2525.64, 2530.20, 2534.45, 2539.08, 2543.74,
2547.56, 2551.99, 2556.48, 2560.40, 2564.85, 2568.85,
2574.16, 2577.83, 2582.55, 2586.56, 2590.94, 2595.02,
2599.27, 2604.01, 2608.18, 2612.66, 2617.04, 2621.31,
2625.53, 2630.01, 2634.76, 2638.22, 2642.62, 2647.45,
2651.71, 2655.53, 2660.64, 2665.04, 2668.49, 2673.34,
2677.64, 2682.35, 2686.08, 2690.44, 2694.56, 2699.42,
2703.25, 2707.80, 2711.88, 2716.78, 2721.02, 2724.75,
2729.91, 2734.10, 2738.26, 2742.07, 2746.65, 2751.18,
2756.01, 2759.61, 2764.34, 2768.42, 2773.24, 2777.18,
2781.53, 2786.25, 2789.72, 2794.10, 2798.64, 2803.03,
2807.25, 2811.84, 2816.47, 2820.75, 2824.50, 2829.11,
2833.94, 2837.38, 2842.29, 2846.26, 2850.77, 2854.80,
2859.29, 2863.69, 2868.55, 2872.46, 2876.83, 2880.91,
2885.31, 2889.77, 2894.45, 2898.31, 2902.38, 2907.40,
2911.81, 2919.13, 2926.68, 2934.28, 2942.32, 2949.54,
2957.41, 2965.41, 2972.83, 2981.06, 2988.60, 2995.65,
3003.99, 3011.57, 3019.54, 3026.40, 3034.55, 3042.19,
3049.36, 3057.05, 3065.20, 3073.39, 3080.51, 3088.13,
3095.60, 3103.73, 3111.71, 3119.00, 3126.78, 3134.04,
3142.03, 3149.72, 3157.54, 3164.92, 3172.59, 3180.81,
3188.39, 3196.34, 3203.53, 3211.62, 3218.57, 3226.61,

3233.99, 3241.95, 3249.30, 3256.98, 3264.92, 3272.59,
3280.48, 3288.17, 3296.06, 3304.04, 3311.65, 3319.11,
3326.79, 3334.23, 3342.18, 3349.67, 3357.49, 3365.28,
3373.29, 3380.72, 3388.37, 3395.47, 3403.55, 3410.83,
3418.53, 3426.17, 3434.08, 3441.87, 3449.49, 3457.56,
3465.50, 3472.70, 3480.77, 3487.79, 3495.90, 3503.05,
3511.21, 3518.48, 3526.99, 3533.82, 3542.44, 3549.82,
3557.75, 3564.62, 3573.15, 3580.72, 3588.43, 3595.37,
3603.75, 3611.28, 3618.98, 3626.87, 3633.89, 3642.24,
3649.97, 3657.57, 3665.40, 3672.51, 3680.11, 3688.19,
3695.42, 3703.91, 3711.61, 3718.46, 3719.81, 3720.63,
3722.13, 3723.34, 3724.16, 3725.61, 3726.34, 3727.52,
3728.53, 3729.32, 3730.81, 3731.07, 3732.03, 3733.23,
3734.60, 3735.56, 3736.34, 3738.18, 3738.69, 3740.04,
3741.04, 3742.12, 3742.93, 3744.54, 3745.32, 3745.96,
3747.68, 3748.41, 3748.93, 3750.60, 3751.20, 3752.50,
3753.08, 3754.82, 3755.37, 3757.07, 3757.63, 3758.75,
3759.36, 3761.33, 3761.61, 3762.53, 3763.68, 3764.75,
3766.00, 3767.40, 3767.88, 3768.83, 3769.87, 3771.47,
3772.83, 3773.36, 3774.29, 3775.96, 3785.06, 3795.54,
3805.50, 3814.80, 3825.26, 3834.89, 3844.46, 3854.30,
3864.75, 3874.48, 3883.91, 3894.11, 3904.03, 3914.02,
3923.94, 3933.34, 3944.02, 3952.98, 3963.45, 3973.08,
3983.39, 3992.76, 4002.38, 4012.62, 4022.13, 4032.15,
4042.75, 4052.30, 4062.40, 4072.44, 4081.94, 4091.99,
4101.65, 4111.01, 4121.84, 4131.41, 4140.78, 4150.87,
4161.20, 4170.46, 4180.31, 4190.20, 4200.83, 4210.37,
4220.61, 4230.12, 4240.03, 4249.34, 4252.71, 4256.06,
4259.53, 4262.36, 4265.56, 4269.59, 4272.22, 4275.43,
4279.04, 4282.18, 4285.69, 4289.01, 4291.54, 4295.50,
4298.01, 4301.58, 4304.96, 4308.52, 4311.02, 4314.11,
4317.87, 4321.06, 4324.53, 4327.48, 4331.07, 4333.79,
4337.30, 4340.16, 4343.93, 4346.86, 4350.05, 4353.42,
4356.54, 4360.24, 4362.85, 4366.47, 4369.43, 4373.31,
4375.80, 4379.17, 4382.53, 4386.17, 4389.00, 4392.79,
4395.49, 4398.47, 4402.26, 4405.04, 4408.25, 4409.99,
4410.81, 4411.38, 4413.04, 4413.48, 4414.82, 4415.70,
4417.12, 4418.03, 4419.43, 4420.04, 4421.87, 4422.80,
4424.05, 4424.94, 4425.78, 4426.72, 4427.61, 4429.20,
4429.82, 4431.44, 4432.29, 4433.88, 4434.62, 4435.47,
4436.17, 4437.80, 4438.83, 4439.89, 4440.58, 4441.78,
4443.61, 4444.08, 4445.04, 4446.35, 4447.51, 4448.57,
4449.50, 4450.34, 4451.70, 4452.68, 4454.26, 4454.97,
4455.86, 4457.36, 4458.05, 4459.54, 4460.46, 4461.58,
4462.12, 4463.53, 4465.15, 4466.10, 4466.56, 4467.73,
4469.49, 4469.67, 4471.33, 4471.84, 4473.22, 4473.96,
4476.00, 4481.29, 4486.74, 4492.11, 4497.10, 4502.92,

```
4508.08, 4513.60, 4519.36, 4524.96, 4529.44, 4535.23,
4540.45, 4546.07, 4551.65, 4557.40, 4562.23, 4568.36,
4573.07, 4578.47, 4584.39, 4589.26, 4595.31, 4600.51,
4606.02, 4611.83, 4617.28, 4622.62, 4627.80, 4633.16,
4638.64, 4644.34, 4649.62, 4655.43, 4660.37, 4665.88,
4671.41, 4677.02, 4681.91, 4687.44, 4692.84, 4698.60,
4704.10, 4709.74, 4714.68, 4720.39, 4726.03, 4731.05,
4737.11, 4742.39, 4747.30, 4753.01, 4758.30, 4764.03,
4769.07, 4774.98, 4779.72, 4785.87, 4791.39, 4796.45,
4801.89, 4807.45, 4813.04, 4818.69, 4824.15, 4829.01,
4834.69, 4839.87, 4845.12, 4851.11, 4856.55, 4861.97,
4867.00, 4873.10, 4878.43, 4883.96, 4888.76, 4894.83,
4900.30, 4905.46, 4911.11, 4916.08, 4926.29, 4936.23,
4945.94, 4955.33, 4965.23, 4975.25, 4985.16, 4994.42,
5004.63, 5014.66, 5024.13, 5033.26, 5043.88, 5053.51,
5063.34, 5073.44, 5082.78, 5092.37, 5102.22, 5112.34,
5122.17, 5131.58, 5141.94, 5151.84, 5161.61, 5170.55,
5180.71, 5190.76, 5200.39, 5209.81, 5220.09, 5229.57,
5239.28, 5249.24, 5258.74, 5269.18, 5278.93, 5288.32]
ymax = 0.0010

# First & second figure: Knuth bins & Bayesian Blocks
fig = plt.figure(figsize=(10, 4))
fig.subplots_adjust(left=0.1, right=0.95, bottom=0.15)

for bins, title, subplot in zip(['knuth', 'blocks'],
["Knuth's Rule", 'Bayesian Blocks'], [121, 122]):

    ax = fig.add_subplot(subplot)

    # Plot a standard histogram in the background, with
alpha transparency
    hist(data_set, bins=25, histtype='stepfilled',
alpha=0.1, normed=True,
    label='Raw Data: Binned Counts')

    # Plot an adaptive-width histogram on top
    hist(data_set, bins=bins, ax=ax, color='black',
histtype='step', normed=True, label=title)

    # Control limits
    ax.legend(prop=dict(size=12))
    plt.ylim(0,ymax) # Change this value based on the graph
    plt.draw() # Draw the figure so the positon of the
legend can be found
    box = ax.get_position()
```

```
    ax.set_position([box.x0, box.y0 + box.height * 0.1,
box.width, box.height * 0.9])

    # Put a legend below current axis
    ax.legend(loc='upper center', bbox_to_anchor=(0.5, -
0.05), fancybox=True, shadow=True)
# Collect and display some metrics
bblocks = bayesian_blocks(data_set)
kbw = knuth_bin_width(data_set)
print("BB Change Points of Data Set: ")
print(bblocks)
print("Knuth Bin Width: ")
print(kbw)

# Update the final plot
plt.show()
```

APPENDIX B

Scargle's Algorithm for BATSE Gamma Ray Data MATLAB Code

```
% For data modes 1 and 2:
% nn_vec is the array of cell populations.
% Preliminary computation:
block_length=tt_stop-[tt_start 0.5*(tt(2:end)+tt(1:end-1))'
tt_stop];
%------------------------------------------------------------
% Start with first data cell; add one cell at each
iteration
%------------------------------------------------------------
best = [];
last = [];
for R = 1:num_points
% Compute fit_vec : fitness of putative last block (end at
R)
if data_mode == 3 % Measurements, normal errors
sum_x_1 = cumsum( cell_data( R:-1:1, 1 ) )'; %sum(x/sig^2)
sum_x_0 = cumsum( cell_data( R:-1:1, 2 ) )'; %sum(1/sig^2)
fit_vec=((sum_x_1(R:-1:1) ) .^ 2 ) ./( 4*sum_x_0(R:-1:1));
else
arg_log = block_length(1:R) - block_length(R+1);
arg_log( find( arg_log <= 0 ) ) = Inf;
nn_cum_vec = cumsum( nn_vec(R:-1:1) );
nn_cum_vec = nn_cum_vec(R:-1:1);
fit_vec = nn_cum_vec .* ( log( nn_cum_vec ) - log( arg_log
) );
end
[ best(R), last(R)] = max( [ 0 best ] + fit_vec - ncp_prior
);
end
%------------------------------------------------------------
% Now find changepoints by iteratively peeling off the last
block
%------------------------------------------------------------
index = last( num_points );
change_points = [];
while index > 1
change_points = [ index change_points ];
```

APPENDIX C

Performance Comparison MATLAB Code

```
clear all; theData =
{'1000','650','500','400','200','100'};

% Load all data sets and optimized histograms
theTTE = cell(size(theData));
theKNF = cell(size(theData));
theBBF = cell(size(theData));
for d = 1:length(theData)
  theTTE{d} = load(['TT_' theData{d} '.txt']); % N-by-1
vector of sample times
  theKNF{d} = load(['KN_' theData{d} '.txt']); % 3-by-M
matrix, row 1 being unnormalized heights, row 2 being bin
edges and row 3 being normalized heights
  theBBF{d} = load(['BB_' theData{d} '.txt']);

  % Count number of samples per bin of each fit and convert
(unnormalized) histogram to inter-arrival-time estimates
  mKN = nan(1,size(theKNF{d},2)-1);
  for k = 1:length(mKN), mKN(k) =
sum(theTTE{d}>=theKNF{d}(2,k) &
theTTE{d}<theKNF{d}(2,k+1)+eps); end
  xKN = diff(theKNF{d}(2,:))./theKNF{d}(1,1:end-1);
  mBB = nan(1,size(theBBF{d},2)-1);
  for k = 1:length(mBB), mBB(k) =
sum(theTTE{d}>=theBBF{d}(2,k) &
theTTE{d}<theBBF{d}(2,k+1)+eps); end
  xBB = diff(theBBF{d}(2,:))./theBBF{d}(1,1:end-1);
  % Compute min, mean and max of the inter-arrival time
samples
  xMin = min(diff(theTTE{d})); xMax = max(diff(theTTE{d}));
xAve = mean(diff(theTTE{d}));
  % Perform anomaly detection analysis
  p = linspace(0,1,10001); F = nan(3,length(p));
  for i = 1:length(p)
    xUB = xAve + (1-p(i))*(xMax-xAve); xLB = xAve - (1-
p(i))*(xAve-xMin);
    F(1,i) = sum(mKN(xKN<xLB | xKN>xUB));
    F(2,i) = sum(mBB(xBB<xLB | xBB>xUB));
    F(3,i) = sum(diff(theTTE{d})<xLB |
diff(theTTE{d})>xUB);
  end
```

```
  F = F/(length(theTTE{d})-1);

  figure(d);
  % Show the (unnormalized) histogram fits to the time-
tagged-event data
  subplot(2,3,[1 2]);
  [x1,y1] = stairs([theKNF{d}(2,1) theKNF{d}(2,:)],[0
theKNF{d}(1,:)]);
  [x2,y2] = stairs([theBBF{d}(2,1) theBBF{d}(2,:)],[0
theBBF{d}(1,:)]);
  plot(x1,y1,'-k.',x2,y2,'-r.','LineWidth',2);
  xlabel('time (sec)'); ylabel('number of arrivals (per
bin)');
  legend({'KN-fit','BB-fit'},'Location','NorthWest');
  % Also show per-fit bin edges and the actual arrival
times
  hold on;
  plot(theTTE{d},zeros(size(theTTE{d})),'b.');
  for k = 1:size(theKNF{d},2), plot(theKNF{d}(2,k)*[1
1],ylim,'k:'); end
  for k = 1:size(theBBF{d},2), plot(theBBF{d}(2,k)*[1
1],ylim,'r:'); end
  hold off;
  title(['Two Histogram-Based Fits to the Length-'
theData{d} ' Time-Tagged-Event Data']);

  % Show the per-fit estimates against the actual inter-
arrival time data
  subplot(2,3,[4 5]);
  plot(theTTE{d}(1:end-1),diff(theTTE{d}),'b.');
  xlabel('time (sec)'); ylabel('inter-arrival times
(sec)');
  % Also show per-fit bin edges and the actual inter-
arrival times
  hold on;
  [x1,y1] = stairs([theKNF{d}(2,1) theKNF{d}(2,:)],[0
(theKNF{d}(1,1:end-1)./diff(theKNF{d}(2,:))).^-1 0]);
  [x2,y2] = stairs([theBBF{d}(2,1) theBBF{d}(2,:)],[0
(theBBF{d}(1,1:end-1)./diff(theBBF{d}(2,:))).^-1 0]);
  plot(x1,y1,'-k.',x2,y2,'-r.','LineWidth',2);
  legend('raw','KN-fit','BB-fit','Location','SouthWest');
  for k = 1:size(theKNF{d},2), plot(theKNF{d}(2,k)*[1
1],ylim,'k:'); end
  for k = 1:size(theBBF{d},2), plot(theBBF{d}(2,k)*[1
1],ylim,'r:'); end
  hold off;
```

```
  title(['Associated Piecewise-Constant Fits to the Inter-
Arrival-Time Samples']);

  % Show the anomaly detection results
  subplot(2,3,[3 6]); plot(p,F(3,:),'-b',p,F(1,:),'-
k',p,F(2,:),'-r','LineWidth',2);
  xlabel('detector sensitivity'); ylabel('fraction
anomalous');
  legend({'raw','KN-fit','BB-fit'},'Location','NorthWest');
  title('Performance Comparison');
end
```

VITA

Alaa (Al) Alkadi is an experienced senior research and development embedded systems and software engineer with an ABET accredited electrical engineering B.S. degree from the University of North Florida. Al's expertise includes: designing, implementing, and extending a backbone framework for advanced security applications by integrating various technologies. These technologies include web and mobile GUI applications; video capture, analysis, archiving and streaming; PLC and other automation controllers; IP based sensors and output devices; custom drivers, third party SDKs, and open source projects. Al specializes in cross platform software development in Qt (Windows and Linux), C#, C/C++, Embedded C and Assembly, Python, XSL/XSLT, Java and VB.NET.