2019-09

# BLOCKCHAIN NETWORK BEHAVIOR-BASED ANOMALY DETECTION

Pendino, Stephanie R.

Monterey, CA; Naval Postgraduate School


# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**BLOCKCHAIN NETWORK BEHAVIOR-BASED ANOMALY DETECTION**

by

Stephanie R. Pendino

September 2019

Co-Advisors: John C. McEachen
Murali Tummala

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | | *Form Approved OMB No. 0704-0188* |
|---|---|---|---|



| **1. AGENCY USE ONLY** *(Leave blank)* | **2. REPORT DATE** September 2019 | **3. REPORT TYPE AND DATES COVERED** Master's thesis |
|---|---|---|

| | |
|---|---|
| **4. TITLE AND SUBTITLE** BLOCKCHAIN NETWORK BEHAVIOR-BASED ANOMALY DETECTION | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)** Stephanie R. Pendino | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Naval Postgraduate School Monterey, CA 93943-5000 | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)** N/A | **10. SPONSORING / MONITORING AGENCY REPORT NUMBER** |
| **11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. | |
| **12a. DISTRIBUTION / AVAILABILITY STATEMENT** Approved for public release. Distribution is unlimited. | **12b. DISTRIBUTION CODE** A |

**13. ABSTRACT (maximum 200 words)**

Blockchain technology has the potential to improve the areas of additive manufacturing, supply chain management, and many others within the Navy. An anomaly detection scheme that characterizes blockchain parameters as normal or anomalous using statistical analysis and hierarchical clustering methods was developed in this thesis. The histograms, probability distributions, and boxplots of the data were used to estimate thresholds for outliers that may indicate attacks. The thresholds obtained from dendrograms were used to form clusters and sub-clusters based on the hierarchical data structure; data point indices that do not fall within the threshold are considered anomalous and not included in the clusters. The anomaly detection scheme was implemented in the MATLAB programming environment and validated by successful anomaly detection corresponding to an attack on the public Ethereum blockchain network and in an experimental doorknob-rattling attack on a local blockchain research network. Hierarchical clustering proved to be a more powerful anomaly detection method than statistical analysis methods.

| **14. SUBJECT TERMS** blockchain, k-means, hierarchical clustering, anomaly detection | **15. NUMBER OF PAGES** 71 |
|---|---|
| | **16. PRICE CODE** |

| **17. SECURITY CLASSIFICATION OF REPORT** | **18. SECURITY CLASSIFICATION OF THIS PAGE** | **19. SECURITY CLASSIFICATION OF ABSTRACT** | **20. LIMITATION OF ABSTRACT** |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UU |

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release. Distribution is unlimited.**

**BLOCKCHAIN NETWORK BEHAVIOR-BASED ANOMALY DETECTION**

Stephanie R. Pendino
Lieutenant Commander, United States Navy
BS, Texas State University - San Marcos, 2005

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL**
**September 2019**

Approved by: John C. McEachen
Co-Advisor

Murali Tummala
Co-Advisor

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Blockchain technology has the potential to improve the areas of additive manufacturing, supply chain management, and many others within the Navy. An anomaly detection scheme that characterizes blockchain parameters as normal or anomalous using statistical analysis and hierarchical clustering methods was developed in this thesis. The histograms, probability distributions, and boxplots of the data were used to estimate thresholds for outliers that may indicate attacks. The thresholds obtained from dendrograms were used to form clusters and sub-clusters based on the hierarchical data structure; data point indices that do not fall within the threshold are considered anomalous and not included in the clusters. The anomaly detection scheme was implemented in the MATLAB programming environment and validated by successful anomaly detection corresponding to an attack on the public Ethereum blockchain network and in an experimental doorknob-rattling attack on a local blockchain research network. Hierarchical clustering proved to be a more powerful anomaly detection method than statistical analysis methods.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF FIGURES

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

Blockchain technology is useful for a number of applications, and it is reasonable to assume that it will be implemented in U. S. Navy and other Department of Defense information networks in the near future. There is a cooperative research and development agreement between the Naval Air System Fleet Readiness Center Southwest and Indiana Technology and Manufacturing Companies to demonstrate a blockchain proof-of-concept in the focus area of supply tracking [1]. The Naval Innovation Advisory Council (NIAC) has determined that blockchain technology has the ability to improve many secure data exchange processes [2].

The fundamental concepts that make blockchain so appealing are security and trust. The most popular applications of blockchains have been in the financial domain as cryptocurrency, such as Bitcoin and Ethereum. These blockchains have been subjected to attacks through vulnerabilities identified in the software implementations and smart contracts [3]. With the integration of this new technology in our networks comes the responsibility of understanding what indications of attack may look like so we can better defend our networks.

The research detailed in this thesis describes methods that may detect some, but not all, anomalies indicative of an attack against a blockchain network. It is an important step toward defining what normal blockchain network behavior is, although there remains much to be investigated.

## A. THESIS OBJECTIVE

The objective of this thesis is to develop an anomaly detection scheme to characterize the behavior of blockchain-based systems using both statistical analysis and unsupervised machine learning methods. Ideally, the network data associated with the addition of blockchain technology for database storage can complement existing network intrusion detection systems.

The anomaly detection scheme proposed in this thesis determines normal network behaviors by analyzing the number of blockchain transactions and average gas usage, or

computational power. The scheme is implemented in MATLAB and applied to both the data of the public Ethereum blockchain and to data collected in a local blockchain research network to demonstrate anomaly detection based on known network attacks.

## B. RELATED WORK

The most well-known application of blockchain technology is in the form of cryptographic currency. Bitcoin, the most widely used cryptocurrency, uses a peer-to-peer network with no central authority but has security vulnerabilities primarily in the proof-of-work consensus protocol [4]. Rahouti et al. describe blockchain security threats and proposed solutions in [4]. T. Pham and S. Lee successfully used k-means clustering, Mahalanobis distance, and an unsupervised support vector machine to determine Bitcoin network anomalous behavior in [5]. All of these methods study the transaction structure in the network.

Ethereum is another public blockchain that uses smart contracts as permanent, irreversible records of transactions [4]. Bogner analyzed the Ethereum network and was able to report some anomalies by using undisclosed machine learning methods on a small subset of the network data, approximately five thousand of the over seven million blocks [6].

Carter et al. present a method for anomaly detection on large-scale networks without using temporal modeling but rather using nodes exhibiting similar behaviors [7]. By measuring dissimilarity and hierarchical clustering, they were able to identify, characterize, and classify groups of anomalous behavior of incoming and outgoing IP traffic [7]. We used these concepts to gain understanding of anomalous behavior in the context of the Ethereum blockchain.

Combining the use of statistical methods, machine learning methods, and hierarchical clustering, we were able to detect an attack associated with the Ethereum blockchain network and associate trends over the entire data set. However, network behavior evolves over time, making it necessary to periodically adjust anomaly detection thresholds. Additionally, there were known attacks on the Ethereum network that were not detected by our methods, meaning that the approach we used can only detect anomalies if

the attack required noticeable transaction or gas usage increases. We also resolve that hierarchical cluster analysis is more powerful than statistical analysis methods because it provides a better understanding of the details of the underlying data structure.

## C. ORGANIZATION

Five chapters and two appendices are contained in this thesis. The background information on several topics including blockchain technology and its application to naval operations, statistical analysis methods, and clustering methods is covered in Chapter II. The anomaly detection scheme and how it is used to detect outliers are described in Chapter III. The scheme described in Chapter III is applied to the public Ethereum blockchain network and to an experimental local blockchain research network in Chapter IV. A summary of significant results and recommendations for future work is included in Chapter V. The MATLAB programming code used to implement the statistical analysis method and the clustering analysis method is contained in Appendix A and Appendix B, respectively.

THIS PAGE INTENTIONALLY LEFT BLANK

# II. BACKGROUND

The dominant concepts from blockchain application and anomaly detection leveraged in this thesis are introduced in this chapter. A discussion of blockchain fundamental concepts and U. S. Navy potential blockchain applications is offered first. Statistical techniques for analyzing network behavior are introduced. The use of k-means and hierarchical cluster analysis for anomaly detection are then presented.

## A. BLOCKCHAIN TECHNOLOGY AND APPLICATIONS TO NAVAL OPERATIONS

In 2017, the NIAC set out to experimentally determine if blockchain technology could really be used as secure data storage, selecting additive manufacturing as its focus [2]. Blockchain technology has the potential to improve the areas of additive manufacturing, supply chain management, and many others [2]. As incorporation of this new technology begins to appear inevitable, we must consider the implications these systems will have on existing network intrusion detection systems.

The 2019 publication of the Cybersecurity Readiness Review by the Secretary of the Navy highlights the importance of cybersecurity and that a logical approach must be taken to anticipate and detect network attacks to maintain robust, continuous systems [8]. We must understand anomaly detection in this new context prior to implementation of blockchains in our networks.

Blockchain technology, introduced in 2008, is an encryption and authentication technology for peer-to-peer networks, relying on a hash-based proof-of-work consensus protocol to overcome the need for a central authority [9]. Blockchain is built on a very simple concept. Each block, which is a container for data, is chained to the block before it by containing the hash of the previous block [9]. One metric of interest that helps to characterize a blockchain is a transaction, or signed and timestamped data package containing messages to be transferred between the participants in the blockchain. Transactions are broadcast to all other nodes in the blockchain and accepted based on

validation of an accompanying signature and the existing user account balance as recorded in the blockchain [9].

Another metric of interest in a blockchain network is average gas usage. Transactions contain a gas value which signifies the maximum computational cost a participant is willing to pay for the codification of the transaction in a block [10]. The average gas usage is then taken across the blocks created in a time period of interest and represents how much participants are willing to pay to process transactions; when this value spikes, we can reasonably assume something malicious is occurring.

In proof-of-work blockchains, miners race to find nonce values that, when grouped with the hash of the previous block and a set of transactions waiting to be added to the chain, will produce a hash value meeting some agreed upon and difficult-to-achieve quality [9], such as a required number of leading zeros. Completed blocks are broadcast to all participants for inclusion in their local copies of the chain. In this way, blockchains protect data immutably, because subsequent changes to a block that is already encapsulated in the chain must be accompanied with simultaneous changes in all of the follow-on blocks [9]. The longest version of the chain in circulation is accepted as the correct chain; each of the distributed participants replaces their version if they receive one that is longer.

Ethereum is one of the most popular blockchain networks because of its use of smart contracts, or computer programs, which carry out an agreement when executed [11]. Although known for its security and trust, attacks have occurred against the public Ethereum blockchain. An attacker stole approximately $60 million in the infamous decentralized autonomous organization (DAO) attack on June 18, 2016 [3]. The DAO attack successfully took advantage of vulnerabilities in the code of the smart contracts. The attacker was able to transfer ether, Ethereum cryptocurrency, out of the DAO account and continue taking withdrawals before updating the balance [3]. The Ethereum blockchain recovered via a hard fork, or a new version of the blockchain, starting from just prior to the attack in order to prevent the loss of ether.

On July 19, 2017, a digital wallet contract permissions vulnerability was exploited, resulting in the theft of 150,000 ether, more than $33 million [12]. On November 7, 2017,

libraries were accidently deleted, freezing 500,000 ether, or $150 million [13]. In this thesis, we attempt to investigate what reflections of the above events we can observe using statistical and clustering analysis.

## B. STATISTICAL ANALYSIS

To understand how anomaly detection could be implemented in blockchain technology, this thesis began with a statistical analysis of blockchain technology data. The principal techniques we used for this preliminary analysis were histograms, probability density functions, and boxplots.

Histograms use bins to portion the data points that fall within a certain value range in that bin, providing a means to assess average and variability [14]. The shape of the histogram can be characterized by probability density functions. This thesis applies three probability density functions: kernel, Gaussian, and generalized extreme value.

A kernel is a smoothing function and a nonparametric representation of the probability density; it is used when other distributions cannot properly be fitted to the data or when attempting to avoid making assumptions about the data [15]. The general form of the estimated kernel probability density function is given by

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \tag{1}$$

where $K$ is the kernel, $X_i$ are the real observations, $n$ is the number of observations, and $h$ is the bandwidth [16]. Only a Gaussian kernel was used in this thesis.

The Gaussian distribution is symmetric about its mean and approaches zero a few standard deviations from the mean. The general form of the Gaussian probability density function is given by:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-m)^2}{2\sigma^2}} \tag{2}$$

where $m$ is the mean and $\sigma$ is the standard deviation [17].

The generalized extreme value distribution is used for modeling extremes of natural phenomena and the cumulative distribution function is given by:

$$F(x) = \exp[-\{1-(^{\kappa(x-x_o)}/_{\alpha})\}^{1/\kappa}] \tag{3}$$

where $\alpha$ is the scale parameter, $x_o$ is the location parameter, and $\kappa$ is the shape parameter [18], [19].

A boxplot is another way of visually assessing the median, or the value that falls in the middle of the dataset, and is given in Figure 1. The boxplot splits the data into four equal parts, or quartiles. The upper ($Q_3$) and lower ($Q_1$) quartile, representing 75% and 25% of the data, respectively, form the edges of the box. The distance between the upper and lower quartiles, $d_{IQR}$, is the interquartile range (*IQR*) and contains 50% of the data. The *whiskers* represent the minimum and maximum values in the data set within $1.5d_{IQR}$ [20]. Values that exceed $1.5d_{IQR}$ above the upper quartile or below the lower quartile are considered minor outliers; values that exceed $3d_{IQR}$ above the upper quartile or below the lower quartile are considered major outliers [20]. These four limits form the *fences* of the boxplot and are illustrated in Figure 1. Outliers are defined as values beyond the whiskers and marked with a red plus sign in Figure 1. The algebraic representations for the limits of the upper inner fence ($L_{UIF}$), lower inner fence ($L_{LIF}$), upper outer fence ($L_{UOF}$), and lower outer fence ($L_{LOF}$) determinations are as follows [20]:

$$L_{UIF} = Q_3 + 1.5d_{IQR} \tag{4}$$

$$L_{LIF} = Q_1 - 1.5d_{IQR} \tag{5}$$

$$L_{UOF} = Q_3 + 3d_{IQR} \tag{6}$$

$$L_{LOF} = Q_1 - 3d_{IQR} \tag{7}$$

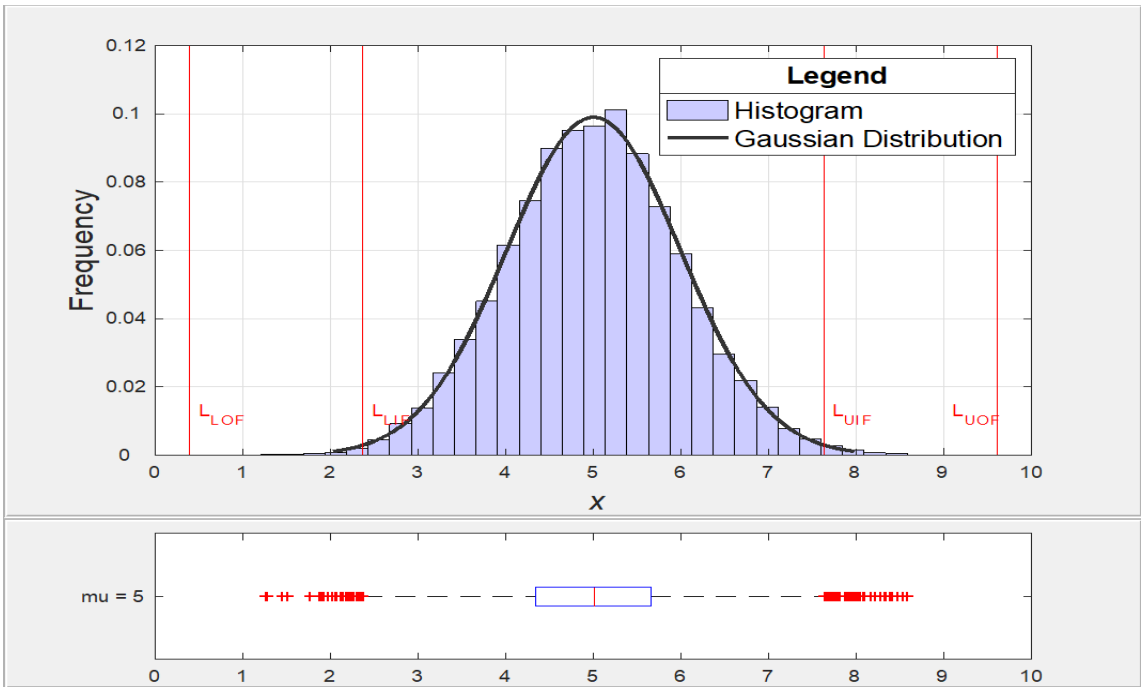


Figure 1. Histogram, distribution fit, and boxplot example.

The combination of these tools can be used for basic anomaly detection by establishing immediate, clearly defined thresholds. We will explore the machine learning methods that might help us characterize large data sets in the next section.

## C. K-MEANS AND HIERARCHICAL CLUSTERING ANALYSIS

The primary machine learning methods we examined for anomaly detection in blockchain technology were k-means and hierarchical clustering.

### 1. K-means Clustering Analysis

The k-means clustering algorithm was first introduced in 1956 by Steinhaus [21] and is widely used today. The k-means method is explained as taking *k* groups, consisting of random initial subsets, and calculating their means. As new points are added, they will be assigned to the group with the closest mean, and the group mean is adjusted each time a new data point is added. These groups of means, or clusters, are known as k-means [22].

The k-means algorithm forms these clusters by taking a set of data points $x = \{x_1, x_2 \ldots x_n\}$ and partitioning them into a set $c$ of $k$ clusters $c = \{c_1, c_2 \ldots c_k\}$. The goal is to minimize the objective function

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \| x_i - \mu_k \|^2 \tag{8}$$

where $\mu_k$ is the centroid of cluster $c_k$ [21]. Centroids are centers for the spherical shapes of the clusters [22]. The objective function can only be minimized for a fixed number of clusters because $J(C) = 0$ when $K = n$ [21]. Once the data is divided into $k$ clusters, subsequent iterations update cluster assignments and centroids until they converge [21].

K-means usually specifies the Euclidean distance as the metric between data points and the centroid; this will be the process used throughout this thesis. One of the difficulties in using the k-means algorithm is determining the correct value for $k$, as there are many known methods [21]. We will discuss hierarchical clustering for use in our analysis in the next subsection.

### 2. Hierarchical Cluster Analysis

Hierarchical cluster analysis is one way to determine the optimal number of clusters best suited for anomaly detection for a given data set. Binary hierarchical clustering begins by starting with one cluster and then splitting it into two clusters based on dissimilarities forming a tree structure, or dendrogram, until there are no similarities remaining [23]. A dendrogram is a hierarchical representation of the similarities between groups of data points [7].

Similarities of the dendrogram are determined by the Euclidean distance, or linkage, between data points [7]. There are three primary choices for linkage, all illustrated in Figure 2: single linkage, complete linkage, and average linkage. Single linkage is the nearest neighbor technique in which the distance between clusters A and B, $d_{AB}^{s}$, is defined as the minimum distance between a data point from cluster A and a data point from cluster B as given by [7], [24]

$$d_{AB}^{s} = \min\{d(a,b) : a \in A, b \in B\}. \tag{9}$$

Complete linkage relates the distance between cluster A and cluster B, $d_{AB}^{c}$, by the maximum distance between a data point from cluster A and a data point from cluster B as given by [7], [24]

$$d_{AB}^{c} = \max\{d(a,b) : a \in A, b \in B\}. \tag{10}$$

Average linkage relates the distance between cluster A and cluster B, $d_{AB}^{a}$, by the average distance as given by [24]

$$d_{AB}^{a} = \frac{\sum_{i}\sum_{j} d_{ij}}{N_A N_B} \tag{11}$$

where $d_{ij}$ is the pairwise distance between all data points from cluster A and all data points from cluster B, $N_A$ is the number of data points in cluster A, and $N_B$ is the number of data points in cluster B.

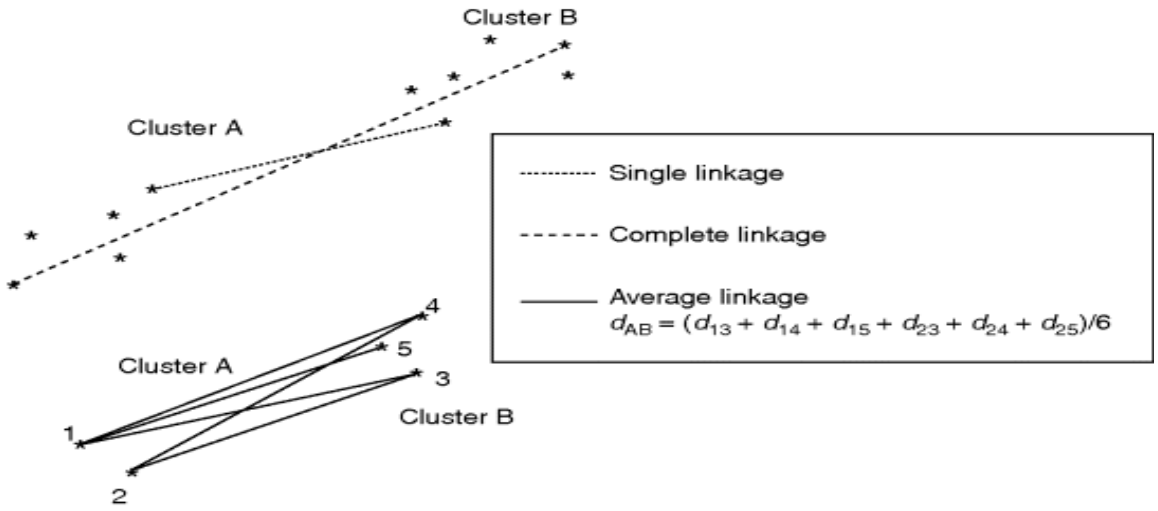


Figure 2. Inter-cluster distance measures. Source: [24].

Hierarchical clustering creates a hierarchical structure regardless of whether that structure actually exists. The cophenetic correlation, $\rho$, indicates how well the dendrogram preserves the pairwise distance between original data points and can be calculated for each linkage method [24]. The closer the cophenetic correlation value is to 1.0, the more similar

the hierarchical representation is to the original data. The cophenetic correlation is one way to determine if the distortion created in the hierarchical structure is acceptable [24].

The dendrogram can be qualitatively assessed to determine the threshold, $\theta$, for the division of clusters [7]. Groups that merge higher in the dendrogram have more similarities and are prime candidates for threshold determination. For example, a threshold of $\theta = 0.75$ forms two clusters, indicated in blue and green, in the dendrogram of Figure 3(a). The resulting clusters are shown in Figure 3(b). The data point indices of the dendrogram correspond to the individual data points that create the two clusters as illustrated in Figure 3. As behavior evolves over time, thresholds will likely need to be reevaluated.

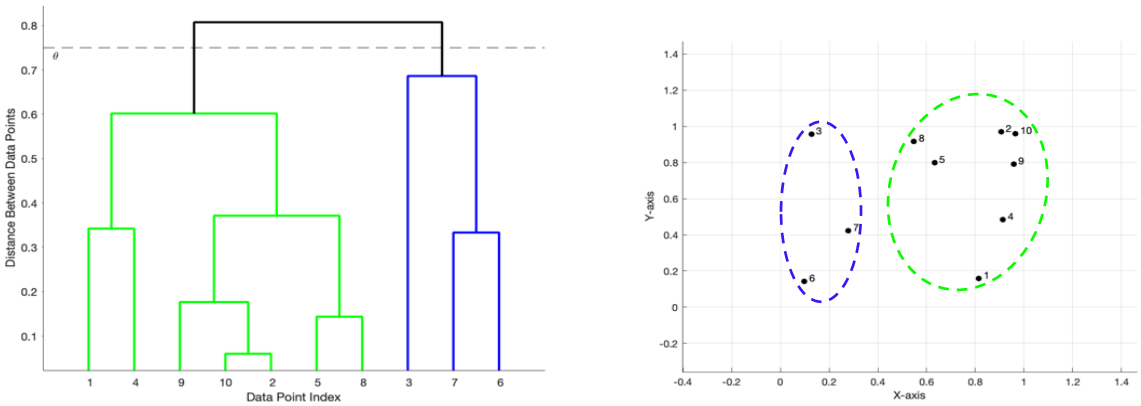


Figure 3. Dendrogram (a) and clusters (b). Adapted from [25].

In this chapter, an overview of blockchain technology was presented to understand blockchain applications and the need for an anomaly detection scheme. Statistical analysis provides a possible means for anomaly detection once blockchain technology is integrated into networks. Relevant discussions for k-means clustering and hierarchical clustering, which are considered machine learning techniques, for anomaly detection were also provided.

# III. ANOMALY DETECTION SCHEME

A background of statistical techniques, unsupervised k-means clustering, and hierarchical clustering approaches were presented in Chapter II. Specifically, the ways and means to determine anomaly detection thresholds were discussed. Here we will explore application of these methods as a way to detect malicious behavior. This chapter is broken into two parts: statistical analysis and clustering analysis.

## A. STATISTICAL ANALYSIS METHOD

We use analysis of the histograms, probability distributions, and boxplots as given in the schematic diagram shown in Figure 4 to estimate thresholds for outliers that may indicate attacks. First, the input data is normalized and a histogram created to provide an indication of a statistical distribution that describes the behavior. After the distribution fit is applied to the histogram, a boxplot is created to determine the median and visualize outliers. The outlier detection limits, $L_{UIF}$ and $L_{UOF}$, are calculated using Equations (4) and (6) and applied to the histogram. $L_{UIF}$ and $L_{UOF}$ can be used to calculate the probabilities of detecting values that exceed those thresholds as follows:

$$p_{UIF} = \int_{L_{UIF}}^{\infty} f(x)dx \tag{12}$$

$$p_{UOF} = \int_{L_{UOF}}^{\infty} f(x)dx \tag{13}$$

where $p_{UIF}$ is the probability range indicating that the behavior is changing and requires evaluation to determine if an attack has occurred, and $p_{UOF}$ is the probability range indicating an attack has occurred. Since realistically we do not know if an attack has occurred or not for a given data set, these probabilities are continuously calculated.

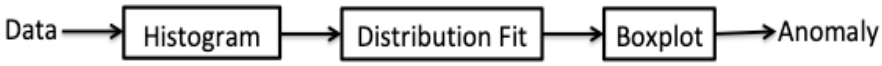


Figure 4. Statistical analysis method.

To illustrate the process, a sample data set was generated and normalized. The histogram for this data set is illustrated in Figure 5 and indicates that the data is bimodal. A kernel distribution was fit to the histogram since it is nonparametric. The kernel distribution follows the general form as given in Equation (1) with the bandwidth parameter specified in Table 1. There appear to be no anomalies in this data set as noted by the lack of outliers in the boxplot. The statistical parameters associated with the data are listed in Table 1. Since the calculated values for $L_{UIF}$ and $L_{UOF}$ exceed the bounds for normalized data, the probability of detecting data points beyond those limits is zero.

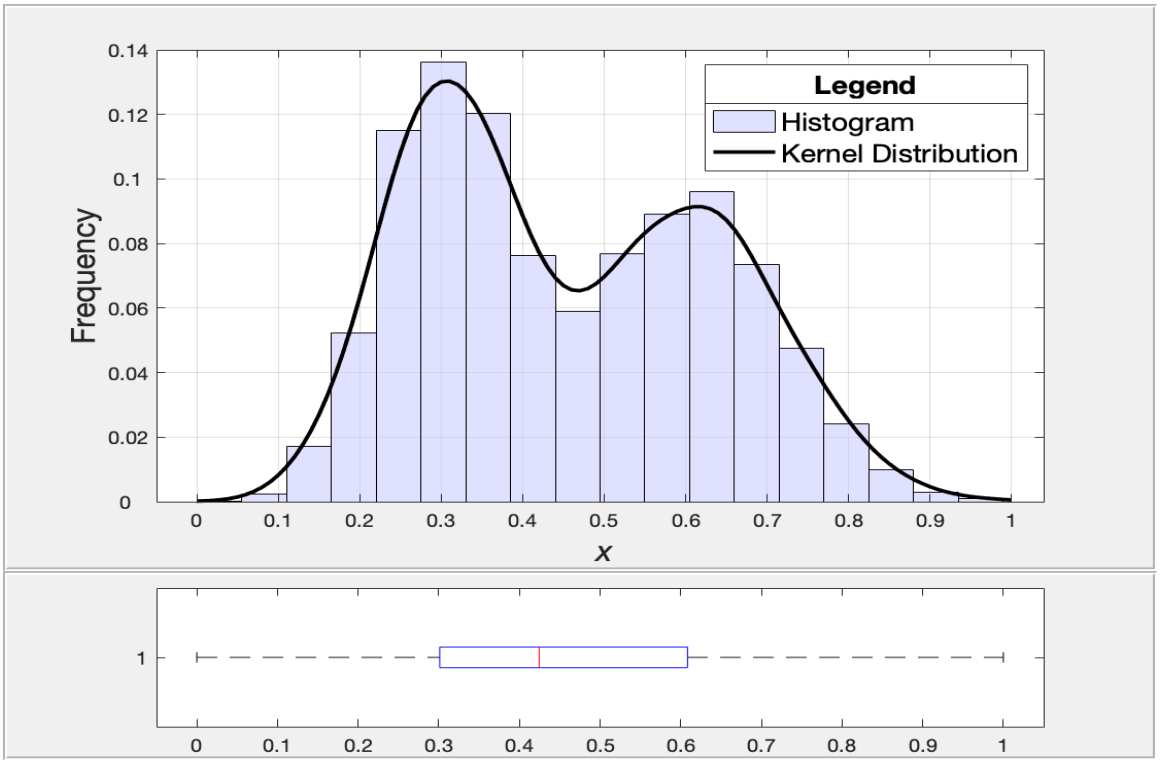


Figure 5. Histogram, distribution fit, and boxplot for sample data set.

Table 1. Parameters for the sample data set (see Figure 5).

| Parameter | Value |
|---|---|
| Bandwidth, $h$ | 0.2388 |
| Median | 0.4245 |
| Upper Quartile, $Q_3$ | 0.6084 |
| Lower Quartile, $Q_1$ | 0.3012 |
| Interquartile Range, $d_{IQR}$ | 0.3072 |
| Upper Inner Fence Limit, $L_{UIF}$ | 1.0692 |
| Upper Outer Fence Limit, $L_{UOF}$ | 1.5301 |
| $p_{UIF}$ | ~ 0 |
| $p_{UOF}$ | ~ 0 |

## B. CLUSTERING ANALYSIS METHOD

The process of anomaly detection using clustering methods is developed in this section and depicted in Figure 6. The same sample data set as in Section A that exhibited a bimodal behavior is input to the k-means algorithm. Then a dendrogram is created to determine the anomaly detection threshold based on the hierarchical structure of the data, which is used to create hierarchical clusters. Anomalies are data points that do not belong in the hierarchical clusters. This section is broken into two parts: cluster refinement and hierarchical clustering.

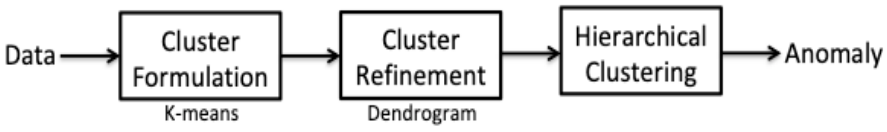


Figure 6. Clustering analysis method.

### 1. Cluster Refinement

For the k-means algorithm, the *k* value is manually selected to minimize the objective function in Equation (8). Although this method relies heavily upon iterative manual adjustment for selecting the acceptable *k* value, it serves as a means to visually analyze the possible data clustering. The sample data set and $k = 2$ were input into the k-means algorithm, resulting in two clusters separated midway between the centroids as given in Figure 7.

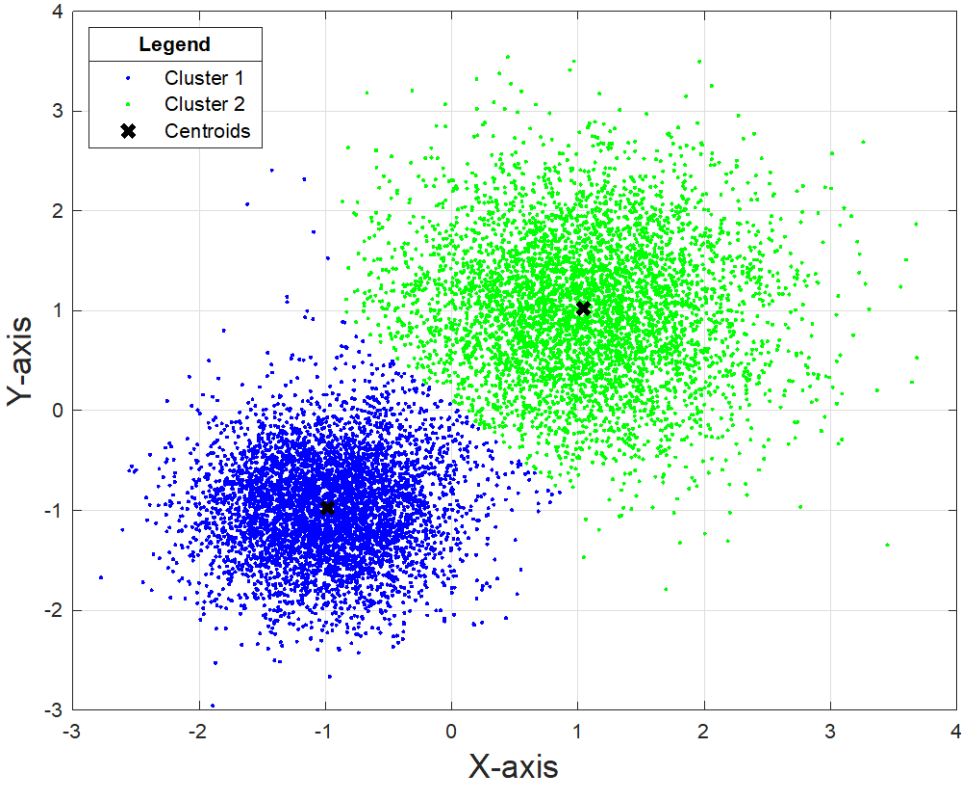


Figure 7. K-means clustering of sample data set, for $k = 2$.

A more efficient way to determine the value for *k* is by creating dendrograms and hierarchical clusters as discussed in Chapter II. Since the dendrogram is based on similarities between the data points, it is structured in a way that helps determine the best number of clusters for a given data set. Different linkage methods are explored to find the highest cophenetic correlation, $\rho$, or measure of how well the dendrogram preserves the pairwise distance between original data points [24]. Cophenetic correlation values for single, complete, and average linkages are provided in Table 2. Average linkage, $d_{AB}^{a}$, provided the highest cophenetic correlation not only for this data set, but also for the publicly available Ethereum data and for our local blockchain; therefore, it is used throughout this thesis.

Table 2. Cophenetic correlation values for sample data set.

| Linkage Method | Cophenetic Correlation |
|---|---|
| Single, $\rho_s$ | 0.4219 |
| Complete, $\rho_c$ | 0.7593 |
| Average, $\rho_a$ | 0.8217 |

Now that we have decided to use average linkage, $d_{AB}^{a}$, to structure the data, an appropriate threshold for hierarchical clustering needs to be determined. The dendrogram shown in Figure 8 is abbreviated because the size of the entire sample data set is too numerous to view, and a clear idea about the data structure can be obtained with only 20 data point indices. For example, selecting a threshold value $\theta_1 = 2.9$ yields two clusters, as noted by the dashed line labeled $\theta_1$. Selecting a threshold values of $\theta_2 = 2.3$ and $\theta_3 = 2.0$ would yield three and five clusters, respectively. Data point index 14 appears to be anomalous because it is separated by a large distance from any other data points. To separate data point index 14 from the majority of the data points, which form two clusters, the best threshold for this data set is $\theta_2$. This decision will result in two primary clusters (blue and green) and one anomaly cluster (red) as further explained in the next subsection.

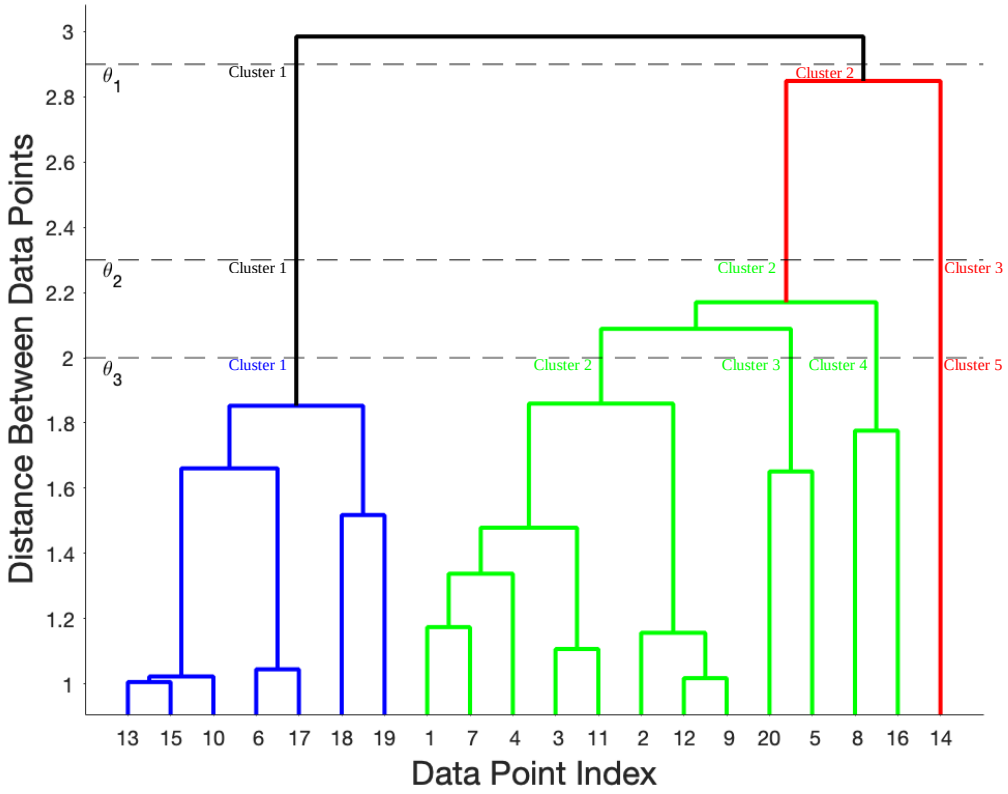


Figure 8. Dendrogram with $\theta_1$ = 2.9, $\theta_2$ = 2.3, and $\theta_3$ = 2.0.

## 2. Hierarchical Clustering

With the optimal number for *k* determined and a threshold identified, hierarchical clustering isolates data points that are dissimilar from the others automatically. The cluster tree created using Euclidean distance and average linkage is then used to create new clusters. If the dendrogram in Figure 8 were expanded to all 10,000 data points, we would see that it does branch into a set of three data points instead of one. These account for the anomalies illustrated in Figure 9. Comparing the clusters in Figures 9 and 7, there are minor differences, as noted in the red ellipse in Figure 9, between the hierarchical and k-means clustering. This will prove to be insignificant in network anomaly detection as discussed in Chapter IV. The percentage of data points, $\gamma_i$, that belong in each cluster can be determined by

$$\gamma_i = \frac{N_i}{N} \times 100 \tag{14}$$

where $N_i$ is the number of data points in the $i^{th}$ cluster and $N$ is the total number of data points in the data set. The results for this sample data set are listed in Table 3 and show that the clusters are approximately the same size with three anomalies as expected from the dendrogram.

Table 3. Cluster comparison for sample data set.

| Cluster | Percentage of Data Points, $\gamma_i$ |
|---|---|
| Cluster 1 | 50.39% |
| Cluster 2 | 49.58% |
| Anomaly | 0.03% |

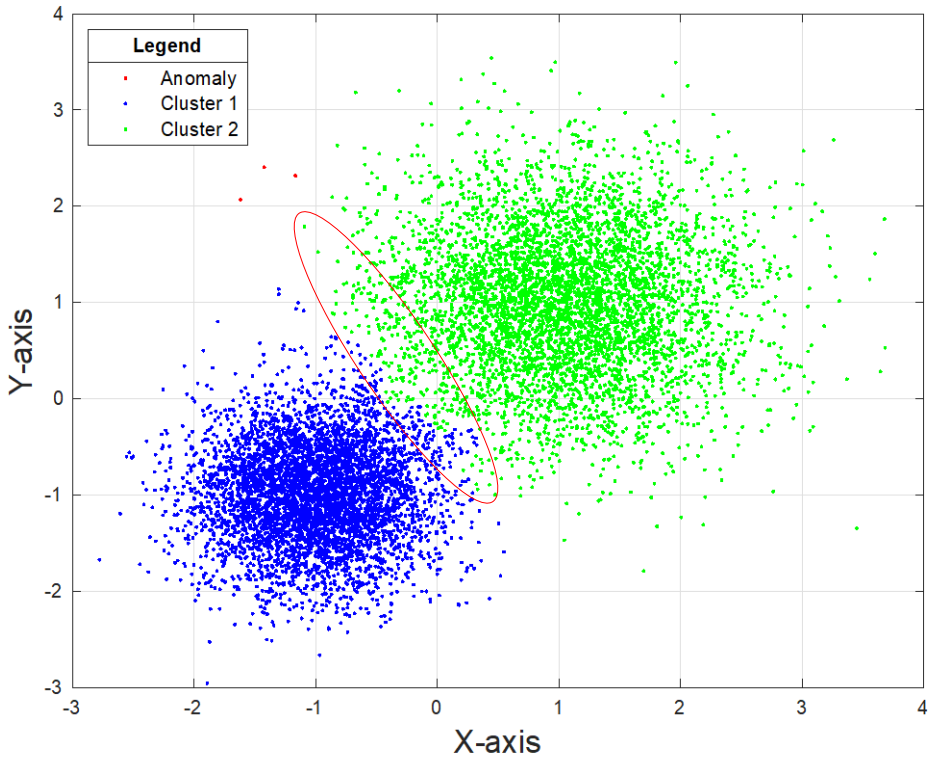


Figure 9. Hierarchical clustering, for $\theta = 2.3$ and $k = 2$.

### 3. Proposed Anomaly Detection Scheme

We propose combining both statistical and clustering methods in order to determine the possible anomaly detection thresholds as illustrated by the proposed anomaly detection scheme in Figure 10. The statistical analysis yielded no outliers as discussed in Section A. However, the hierarchical clustering analysis alerted to three anomalous data points that require further investigation as discussed in Section B. These data points could represent new normal behavior or may indicate malicious activity.

The benefit of the hierarchical clustering is that the threshold from the dendrogram creates clusters and sub-clusters; data points that exceed the threshold are considered anomalies and not included in the clusters. The sub-clusters assist in determining new normal behavior as the network evolves. Determination of anomaly detection thresholds is a challenging process. Care must be taken because a threshold too low will result in detection of an excessive number of anomalies and a threshold too high will result in no anomaly detection.

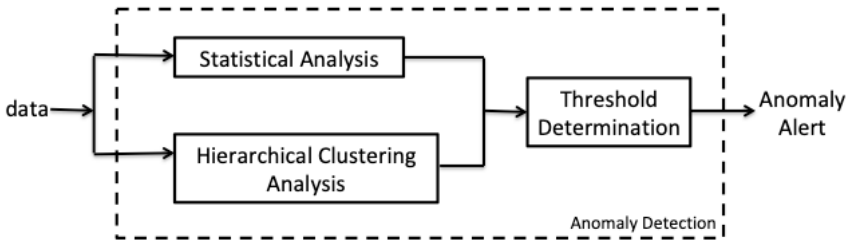


Figure 10. Anomaly detection scheme.

In this chapter, we proposed an anomaly detection scheme for use with blockchain-based systems. The statistical and clustering analysis methods were described in detail and demonstrated using a sample data set. The anomaly detection scheme is applied to the publicly available Ethereum blockchain and to an experimental local research blockchain in Chapter IV.

# IV. RESULTS

The methods described in Chapter III are implemented and verified in this chapter. We evaluated the performance of our proposed anomaly detection scheme using data from the publicly available Ethereum blockchain as well as data from an Ethereum-based blockchain built locally. Although there have been a limited number of known malicious network attacks of the public Ethereum network, data from these known attacks provide significant, historical blockchain behavior with at least some ground truth against which to measure the performance of our scheme. Although limited in volume and duration, the controlled attacks on our local network, independent of the social influences of cryptocurrency, provide successful anomaly detection results.

## A. PUBLIC ETHEREUM BLOCKCHAIN

One of the most popular blockchain networks, Ethereum, provides historical network measurements that are publicly available in [26], making it a good candidate for initial evaluation of anomaly detection techniques for blockchain-based systems. There are many measurable parameters for the Ethereum blockchain: total number of transactions per day, average gas used in a block, total gas used by the network in a day, average utilization of blocks, average hash rate of the network, average block time, and others [26]. This thesis studies the behavior of the total number of transactions per day and the average gas used in a block per day as there is a clear correlation between them. This correlation allows us to attempt to identify an attack on the network, indicated by increased computational effort associated with transactions, through the analysis of gas usage.

Network measurements from 7 July 2015 to 18 February 2019 were used to study normal behavior and validate our ability to detect anomalous behavior. The daily number of Ethereum network transactions and average gas usage were plotted first to visually assess network behavior over time. The red lines in Figures 11 and 12 represent dates corresponding to immediately apparent changes in normal parameter behavior as the network evolved, thereby resulting in three data segments. From the beginning of Ethereum network existence on 7 July 2015 through 2 May 2017, the number of transactions and

average gas usage values were relatively low. The increase of activity through 25 January 2018 was likely a result of the increased popularity of the smart contracts introduced by the Ethereum blockchain. The activity from 25 January 2018 to 18 February 2019

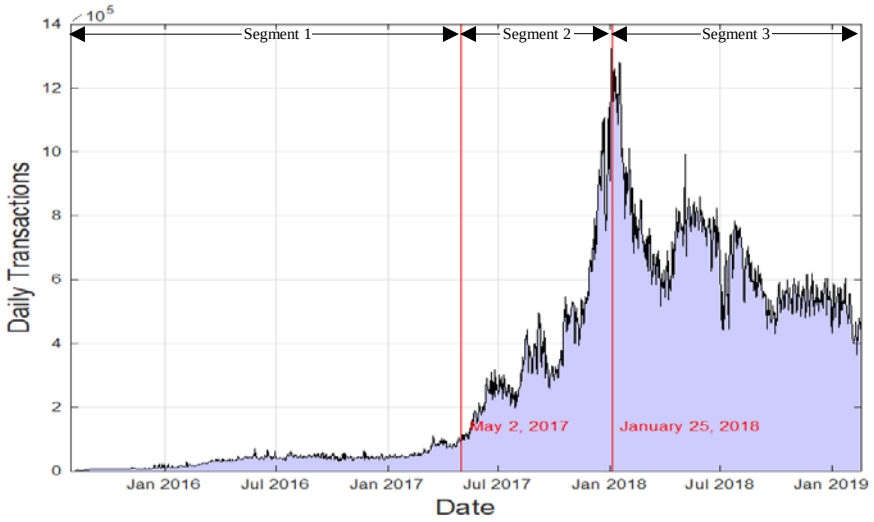


Figure 11. Time series plot of the daily number of Ethereum transactions, 7 July 2015 – 18 February 2019.

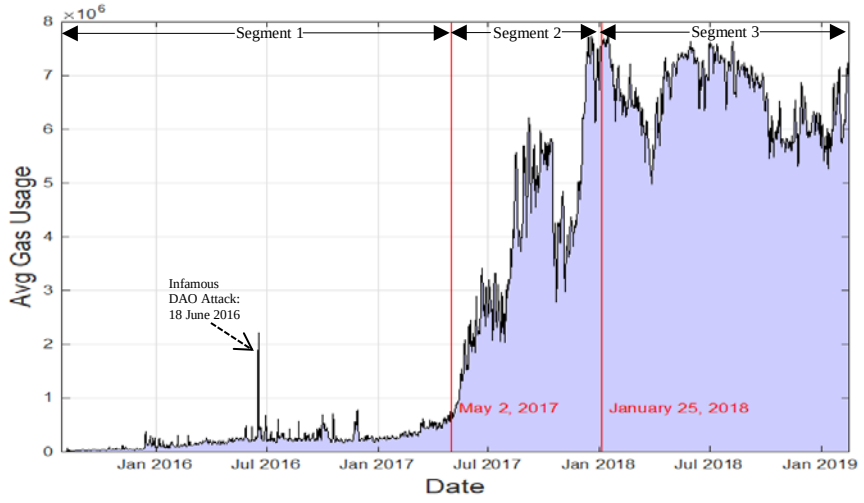


Figure 12. Time series plot of the daily average Ethereum gas usage, 7 July 2015 – 18 February 2019.

### 1. Statistical Analysis Results

While the time series plots provided some initial understanding of the data, further statistical analysis provided better comprehension of how best to characterize normal network behavior. The histograms, probability distributions, and boxplots were used to estimate thresholds for outliers. The combination of these methods for the first data segment, 7 July 2015 – 2 May 2017, is given in Figure 13. First, the average gas usage data was normalized and a histogram was created to provide an indication of a statistical distribution. Since the histogram is bimodal, a kernel distribution was fitted because it is nonparametric. The kernel distribution follows the general form as given in Equation (1) with the bandwidth parameter specified in Table 4.

The graphs labeled *Attack* correspond to the DAO attack on 18 June 2016 that was discussed in Chapter II. Processing the continuous withdrawals required additional computational efforts, which is why we see the average gas usage spike that day in Figure 12. The network behavior immediately returned to normal after implementation of the hard fork in the Ethereum blockchain. The outliers associated with the attack were then removed, so we could visualize what the normal behavior would have looked like in the absence of an attack. The boxplot was especially useful in this case, providing the median and thresholds for outlier detection comparison. The code used to create the statistical analysis plots is included in Appendix A.

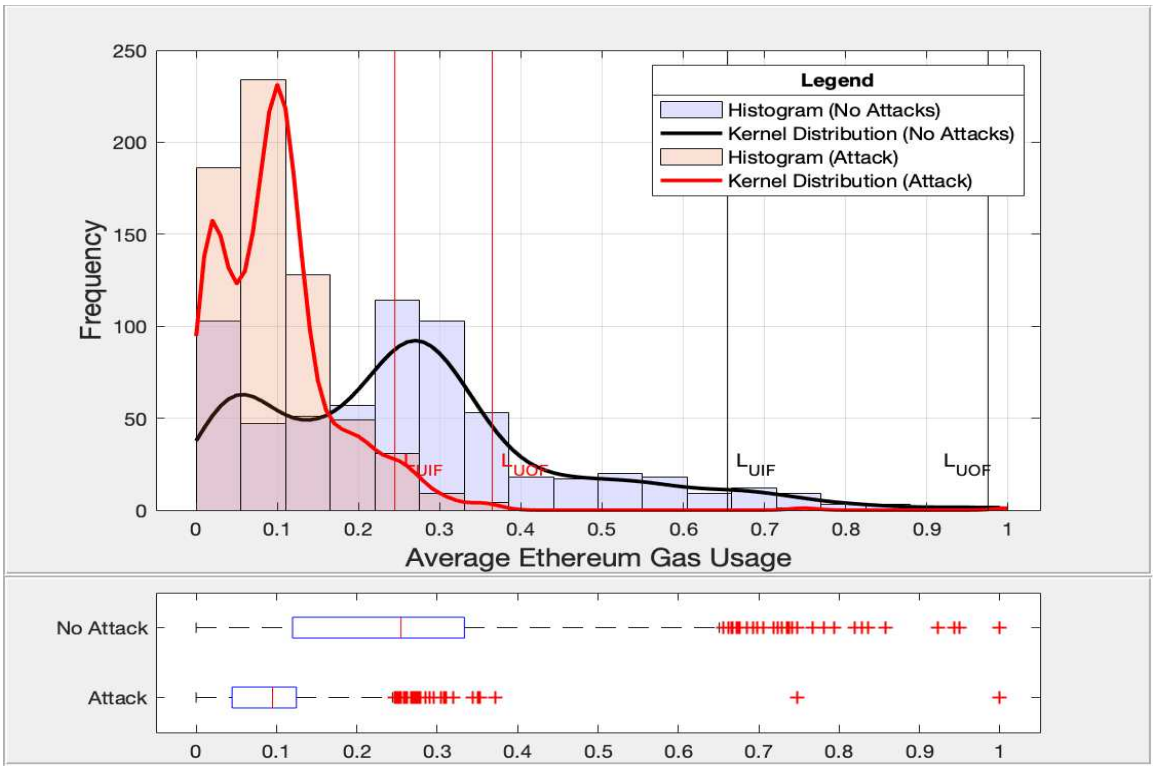


Figure 13. Comparative histograms, distribution fits, and boxplots of the average Ethereum gas usage for the first segment. The attack data corresponds to the DAO attack.

A summary of the statistical parameters with and without the attack are listed in Table 4. Values that exceed the $L_{UIF}$ and $L_{UOF}$ limits are considered outliers. The *Attack* parameters are significantly lower than the *No Attack* parameters; however, the shift in the median and tail of the *Attack* distribution resulted in an increased probability of detecting values above the $L_{UIF}$ and $L_{UOF}$ limits of 0.0019 and 0.0029, respectively. The only values that exceed the *Attack* $L_{UOF}$ limit of 0.3647 are the two major outliers associated with the DAO attack. The daily number of Ethereum transactions over all three timeframes with and without the attack yielded no results and thus are not discussed further.

Table 4. Average Ethereum gas usage statistical analysis parameters for the first segment (see Figure 13).

| Parameter | No Attack Value | Attack Value |
|---|---|---|
| Bandwidth, $h$ | 0.0448 | 0.0167 |
| Median | 0.2139 | 0.0949 |
| Upper Quartile, $Q_3$ | 0.3337 | 0.1247 |
| Lower Quartile, $Q_1$ | 0.1199 | 0.0446 |
| Interquartile Range, $d_{IQR}$ | 0.2139 | 0.0800 |
| Upper Inner Fence Limit, $L_{UIF}$ | 0.6545 | 0.2447 |
| Upper Outer Fence Limit, $L_{UOF}$ | 0.9754 | 0.3647 |
| $p_{UIF}$ | 0.0489 | 0.0508 |
| $p_{UOF}$ | 0.0021 | 0.0050 |

The statistical representation of the second segment, 3 May 2017 – 5 January 2018, is given in Figure 14. There were no reflections of the two notable attacks during this timeframe as described in Chapter II, likely because the attackers took advantage of vulnerabilities in the software code that required no additional computational efforts, or gas usage. An attack similar to that of the DAO attack in the first data set would require an exceptionally high gas usage in order to be detected because the average daily gas usage has substantially increased.

A hypothetical attack on this data segment was inserted purely for the sake of analysis. The boxplot provided the medians and thresholds for outlier detection and was useful in determining the magnitude of the hypothetical attack. The boxplot values for the original network data, labeled *No Attack*, are provided in Table 5. These values were used to determine the average gas values necessary to be detected using statistical analysis. To simulate this hypothetical attack, a major outlier value of 14,411,400 and a minor outlier value of 10,411,400 were inserted into the original data before it was normalized. Many distributions were fitted to the data, but the distribution with the tightest fit was a Gaussian distribution. The Gaussian distribution follows the general form as given in Equation (2) with the mean and standard deviation parameters specified in Table 5. The statistical

representation for the simulated attack was overlaid for comparison in Figure 14 with statistical parameter values listed in Table 5.

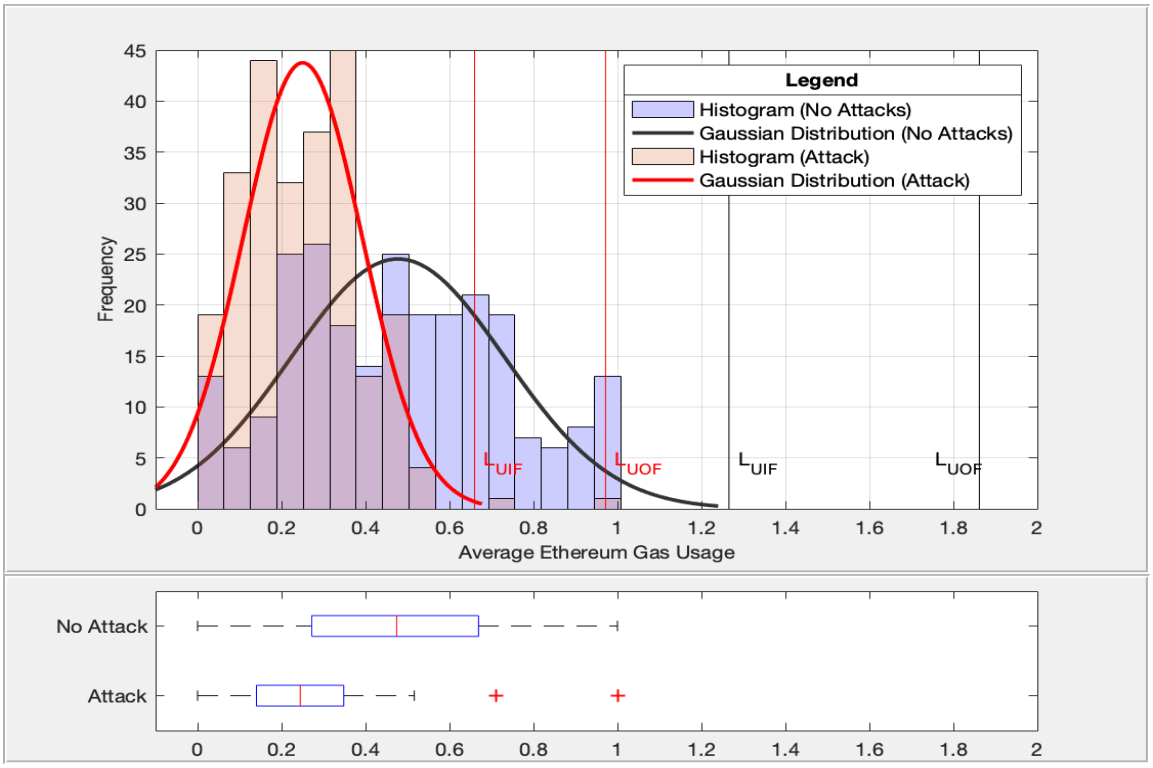


Figure 14. Comparative histograms, distribution fits, and boxplots of the average Ethereum gas usage for the second segment. The attack was simulated for analysis.

Again, we observed the *Attack* parameters were lower than the *No Attack* parameters, and there was an increased probability of detecting values above the $L_{UIF}$ limit from 0.000096 to 0.0016. At this point it is not clear whether an attack has occurred, but the probability of values that can be attributed to an attack has increased.

Table 5. Average Ethereum gas usage statistical analysis parameters for the second segment (see Figure 14).

| Parameter | No Attack Value | Attack Value |
|---|---|---|
| Mean, $m$ | 0.477137 | 0.248901 |
| Standard Deviation, $\sigma$ | 0.25423 | 0.13954 |
| Median | 0.4740 | 0.2446 |
| Upper Quartile, $Q_3$ | 0.6693 | 0.3481 |
| Lower Quartile, $Q_1$ | 0.2718 | 0.1403 |
| Interquartile Range, $d_{IQR}$ | 0.3975 | 0.2078 |
| Upper Inner Fence Limit, $L_{UIF}$ | 1.2656 | 0.6598 |
| Upper Outer Fence Limit, $L_{UOF}$ | 1.8619 | 0.9715 |
| $p_{UIF}$ | 0.000096 | 0.0016 |
| $p_{UOF}$ | ~ 0 | ~ 0 |

The statistical representation for the third segment, 6 January 2018 – 18 February 2019, is given in Figure 15. The average gas usage data was normalized and the histogram generated showed that the data is bimodal. A kernel distribution was fit to the histogram since it is nonparametric. The kernel distribution follows the general form as given in Equation (1) with the bandwidth parameter specified in Table 6. To ensure a kernel distribution was the best choice for the data, many other distributions were fitted to the histogram. The generalized extreme value distribution was a potential fit, but the shape parameter, $\kappa$, exceeds the expected range of $-\frac{1}{2} < \kappa < \frac{1}{2}$ [18]; therefore, the kernel distribution was the best choice for this data set.

Since there are no known attacks during this timeframe, a hypothetical attack similar to the one in the second segment was simulated. The average gas usage is considerably higher for this segment than for the first. A major outlier value of 13,015,700 and a minor outlier value of 9,015,700 were inserted into the original data before it was normalized.

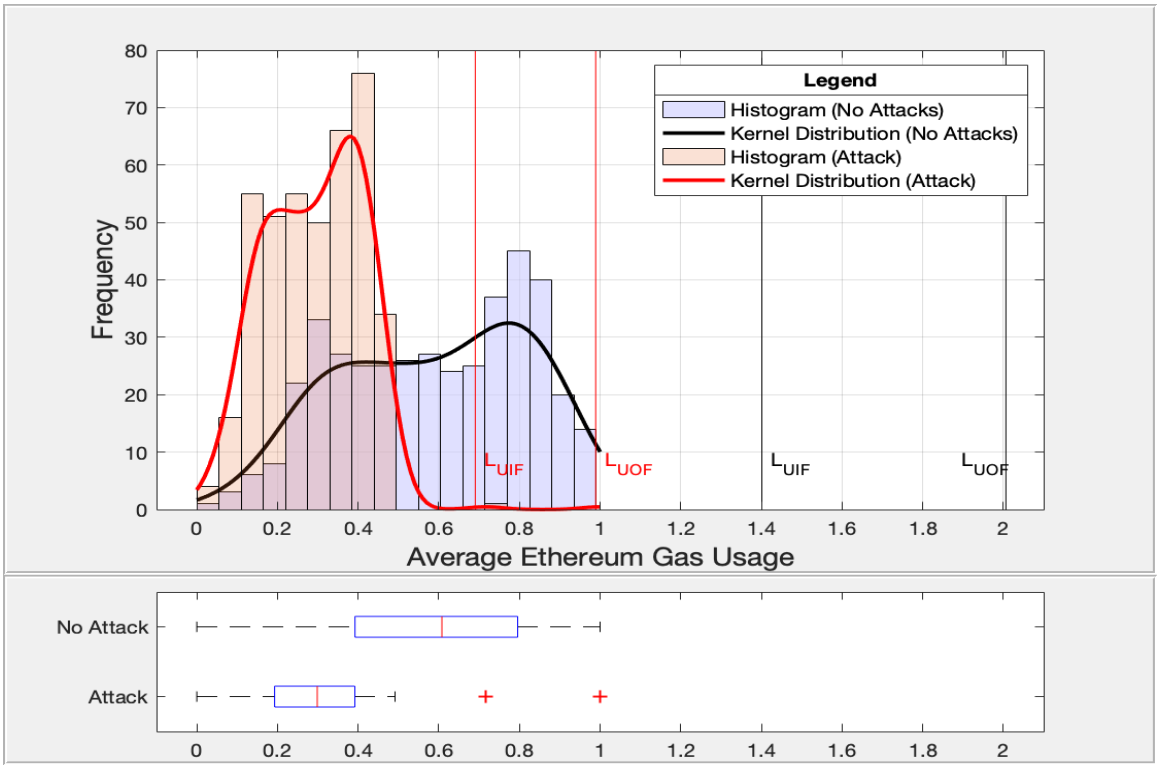


Figure 15. Comparative histograms, distribution fits, and boxplots of the average Ethereum gas usage for the third segment. The attack was simulated for analysis.

A summary of the statistical parameters with and without the simulated attack is listed in Table 6. Similar results as the previous two data segments were observed in that the *Attack* values were lower than the *No Attack* parameters. The distributions were not long-tailed as the others have been, which is why the probabilities of outlier detection, $p_{UIF}$ and $p_{UOF}$, were approximately zero prior to the simulated attack. In all three segments, the attack values shifted the distribution to the left significantly, allowing for a higher probability of more values to exceed the limits of outlier detection, $L_{UIF}$ and $L_{UOF}$.

Table 6. Average Ethereum gas usage statistical analysis parameters for the second segment (see Figure 15).

| Parameter | No Attack Value | Attack Value |
|---|---|---|
| Bandwidth, $h$ | 0.0937 | 0.0465 |
| Median | 0.6078 | 0.2987 |
| Upper Quartile, $Q_3$ | 0.7957 | 0.3918 |
| Lower Quartile, $Q_1$ | 0.3922 | 0.1927 |
| Interquartile Range, $d_{IQR}$ | 0.4036 | 0.1991 |
| Upper Inner Fence Limit, $L_{UIF}$ | 1.4011 | 0.6904 |
| Upper Outer Fence Limit, $L_{UOF}$ | 2.0064 | 0.9890 |
| $p_{UIF}$ | ~ 0 | 0.0042 |
| $p_{UOF}$ | ~ 0 | 0.0015 |

## 2. Clustering Analysis Results

While the statistical analysis method provided meaningful results for the DAO attack, the clustering method was applied to compare the results. We are attempting to best characterize normal network behavior and provide suitable thresholds for anomaly detection.

### *a. Cluster Refinement*

K-means plots for each of the three data segments of the Ethereum network were analyzed. There were minor differences observed in the way k-means and hierarchical clusters were formed as discussed in Chapter III. Only hierarchical clusters are provided because they are more indicative of the data structure and isolate anomalies.

The dendrogram for the first segment is given in Figure 16. Using average linkage, a cophenetic correlation of $\rho_a = 0.8747$ indicates that the abbreviated dendrogram is closely related to the original data structure and a good model for cluster analysis. A threshold of $\theta = 3\times10^5$ was selected because it is the highest level of similarity between the green and blue data indices before they are connected to the data indices in red. If the

distance between data points we select for a threshold is too small, data points above that threshold will be excluded from the clusters. A threshold of $\theta = 3\times10^5$ should result in two clusters and two outliers when hierarchical clusters are created.

Additionally. we expect the blue cluster to be more tightly bound than the green cluster because there is a shorter distance between the data points of the blue cluster. Since the data point indices highlighted in red are between the indices for the blue and green clusters, it alerts to an anomaly instead of a new trend in network behavior. The red data point indices are associated with the DAO attack in June 2016. The code used to create the k-means, dendrogram, and hierarchical clusters is included in Appendix B.

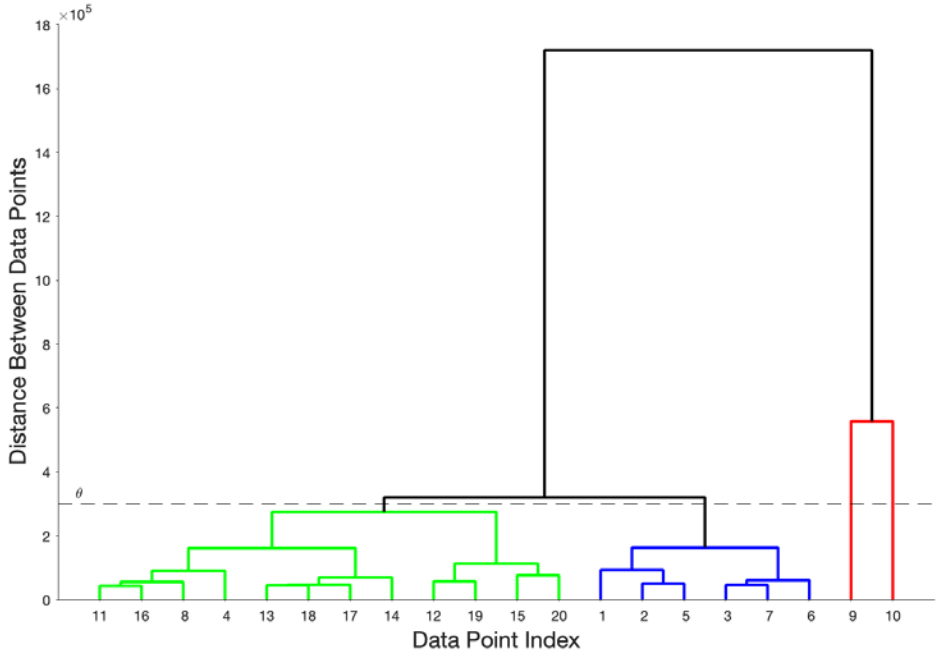


Figure 16. Dendrogram of Ethereum network parameters for the first segment. A threshold of $\theta = 3\times10^5$ results in two clusters (green and blue) and two anomalies (red). The anomalies correspond to the DAO attack.

The dendrogram for the second segment is given in Figure 17. A cophenetic correlation of $\rho_a = 0.7111$ indicates that the dendrogram is less closely related to the original data structure than the first segment. Selecting a threshold of $\theta = 1.7 \times 10^6$ should result in three clusters and no outliers when hierarchical clusters are formed. The increase in magnitude of the threshold for the second segment makes sense as the distance between data points increased with the popularity of the Ethereum network as previously observed in Figures 11 and 12.

As in the statistical analysis subsection for this data segment, we did not see any reflections of the attacks that occurred during this timeframe. Unless there is an excessive amount of gas usage or an excessive number of transactions, we will not be alerted to the attack using clustering analysis.

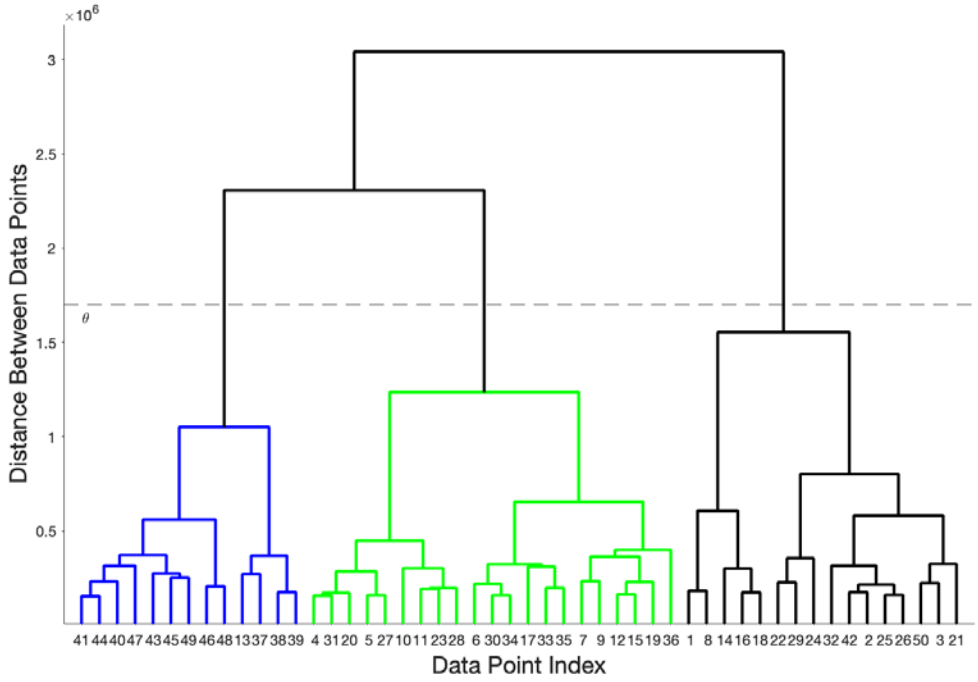


Figure 17. Dendrogram of Ethereum network parameters for the second segment. A threshold of $\theta = 1.7 \times 10^6$ results in three clusters (blue, green, and black).

The dendrogram for the third segment is given in Figure 18. A cophenetic correlation of $\rho_a = 0.7240$ indicates that the dendrogram is less closely related to the original data structure than the first segment. Selecting a threshold of $\theta = 8\times10^5$ should result in two clusters and no anomalies.

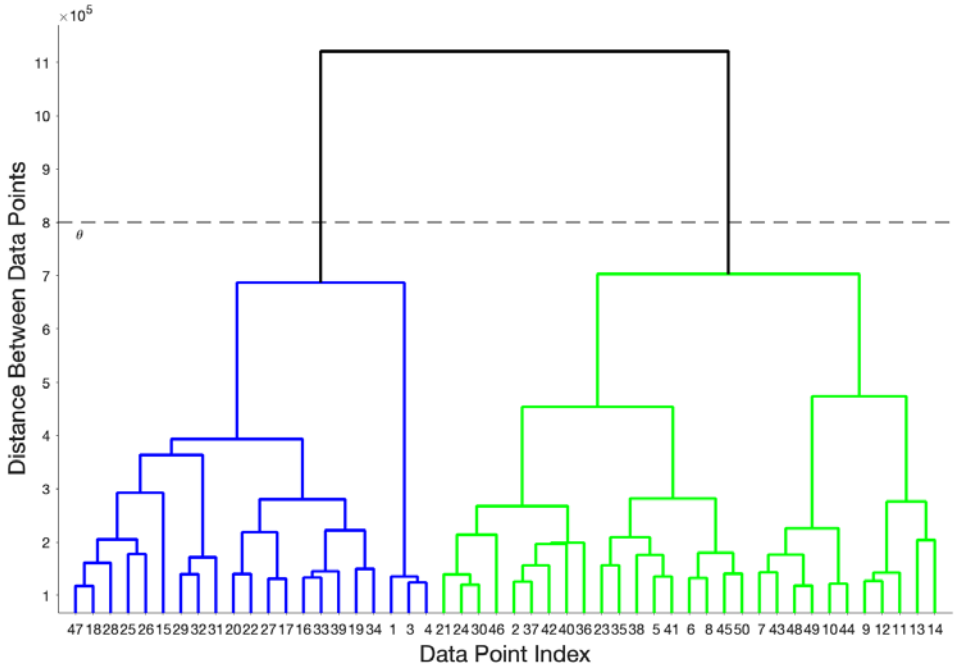


Figure 18. Dendrogram of Ethereum network parameters for the third segment. A threshold of $\theta = 8\times10^5$ results in two clusters (blue and green).

### *b. Hierarchical Clustering*

The thresholds determined from the dendrograms were then used to form clusters; data points that exceed the threshold were identified as outliers and not part of the clusters. The hierarchical clustering algorithm for the first data segment automatically determined one of the two anomalous values associated with the DAO attack as indicated by the red data point in Figure 19. It is the only data point that is clearly not correlated with the rest of the data points. Both anomalous data points from the dendrogram in Figure 16 would have been displayed if a threshold of $\theta = 1.72\times10^6$ were used; however, we would lose the benefit of the sub-clusters. The sub-clusters are valuable in determining whether data points

correspond to new normal network behavior as the distance between data points increases with network evolution or if they correspond to malicious activity.

The number of data points for each cluster were calculated using Equation (11) and are listed in Table 7. The majority of the data points are included in Cluster 1. The cluster is also more tightly bound as expected from the dendrogram. The data points in Cluster 2 represent the beginning of the increase in popularity of the Ethereum blockchain which resulted in more gas usage. This also serves as an indicator that the thresholds will need to be reevaluated once the network activity settles into a new normal.

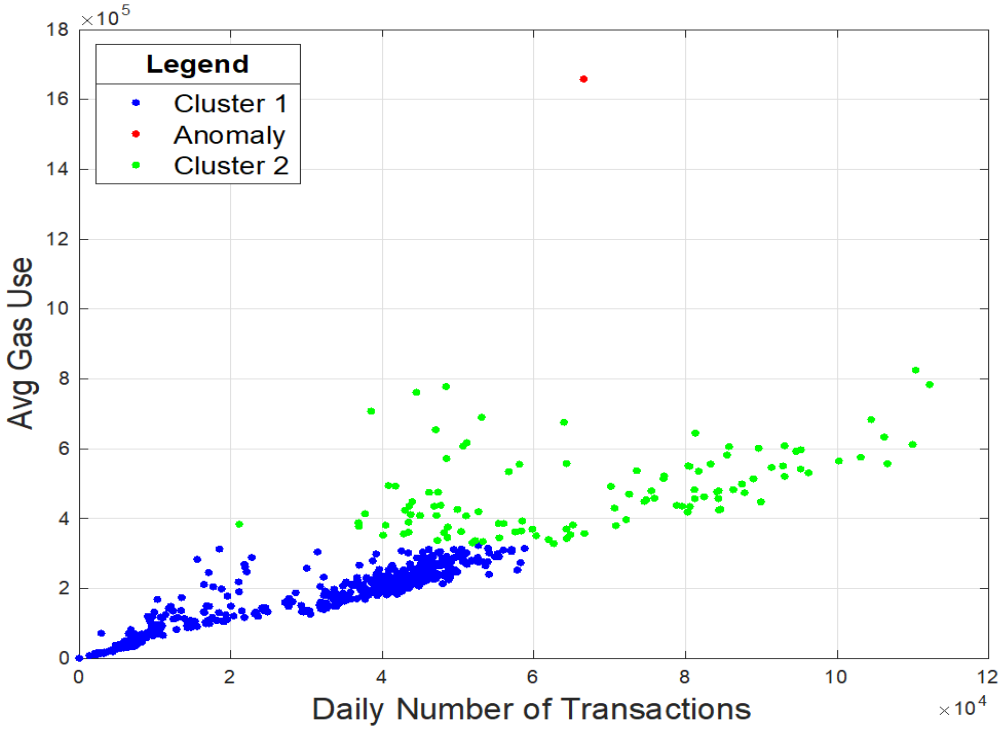


Figure 19. Hierarchical clustering of Ethereum network parameters for the first segment. The anomaly corresponds to the DAO attack.

Table 7. Cluster comparison of the Ethereum data for the first segment (see Figure 19).

| Cluster | Percentage of Data Points, $\gamma_i$ |
|---|---|
| Cluster 1 | 82.27% |
| Cluster 2 | 17.42% |
| Anomaly | 0.31% |

The hierarchical clustering for the second segment displayed no anomalous values as expected from the corresponding dendrogram in the previous subsection and from the lack of additional computational efforts required in the attacks that occurred during this timeframe. A hypothetical attack was simulated to determine at what magnitude the clustering algorithm would detect an attack using normal network behavior thresholds. We determined a threshold of $\theta = 1.7 \times 10^6$ from the dendrogram of the second data segment. An average gas value of $9.4 \times 10^6$ was inserted into the original data set with a corresponding number of transactions of $8.04 \times 10^5$; the algorithm correctly identified the anomaly as illustrated by the red data point in Figure 20. The DAO attack average gas values were $1.6583 \times 10^6$ and $2.2162 \times 10^6$. These values would be reflected as minimum values for the second and third segments because the daily average values have increased substantially from the beginning of the life of the Ethereum blockchain network.

The number of data points for each cluster prior to the hypothetical attack were calculated using Equation (11) and are listed in Table 8. The cluster trends and percentages represent the changing gas usage illustrated in Figures 11 and 12. During the dramatic increase in public use of the Ethereum network during this timeframe, the network behavior became less predictable. All networks are dynamic; during periods of time where the behavior is changing, thresholds need to be reevaluated more frequently to ensure the network is not under attack and to gain a sense of the new network normal once it settles.

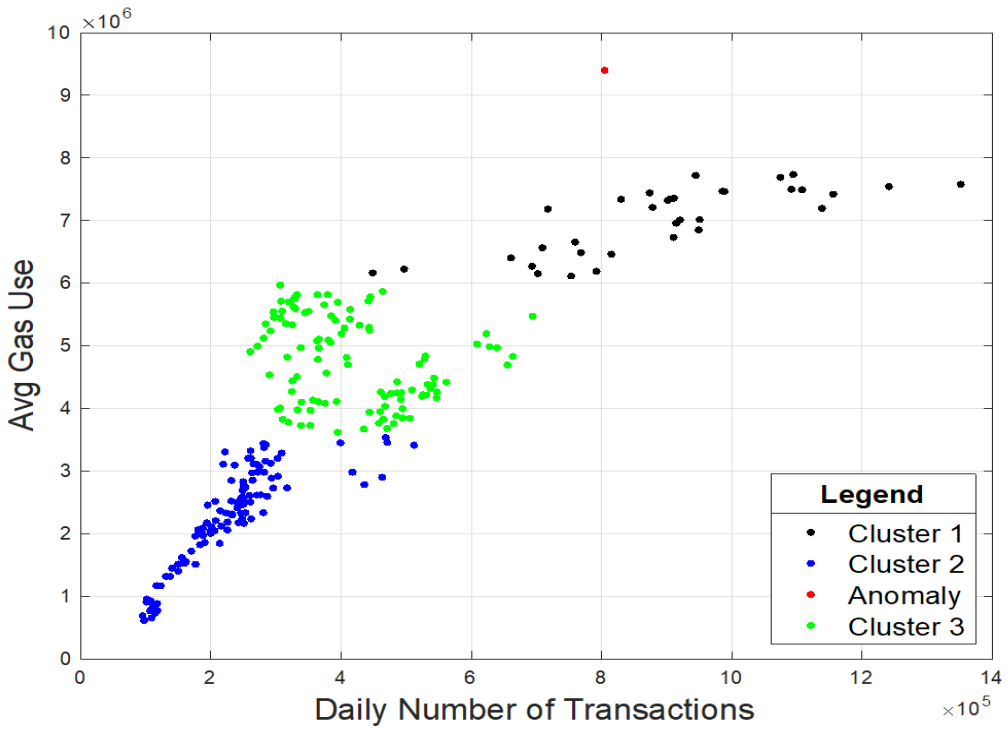


Figure 20. Hierarchical clustering of Ethereum network parameters for the second segment. The cluster behavior reflects the increase in Ethereum network usage. The anomaly corresponds to a simulated attack.

Table 8. Cluster comparison of the Ethereum data from the second segment (see Figure 20).

| Cluster | Percentage of Data Points, $\gamma_i$ |
|---|---|
| Cluster 1 | 43.15% |
| Cluster 2 | 13.71% |
| Cluster 3 | 42.74% |

The hierarchical clustering for the third segment also displayed no anomalous values because there are no reflections of any attacks as discussed in the statistical analysis results subsection. A hypothetical attack was simulated in the same way as for the second segment. We determined a threshold of $\theta = 8\times10^5$ from the dendrogram of the original data. A value of $8.33\times10^6$ was inserted into the original data set as a gas value with a

corresponding number of transactions of $8\times10^5$, and the algorithm again correctly identified the anomaly as given by the red data point in Figure 21.

The number of data points for each cluster prior to the hypothetical attack were calculated using Equation (11) and are listed in Table 9. Cluster 2 includes the majority of the data points and represents the gas usage once the popularity of the Ethereum blockchain network had settled into a new normal behavior.

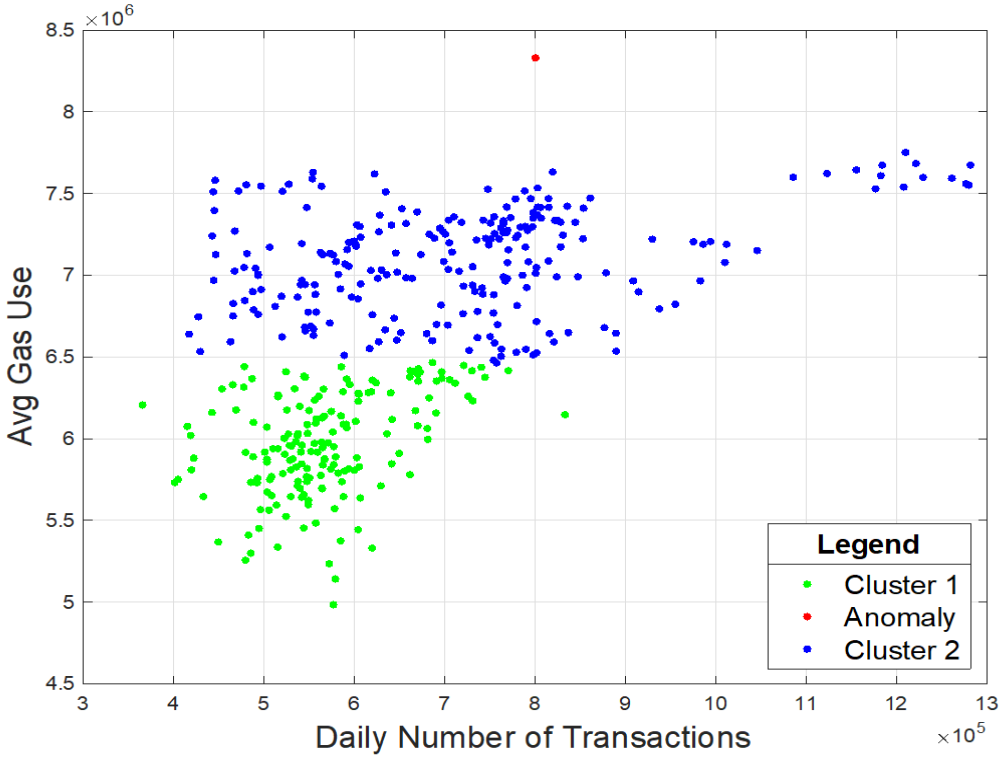


Figure 21. Hierarchical clustering of Ethereum network parameters for the third segment. The cluster behavior reflects the new normal behavior in Ethereum network usage. The anomaly corresponds to a simulated attack.

Table 9. Cluster comparison of the Ethereum data from the third segment (see Figure 21).

| Cluster | Percentage of Data Points, $\gamma_i$ |
|---|---|
| Cluster 1 | 43.77% |
| Cluster 2 | 56.23% |

The intention of this thesis is to explore the capabilities and limitations of traditional statistical methods and machine learning clustering methods and, if possible, to resolve which is best suited for anomaly detection in blockchain-based systems. A comparison of the threshold values from both statistical analysis, $L_{UIF}$ and $L_{UOF}$, and hierarchical clustering analysis, $\theta$, for all three segments is provided in Table 10, indicating that the clustering method detects attacks at a lower magnitude than statistical methods.

For example, consider the first data segment. Statistical analysis provided a threshold of $L_{UOF} = 8.0833 \times 10^5$ and clustering analysis provided a threshold of $\theta = 3 \times 10^5$; both were successful at detecting the DAO attack. The second segment showed a much-improved performance using clustering methods than statistical methods. In this case, statistical analysis provided a threshold of $L_{UOF} = 1.4019 \times 10^7$ and required a gas usage value in excess of $1.4411 \times 10^7$ to be detected. For the same data segment, clustering analysis provided a threshold of $\theta = 1.7 \times 10^6$ and detected an attack resulting in a gas usage of $9.4 \times 10^6$, which is a significantly lower threshold for detection capability. Additionally, hierarchical clustering provides better insight into the blockchain data structure.

Table 10. Comparison of thresholds from statistical analysis, $L_{UIF}$ and $L_{UOF}$, and hierarchical clustering analysis, $\theta$, for the Ethereum data from all three segments.

| Method | Threshold | Attack Value |
|---|---|---|
| First Segment | | |
| Statistical | $8.0833 \times 10^5$ | $2.2162 \times 10^6$, $1.6583 \times 10^6$ |
| Clustering | $3.0 \times 10^5$ | $2.2162 \times 10^6$, $1.6583 \times 10^6$ |
| Second Segment | | |
| Statistical | $1.4019 \times 10^7$ | $1.4411 \times 10^7$, $1.0411 \times 10^7$ |
| Clustering | $1.7 \times 10^6$ | $9.4 \times 10^6$ |
| Third Segment | | |
| Statistical | $1.0554 \times 10^7$ | $1.3016 \times 10^7$, $9.0157 \times 10^6$ |
| Clustering | $8.0 \times 10^5$ | $1.1 \times 10^6$ |

## B. LOCAL BLOCKCHAIN RESEARCH NETWORK

We wanted to analyze some blockchain network data that is independent of the financial realm to see what reflections of an attack may be identified by the anomaly detection scheme. We were able to design an experiment in which we conducted a doorknob-rattling attack on a local blockchain research network; in the scenario, login attempts were counted as transactions [27]. Normal network behavior was established as 20 to 30 attempted logins in five-minute intervals on our victim machine. A few extra login attempts were inserted in specified intervals to see if we would be able to observe indications of outliers. A randomly varying gas value associated with each transaction was similarly inserted since average daily gas usage was critical in outlier detection for the Ethereum network data. The intent was to vary the gas usage proportionally with the number of transactions.

A visual representation of the number of transactions and total gas usage for the doorknob-rattling scenario are given in Figures 22 and 23, respectively. For this data set, total instead of average gas was used because the random insertion of gas values created a minimum value that theoretically could not happen based on the number of transactions. A doorknob-rattling attack would not create an anomaly with too few transactions nor too little gas usage as computational efforts are necessary to process transactions. Since the data collected for this experiment was relatively small, consisting of only sixteen data points, the randomly inserted smaller gas usage value had a substantial effect on the interval average. Using the total gas usage per interval corrected this inaccuracy and further helped us realize that both the statistical and clustering methods require case-by-case parameter considerations.

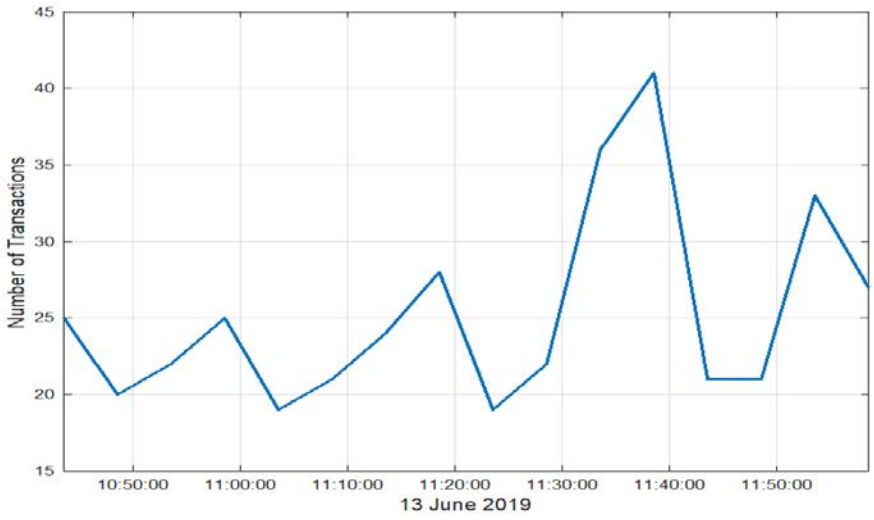


Figure 22. Time series plot of the number of transactions during the doorknob-rattling scenario. Normally, 20 to 30 attempted logins are expected every five-minutes.

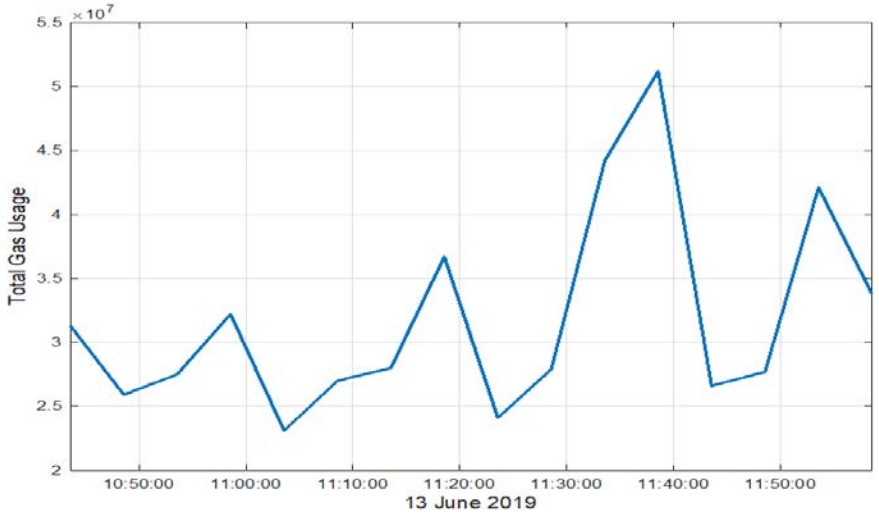


Figure 23. Time series plot of the total gas usage during the doorknob-rattling scenario. Gas usage varies proportionally with the number of transactions.

### 1. Statistical Analysis Results, Doorknob-Rattling Attack

The histogram, probability distribution, and boxplot were used to determine thresholds for anomaly detection as before. The distribution with the tightest fit was a generalized extreme value distribution. The boxplot proved to be more advantageous in

visualizing the presence of outliers as given in Figure 24. Although three intervals exceeded the established normal network behavior of 20 to 30 transactions, only one was significant enough to be classified as an outlier using statistical analysis. The generalized extreme value distribution follows the general form as given in Equation (3) with the scale, location, and shape parameters specified in Table 11.

A summary of the statistical values is provided in Table 11. The limits of $L_{UIF}$ and $L_{UOF}$ are 37 and 47 transactions, respectively. This resulted in the determination of one minor outlier of 41 transactions in one time interval. We know that this is not a very conservative threshold because we defined normal behavior to be 20 to 30 transactions. Additionally, we expect to get an anomaly about 7.3% of the time. For this experiment, that value is not significant, but requiring that many data points to be investigated for a large network is unreasonable.

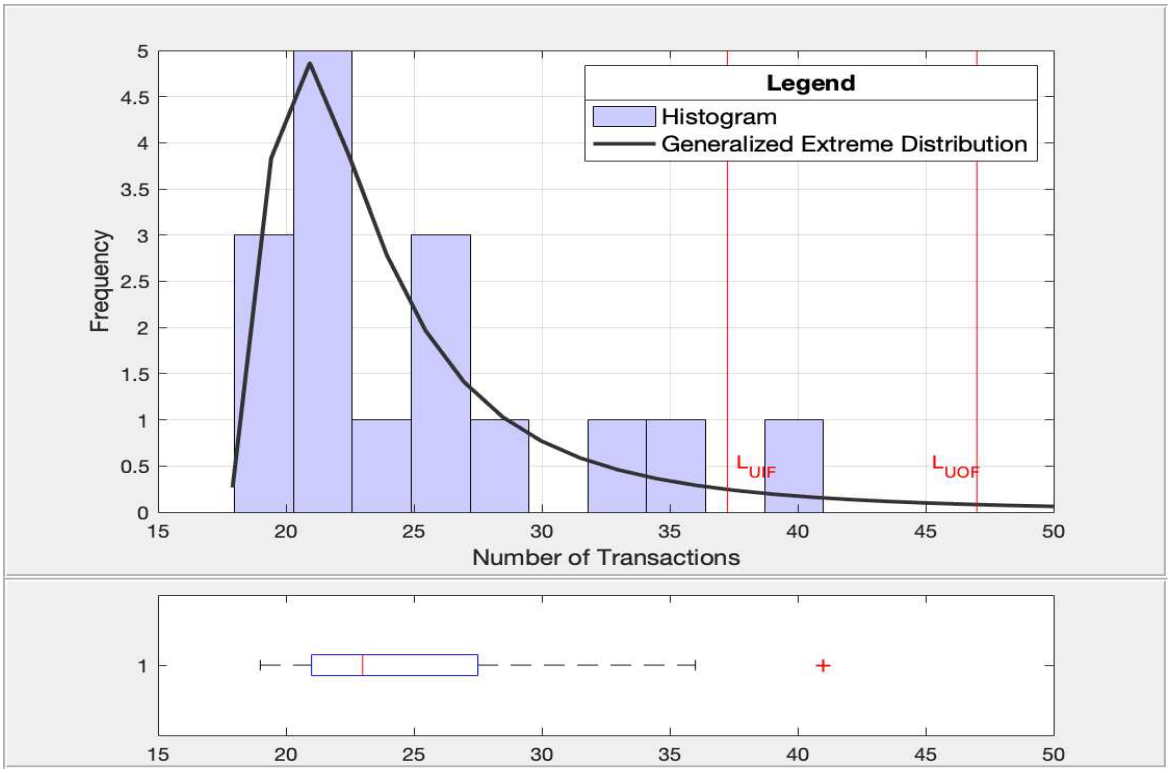


Figure 24. Histogram, distribution fit, and boxplot of the number of transactions during the doorknob-rattling attack scenario. There is one minor outlier of 41 transactions.

Table 11. Doorknob-rattling attack scenario statistical parameters (see Figure 24).

| Parameter | Value |
|---|---|
| Scale, $\alpha$ | 3.04272 |
| Location, $x_o$ | 21.6909 |
| Shape, $\kappa$ | 0.48112 |
| Median | 23 |
| Upper Quartile, $Q_3$ | 27.5 |
| Lower Quartile, $Q_1$ | 21 |
| Interquartile Range, $d_{IQR}$ | 6.5 |
| Upper Inner Fence Limit, $L_{UIF}$ | 37.25 |
| Upper Outer Fence Limit, $L_{UOF}$ | 47 |
| $p_{UIF}$ | 0.0730 |
| $p_{UOF}$ | 0.0346 |

## 2. Clustering Analysis Results, Doorknob-Rattling Attack

This section studies the behavior of the total number of transactions per five-minute interval and the total gas used during the doorknob-rattling scenario using machine learning clustering methods.

### *a. Cluster Refinement*

We determined from the dendrogram in Figure 25 that a threshold of $\theta = 7.078 \times 10^6$ should result in one cluster and three outliers. The cophenetic correlation of $\rho_a = 0.8816$ indicates that the dendrogram is a good representation of the original data structure. The blue section corresponds to the normal behavior and can be further divided into two sub-clusters. The green section represents two data points that do not reach the threshold for anomaly detection but are suspicious in nature because they have some similarity to the attack value in red. Therefore, these data points require further evaluation to determine if they represent new network normal behavior or if they are a result of an

attack. The dendrogram provides a more accurate visualization of how the data is connected than the statistical analysis.

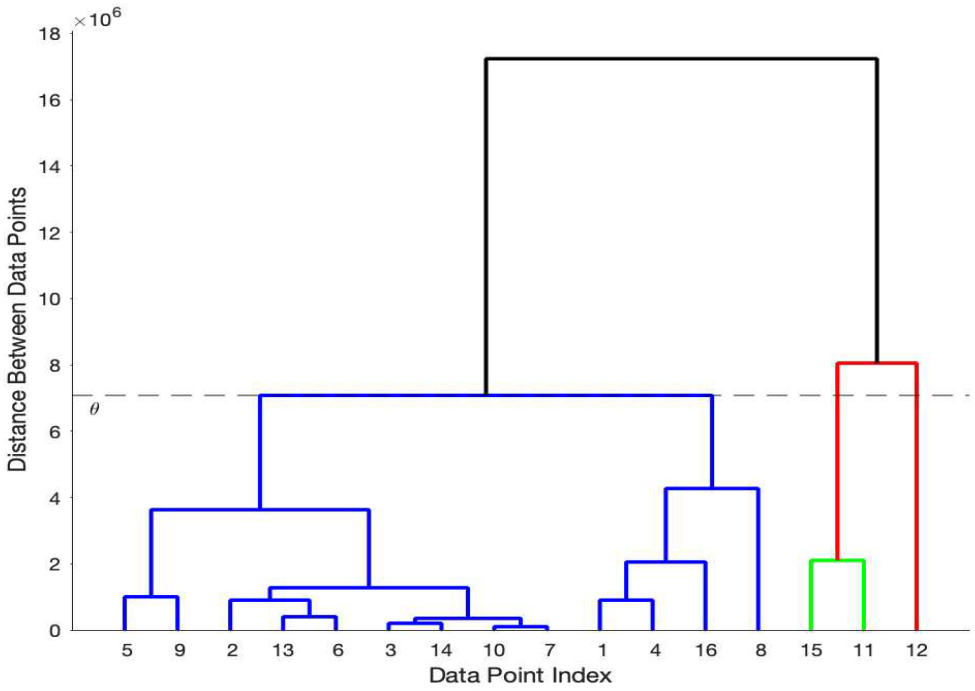


Figure 25. Dendrogram during doorknob-rattling scenario. A threshold of $\theta = 7.078 \times 10^6$ results in one cluster (blue), suspicious behavior (green), and one anomaly (red).

### *b. Hierarchical Clustering*

With the optimal *k* value determined and a threshold identified, hierarchical clustering isolates data points that are dissimilar from the others automatically as illustrated in Figure 26. Using Equation (11), the number of data points for each cluster were calculated and listed in Table 12. The anomaly is able to make up a significant percentage of the total data because the data set was very small; we did not see the same impact for the larger Ethereum network.

When we ran the experiment, specific information regarding the magnitudes or time intervals of attacks were not provided to the scheme. The anomaly detection scheme was successful in identifying normal behavior and doorknob-rattling attack behavior. The hierarchical clustering methods were successful in detecting all three attacks whereas statistical analysis only detected one attack.

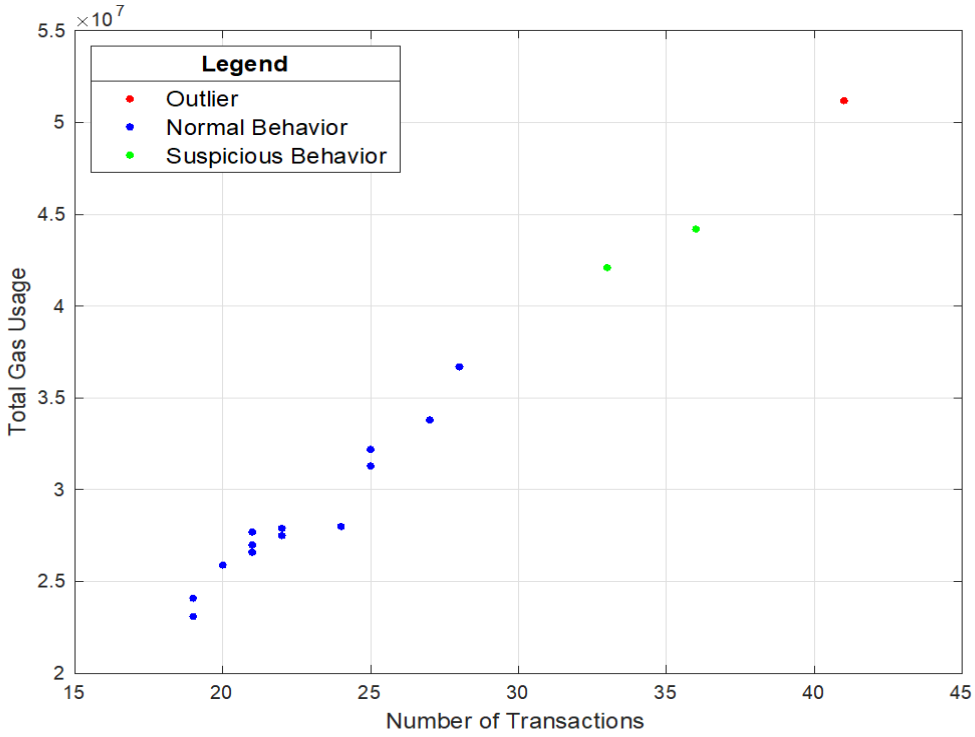


Figure 26. Hierarchical clustering during doorknob-rattling scenario. The cluster behavior reflects normal activity of 19 to 28 transactions. Suspicious behavior is 30 to 40 transactions, and more than 40 transactions are attacks.

Table 12. Cluster comparison for doorknob-rattling attack scenario (see Figure 26).

| Cluster | Percentage of Data Points, $\gamma_i$ |
|---|---|
| Normal Behavior | 68.75% |
| Suspicious Behavior | 12.50% |
| Anomaly | 6.25% |

In the first section of this chapter, statistical analysis including the analysis of histograms, probability distributions, and boxplots were used to estimate thresholds for outliers in the public Ethereum network data. Then, the hierarchical clustering process was conducted to compare the results. The second section of this chapter successfully applied the anomaly detection scheme to an experimental local blockchain research network under a doorknob-rattling attack scenario.

# V. CONCLUSIONS

The objective of this thesis was to characterize normal network behavior in blockchain-based systems and to develop a way to detect anomalous behavior by exploring statistical and machine learning techniques.

The general framework of the anomaly detection scheme we developed characterizes the number of transactions and average gas usage parameters as normal or anomalous using statistical analysis and hierarchical clustering methods. Statistical analysis included histograms, statistical distributions, and boxplots to calculate limits to evaluate outliers. In the presence of an attack, the statistical values shifted the distributions to the left significantly, allowing for a higher probability of more values to exceed the limits of outlier detection.

Hierarchical clusters were formed through the use of dendrograms, which determined lower thresholds while proving more effective at anomaly detection than statistical methods in our applications. An additional benefit of the hierarchical clustering is that the threshold from the dendrogram creates clusters and sub-clusters. The sub-clusters are valuable in determining whether data points correspond to new normal network behavior or if they correspond to malicious activity.

## A. SIGNIFICANT RESULTS

The work presented in this thesis contributes to understanding data behavior associated with blockchain-based systems and how anomaly detection could occur in such systems. We proposed an anomaly detection scheme that inputs data into statistical and clustering methods for analysis. The anomaly detection scheme was validated by analyzing the DAO attack on the public Ethereum network and by simulating hypothetical attacks on the same network as it evolved over time. Both methods were successful at detecting attacks that required additional gas usage.

We demonstrated that hierarchical cluster analysis was more powerful because it detected attacks at a significantly lower magnitude than statistical methods. We also showed that threshold determination for anomaly detection can be achieved without prior

knowledge of the network behavior. The determination is affected by the dynamic nature of a network and thus requires constant reevaluation.

We extended our evaluation by conducting experiments on a local blockchain research network to test the proposed anomaly detection scheme. Normal network behavior of 20 to 30 logins per five-minute time interval were established. Additional login attempts were made during certain intervals to simulate a doorknob-rattling attack. Both the statistical and clustering methods proved successful in determining normal network behavior and alerting to attacks. Statistical analysis indicated one minor outlier at 41 login attempts in one interval. The clustering analysis method indicated three anomalies at 33, 36, and 41 logins in three separate intervals, indicating three separate attacks. Hierarchical clustering provided better insight into the blockchain data structure because there were in fact three doorknob-rattling attacks on the network.

## B. RECOMMENDATIONS FOR FUTURE WORK

There are several possibilities for future work. First, the process for cluster refinement and threshold determination was manually performed. An automated process for taking measurable parameters, clustering the data, determining the centroids, determining which outliers within a specified distance are new normal behaviors and which should be evaluated as possibly malicious could be made similar to that described in [28].

There is much yet to be explored within the measurable network parameters of blockchains for determining normal behavior. This thesis only studied two of many parameters publicly archived for analysis; exploring the others may provide insight into behavior characterization or attack indication. Additionally, adding a node to the public Ethereum network in order to measure and process the parameters directly would provide more granularity of all the parameters for more extensive analysis.

The proposed anomaly detection scheme only detected attacks that involved an increase in computational power; attacks that exploit vulnerabilities in software or smart contract code were not detected. There are several avenues that would provide additional data sets for analysis that are independent of the Ethereum network. For example, Walmart plans to launch a major blockchain-based supply-tracking project in 2019 [29], [30], and

there are several blockchain projects currently underway within the Department of Defense [1], [31]. Evaluating these blockchain-based system behaviors and the types of attacks they are susceptible to would provide additional insight into anomaly detection capabilities and limitations.

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX A. STATISTICAL ANALYSIS SCRIPT

This appendix includes the code written as a MATLAB script to perform the statistical analysis method of a given data set. The code is from the MATLAB documentation. Comments are in green text.

```
% Data downloaded on 18FEB19 from: https://www.etherchain.org/charts
filename1=('TransVsGas.xlsx');
Date1=xlsread(filename1,'A:A');
DailyNumTx=xlsread(filename1,'B:B');
DailyNumTx=normalize(DailyNumTx,'range');
% DailyNumTx = data with DAO attack
AvgGasUse=xlsread(filename1,'C:C');
AvgGasUse=normalize(AvgGasUse,'range');
% AvgGasUse = data with DAO attack
t1=datetime(2015,7,30) + caldays(0:642);

filename2=('TransVsGas1.xlsx');
Date2=xlsread(filename2,'A:A');
DailyNumTx2=xlsread(filename2,'B:B');
DailyNumTx2=normalize(DailyNumTx2,'range');
% DailyNumTx2 = data with DAO attack removed
AvgGasUse2=xlsread(filename2,'C:C');
AvgGasUse2=normalize(AvgGasUse2,'range');
% AvgGasUse2 = data with DAO attack removed
t1=datetime(2015,7,30) + caldays(0:640);

% Histogram
% Matlab documentation:
% https://www.mathworks.com/help/matlab/ref/hist.html

% Distribution Fit
% Matlab documentation:
% https://www.mathworks.com/help/stats/fitdist.html

figure
hp1 = uipanel('position',[0 .25 1 .75]);
hp2 = uipanel('position',[0 0 1 .25]);
axes('Parent',hp1);
X = 0:.055:1;
h=histogram(AvgGasUse2,X,'FaceColor',[.8 .8 1]);
hold on
pd_kernel = fitdist(AvgGasUse2,'Kernel');
x = 0:.01:1;
pdf_kernel = pdf(pd_kernel,x)*32;
plot(x,pdf_kernel,'Color','k','LineWidth',2);
hold on
xlabel('Average Ethereum Gas Usage','FontSize',14)
ylabel('Frequency','FontSize',14)

h1=histogram(AvgGasUse,X,'FaceColor',[0.91 0.41 0.17]);
```

```
hold on
pd_kernel1 = fitdist(AvgGasUse,'Kernel');
pdf_kernel1 = pdf(pd_kernel1,x)*30;
plot(x,pdf_kernel,'Color','r','LineWidth',2);
grid on;
hold on

% calculate the statistics
q1=quantile(AvgGasUse,[0.25 0.75]);
iqr1=iqr(AvgGasUse);
q2=quantile(AvgGasUse2,[0.25 0.75]);
iqr2=iqr(AvgGasUse2);

% find the "inner fences"
minor_attack=q1(2)+(1.5*iqr1);
minor_out=q2(2)+(1.5*iqr2);

% find the "outer fences"
major_attack=q1(2)+(3*iqr1);
major_out=q2(2)+(3*iqr2);

% plot the limits of the fences
% vline function from MathWorks:
% https://www.mathworks.com/matlabcentral/fileexchange/1039-hline-and- % vline
H1=vline(minor_out, 'k', 'L{_U_I_F}');
hold on
H2=vline(major_out, 'k', 'L{_U_O_F}');
hold on
H3=vline(minor_attack, 'r', 'L{_U_I_F}');
hold on
H4=vline(major_attack, 'r', 'L{_U_O_F}');

% find the probabilities
p1_UIF=cdf(pd_kernel,minor_out,'upper');
p1_UOF=cdf(pd_kernel,major_out,'upper');
p2_UIF=cdf(pd_kernel1,minor_attack,'upper');
p2_UOF=cdf(pd_kernel1,major_attack,'upper');

hold off
alpha(h1,.2)
lgd=legend({'Histogram (No Attacks)','Kernel Distribution (No Attacks)','Histogram (Attack)','Kernel
Distribution (Attack)'},'Location','ne','FontSize',10);
title(lgd,'Legend')

% add the boxplot
% Matlab documentation:
% https://www.mathworks.com/help/stats/boxplot.html
axes('Parent',hp2)
y=[AvgGasUse;AvgGasUse2];
g=[ones(size(AvgGasUse));2*ones(size(AvgGasUse2))];
boxplot(y,g,'Labels',{'Attack','No Attack'},'Orientation','horizontal')
```

# APPENDIX B. K-MEANS AND HIERARCHICAL CLUSTERING SCRIPT

This appendix includes the code written as a MATLAB script to perform the k-means and hierarchical clustering of a given data set. The code is from the MATLAB documentation. Comments are in green text.

```
% Data downloaded on 18FEB19 from: https://www.etherchain.org/charts
filename=('TransVsGas.xlsx');
Date=xlsread(filename,'A:A');
DailyNumTx=xlsread(filename,'B:B');
AvgGasUse=xlsread(filename,'C:C');
t1=datetime(2015,7,30) + caldays(0:642);

% K-means
% Matlab documentation:
% https://www.mathworks.com/help/stats/kmeans.html
k=2;
opts = statset('Display','final');
X=[DailyNumTx, AvgGasUse];
[idx,C] = kmeans(X,k,'Distance','sqeuclidean','Replicates',5,'Options',opts);

figure
plot(X(idx==1,1),X(idx==1,2),'b.','MarkerSize',12)
hold on
plot(X(idx==2,1),X(idx==2,2),'g.','MarkerSize',12)
plot(C(:,1),C(:,2),'kx','MarkerSize',15,'LineWidth',3)
grid on
xlabel('Daily Number of Transactions','FontSize',15)
ylabel('Avg Gas Usage','FontSize',15)
lgd=legend({'Cluster 1','Cluster 2','Centroids'},'Location','nw');
title(lgd,'Legend')

% Hierarchical Clustering
% Matlab documentation:
% https://www.mathworks.com/help/stats/examples/cluster-analysis.html
eucD = pdist(X,'euclidean');
clustTreeEuc = linkage(eucD,'average');
cophenet=cophenet(clustTreeEuc,eucD)
[h,nodes] = dendrogram(clustTreeEuc,20);
set(h,'LineWidth',2)
h_gca = gca;
h_gca.TickDir = 'out';
h_gca.TickLength = [.002 0];
H=hline(3*10^5,'--k','{\theta}');
% hline function from MathWorks: https://www.mathworks.com/matlabcentral/fileexchange/1039-hline-and-vline

xlabel('Data Point Index','FontSize',16)
ylabel('Distance Between Data Points','FontSize',16)
```

```
figure
ptsymb = {'b.','r.','g.'};
hidx = cluster(clustTreeEuc,'criterion','distance','cutoff',3e+05);
% cutoff = 1.72e+06 for no sub-clusters, both anomalies detected
for i = 1:3
    clust = find(hidx==i);
    plot(X(clust,1),X(clust,2),ptsymb{i},'MarkerSize',12);
    hold on
end

h(1).Color = 'g';
h(2).Color = 'g';
h(3).Color = 'b';
h(4).Color = 'g';
h(5).Color = 'b';
h(6).Color = 'g';
h(7).Color = 'g';
h(8).Color = 'b';
h(9).Color = 'g';
h(10).Color = 'g';
h(11).Color = 'g';
h(12).Color = 'b';
h(13).Color = 'g';
h(14).Color = 'g';
h(15).Color = 'b';
h(16).Color = 'g';
h(17).Color = 'k';
hold off

xlabel('Daily Number of Transactions','FontSize',16);
ylabel('Avg Gas Use','FontSize',16);
grid on
lgd2=legend({'Cluster 1','Anomaly','Cluster 2'},'Location','nw','FontSize',14);
title(lgd2,'Legend')

cluster_counts=[sum(ismember(nodes,[11 16 8 4])) sum(ismember(nodes,[13 18 17 14]))
sum(ismember(nodes,[12 19 15 20])) sum(ismember(nodes,[1 2 5])) sum(ismember(nodes,[3 7 6]))
sum(ismember(nodes,[9 10]))];
```

# LIST OF REFERENCES


[1] H. S. Kenyon, “Navy raises anchor on blockchain,” *Signal*, Mar. 2019. [Online]. Available: https://www.afcea.org/content/navy-raises-anchor-blockchain

[2] J. McCarter, “DON innovator embraces a new disruptive technology: Blockchain,” Naval Innovation Advisory Council, Secretary of the Navy, June 2017. [Online]. Available: https://www.secnav.navy.mil/innovation/Pages/2017/06/BlockChain.aspx

[3] N. Atzei, M. Bartoletti, and T. Cimoli, “A survey of attacks on Ethereum smart contracts,” in *Proc. of the 6th Intl. Conf. on Principles of Security and Trust*, 2017. [Online]. doi: 10.1007/978-3-662-54455-6_8

[4] M. Rahouti, K. Xiog, and N. Ghani, “Bitcoin concepts, threats, and machine-learning security solutions,” *IEEE Access*, vol. 6, pp. 67189–67205, Nov. 2018. [Online]. doi: 10.1109/ACCESS.2018.2874539

[5] T. Pham and S. Lee, “Anomaly detection in bitcoin network using unsupervised learning methods,” arXiv, Feb. 2017. [Online]. Available: https://arxiv.org/pdf/1611.03941.pdf

[6] A. Bogner, “Seeing is understanding: anomaly detection in blockchains with visualized features,” in *Proc. of the 2017 ACM Intl. Joint Conf. on Pervasive and Ubiquitous Computing* and *Proc. of the 2017 ACM Intl. Symp. on Wearable Computers*, 2017. [Online]. doi: 10.1145/3123024.3123157

[7] K. M. Carter, R. P. Lippmann, and S. W. Boyer. “Temporally oblivious anomaly detection on large networks using functional peers,” in *IMC ‘10 Proc. of the 10th ACM SIGCOMM Conf. on Internet Measurement*, 2010. [Online]. doi: 10.1145/1879141.1879201

[8] Secretary of the Navy, “Cybersecurity readiness review,” Mar. 2019. [Online]. Available: https://www.navy.mil/strategic/CyberSecurityReview.pdf

[9] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” bitcoin.org, October 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[10] The Ethereum Wiki, “A next-generation smart contract and decentralized application platform,” accessed September 6, 2019. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[11] G. Destefanis, M. Marchesi, M. Ortu, R. Tonelli, A. Bracciali, and R. Hierons, "Smart contracts vulnerabilities: a call for blockchain software engineering?" in *Proc. of the 2018 Intl. Workshop on Blockchain Oriented Software Eng.*, 2018. [Online]. doi: 10.1109/IWBOSE.2018.8327567

[12] E. Bennett, "Ethereum attacks," GitHub Gist, accessed June 27, 2019. [Online]. Available: https://gist.github.com/ethanbennett/7396bf3f61dd985d3426f2ee184d8822

[13] R. R. OLeary, "Ethereum security lead: Hard fork required to release frozen parity funds," Coindesk, November 8, 2017. [Online]. Available: https://www.coindesk.com/ethereum-security-lead-hard-fork-required-to-release-frozen-parity-funds

[14] J. LeBoudec, *Performance Evaluation of Computer and Communication Systems*. Boca Raton, FL, USA: CRC Press, 2010.

[15] A. W. Bowman and A. Azzalini, *Applied Smoothing Techniques for Data Analysis*. New York, NY, USA: Oxford University Press Inc., 1997.

[16] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, England: Chapman and Hall/CRC Press, 1986.

[17] M. Tummala and C. Therrien, *Probability and Random Processes for Electrical and Computer Engineers*. Boca Raton, FL, USA: CRC Press, 2012.

[18] P. Prescott and A. T. Walden, "Maximum likelihood estimation of the parameters of the three-parameter generalized extreme-value distribution from censored samples," *J. of Statistical Computation and Simulation*, vol. 16, pp. 241–250, 1983. [Online]. Available: https://doi.org/10.1080/00949658308810625

[19] J. R. M. Hosking, J. R. Wallis, and E. F. Wood, "Estimation of the generalized extreme-value distribution by the method of probability-weighted moments," *Technometrics*, vol. 27, pp. 251–261, 1985. [Online]. doi: 10.1080/00401706.1985.10488049

[20] C. Reimann, P. Filzmoser, and R. G. Garrett. "Background and threshold: critical comparison of methods of determination." *Sci. of the Total Environment*, vol. 346, pp. 1–16, June 2010. [Online]. doi: 10.1016/j.scitotenv.2004.11.023

[21] A. K. Jain. "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, June 2010. [Online]. doi: 10.1016/j.patrec.2009.09.011

[22] J. MacQueen, “Some methods for classification and analysis of multivariate observations.” *Proc. Fifth Berkley Symp. on Math. Statist. and Prob.*, vol. 1, pp. 281–297, 1967. [Online]. Available: https://projecteuclid.org/euclid.bsmsp/1200512992

[23] L. K. Szeto, A. W. Liew, H. Yan, and S. Tang. “Gene expression data clustering and visualization based on binary hierarchical clustering framework,” *J. of Visual Languages and Computing*, vol. 14, pp. 341–362, Aug. 2003. [Online]. doi: 10.1016/S1045-926X(03)00033-8

[24] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, *Cluster Analysis*, 5th ed. Hoboken, NJ, USA: John Wiley and Sons, 2011.

[25] MathWorks. Natick, MA, USA. MATLAB, ver. R2019a. [Online]. Available: https://www.mathworks.com/help/stats/ dendrogram.html

[26] Etherchain, “Charts from the Ethereum network,” accessed February 18, 2019. [Online]. Available: https://www.etherchain.org/charts

[27] V. Kanth, “Blockchain for use in collaborative intrusion detection systems,” unpublished.

[28] V. Frias-Martinez, J. Sherrick, S.J. Stolfo, A.D. Keromytis, “Network access control mechanism based on behavior profiles,” in *2009 IEEE Annu. Compu. Security Applicat. Conf.*, 2009. [Online]. doi: 10.1109/ACSAC.2009.10

[29] M. Orcutt, “In 2019, blockchains will start to become boring,” MIT Technology Review, January 2, 2019. [Online]. Available: https://www.technology review.com/s/612687/in-2019-blockchains-will-start-to-become-boring/

[30] A. Gervais, and K. Wust. “Do you need a blockchain?,” Cryptology e-Print Archive of the International Association for Cryptologic Research, 2017/375. [Online]. Available: https://eprint.iacr.org/2017/

[31] J. McCarter, “Blockchain: Peer to peer naval operations,” Naval Innovation Advisory Council Report to the Office of Strategy and Innovation, 2017. [Online]. Available: https://portal.secnav.navy.mil/cop/NIN/NIAC/FY17/Blockchain.aspx

THIS PAGE INTENTIONALLY LEFT BLANK

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California