



# **NAVAL POSTGRADUATE SCHOOL**

**MONTEREY, CALIFORNIA**

## **THESIS**

**THROUGH THE LOOKING GLASS: CLASSIFYING  
ANOMALOUS BGP COMMUNITIES**

by

Josh Welch

September 2020

Thesis Advisor:

Co-Advisor:

Robert Beverly

Thomas J. Krenc

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

THIS PAGE INTENTIONALLY LEFT BLANK

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |  |                                                                 |                                            |                                                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|-----------------------------------------------------------------|--------------------------------------------|----------------------------------------------------------------|
| <b>REPORT DOCUMENTATION PAGE</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |  |                                                                 | <i>Form Approved OMB<br/>No. 0704-0188</i> |                                                                |
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.                                                         |  |                                                                 |                                            |                                                                |
| <b>1. AGENCY USE ONLY</b><br>(Leave blank)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |  | <b>2. REPORT DATE</b><br>September 2020                         |                                            | <b>3. REPORT TYPE AND DATES COVERED</b><br>Master's thesis     |
| <b>4. TITLE AND SUBTITLE</b><br>THROUGH THE LOOKING GLASS: CLASSIFYING ANOMALOUS BGP COMMUNITIES                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |  |                                                                 |                                            | <b>5. FUNDING NUMBERS</b>                                      |
| <b>6. AUTHOR(S)</b> Josh Welch                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |  |                                                                 |                                            |                                                                |
| <b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b><br>Naval Postgraduate School<br>Monterey, CA 93943-5000                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |  |                                                                 |                                            | <b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>                |
| <b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b><br>N/A                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |  |                                                                 |                                            | <b>10. SPONSORING / MONITORING AGENCY REPORT NUMBER</b>        |
| <b>11. SUPPLEMENTARY NOTES</b> 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.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |  |                                                                 |                                            |                                                                |
| <b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b><br>Approved for public release. Distribution is unlimited.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |  |                                                                 |                                            | <b>12b. DISTRIBUTION CODE</b><br>A                             |
| <b>13. ABSTRACT (maximum 200 words)</b><br><br>The Border Gateway Protocol (BGP) community field is poorly defined and has no means of authentication. This BGP attribute has the power to reroute and black hole traffic across the internet. The BGP communities' path attribute is normally prevalent and persistent. I hypothesize the persistence and prevalence of the path attribute can be used to develop a BGP community anomaly detector. This anomaly detector will allow for the discovery of and filtering out both malicious or misconfigured BGP community announcements. Future work could build upon the detection of anomaly and define the anomaly. In addition, an anomaly detector can be used to limit the spread and power of anomalous BGP communities. |  |                                                                 |                                            |                                                                |
| <b>14. SUBJECT TERMS</b><br>network, networking, BGP, routing, reroute, protocol, community, communities, internet, traffic, malicious, anomaly, detector, detection, machine learning                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |  |                                                                 |                                            | <b>15. NUMBER OF PAGES</b><br>103                              |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |  |                                                                 |                                            | <b>16. PRICE CODE</b>                                          |
| <b>17. SECURITY CLASSIFICATION OF REPORT</b><br>Unclassified                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |  | <b>18. SECURITY CLASSIFICATION OF THIS PAGE</b><br>Unclassified |                                            | <b>19. SECURITY CLASSIFICATION OF ABSTRACT</b><br>Unclassified |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |  |                                                                 |                                            | <b>20. LIMITATION OF ABSTRACT</b><br>UU                        |

THIS PAGE INTENTIONALLY LEFT BLANK

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

**THROUGH THE LOOKING GLASS:  
CLASSIFYING ANOMALOUS BGP COMMUNITIES**

Josh Welch  
Civilian, Scholarship For Service  
BS, California State University, Monterey Bay

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN COMPUTER SCIENCE**

from the

**NAVAL POSTGRADUATE SCHOOL  
September 2020**

Approved by: Robert Beverly  
Advisor

Thomas J. Krenc  
Co-Advisor

Gurminder Singh  
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

## **ABSTRACT**

The Border Gateway Protocol (BGP) community field is poorly defined and has no means of authentication. This BGP attribute has the power to reroute and black hole traffic across the internet. The BGP communities' path attribute is normally prevalent and persistent. I hypothesize the persistence and prevalence of the path attribute can be used to develop a BGP community anomaly detector. This anomaly detector will allow for the discovery of and filtering out both malicious or misconfigured BGP community announcements. Future work could build upon the detection of anomaly and define the anomaly. In addition, an anomaly detector can be used to limit the spread and power of anomalous BGP communities.

THIS PAGE INTENTIONALLY LEFT BLANK

---

---

# Table of Contents

---

|          |                                           |           |
|----------|-------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                       | <b>1</b>  |
| 1.1      | Motivation . . . . .                      | 2         |
| 1.2      | Scope . . . . .                           | 3         |
| 1.3      | Primary Research Questions . . . . .      | 4         |
| 1.4      | Summarized Findings . . . . .             | 4         |
| 1.5      | Thesis Structure. . . . .                 | 5         |
| <br>     |                                           |           |
| <b>2</b> | <b>Background Information</b>             | <b>7</b>  |
| 2.1      | Key Concepts and Terms . . . . .          | 7         |
| 2.2      | Related Work . . . . .                    | 14        |
| <br>     |                                           |           |
| <b>3</b> | <b>Methodology</b>                        | <b>23</b> |
| 3.1      | Data Input . . . . .                      | 23        |
| 3.2      | Pre-processing . . . . .                  | 25        |
| 3.3      | Feature Engineering . . . . .             | 27        |
| 3.4      | Machine Learning and Algorithms . . . . . | 48        |
| <br>     |                                           |           |
| <b>4</b> | <b>Analysis</b>                           | <b>53</b> |
| 4.1      | Algorithms. . . . .                       | 53        |
| 4.2      | Collectors and Time . . . . .             | 59        |
| <br>     |                                           |           |
| <b>5</b> | <b>Discussion</b>                         | <b>71</b> |
| 5.1      | Anomaly Analysis . . . . .                | 71        |
| 5.2      | Conclusion. . . . .                       | 77        |
| 5.3      | Future Work . . . . .                     | 79        |
| <br>     |                                           |           |
|          | <b>List of References</b>                 | <b>81</b> |
| <br>     |                                           |           |
|          | <b>Initial Distribution List</b>          | <b>85</b> |

THIS PAGE INTENTIONALLY LEFT BLANK

---



---

## List of Figures

---

|             |                                                                             |    |
|-------------|-----------------------------------------------------------------------------|----|
| Figure 2.1  | AS Path Prepending . . . . .                                                | 12 |
| Figure 3.1  | Looking Glass Update Frequency Comparison . . . . .                         | 27 |
| Figure 3.2  | CDF: Time Since Change . . . . .                                            | 33 |
| Figure 3.3  | CDF: Prevalence . . . . .                                                   | 35 |
| Figure 3.4  | CDF: Persistence . . . . .                                                  | 36 |
| Figure 3.5  | CDF: Foreign Communities . . . . .                                          | 37 |
| Figure 3.6  | CDF: Community Change . . . . .                                             | 41 |
| Figure 3.7  | CDF: Granularity of Change . . . . .                                        | 42 |
| Figure 3.8  | CDF: Community Length Change . . . . .                                      | 47 |
| Figure 4.1  | CDF: Algorithm Count by Looking Glass . . . . .                             | 55 |
| Figure 4.2  | CDF: Algorithm Count by Days . . . . .                                      | 56 |
| Figure 4.3  | Algorithm Time Comparison (Route-Views 2) . . . . .                         | 57 |
| Figure 4.4  | Cumulative Anomaly Count by Algorithm (Route-Views 2) . . .                 | 58 |
| Figure 4.5  | Cumulative Anomaly Count for Same Anomalies Found (Route-Views 2) . . . . . | 59 |
| Figure 4.6  | Looking Glass Anomaly Frequency Comparison . . . . .                        | 61 |
| Figure 4.7  | Day Comparison (Route-Views 2) . . . . .                                    | 62 |
| Figure 4.8  | Cumulative Anomaly Count by Looking Glass . . . . .                         | 64 |
| Figure 4.9  | Cumulative Anomaly Count by Looking Glass for Same Anomaly                  | 65 |
| Figure 4.10 | Cumulative Anomaly Count by Day . . . . .                                   | 67 |
| Figure 4.11 | Cumulative Anomaly Count by Day for Same Anomaly . . . . .                  | 67 |

Figure 4.12      CDF: Time Length Anomalies Persisted      . . . . . 69

---

---

## List of Tables

---

|           |                                                |    |
|-----------|------------------------------------------------|----|
| Table 3.1 | Looking Glass Input Data . . . . .             | 25 |
| Table 3.2 | Engineered Features . . . . .                  | 30 |
| Table 3.3 | Mean Values Looking Glass Comparison . . . . . | 31 |
| Table 3.4 | Seen Before . . . . .                          | 44 |
| Table 3.5 | Blackhole . . . . .                            | 45 |
| Table 3.6 | RIR . . . . .                                  | 48 |
| Table 4.1 | Algorithm Comparison . . . . .                 | 54 |
| Table 4.2 | Anomaly Counts . . . . .                       | 60 |
| Table 4.3 | Time Length Anomalies Persisted . . . . .      | 68 |
| Table 5.1 | Top 3 Unique Anomaly Comparison . . . . .      | 72 |

THIS PAGE INTENTIONALLY LEFT BLANK

---

## List of Acronyms and Abbreviations

---

|                    |                                     |
|--------------------|-------------------------------------|
| <b>ABOD</b>        | Angle-based Outlier Detector        |
| <b>AS</b>          | Autonomous System                   |
| <b>average KNN</b> | average K Nearest Neighbors         |
| <b>BGP</b>         | Border Gateway Protocol             |
| <b>CBLOF</b>       | Cluster-based Local Outlier Factor  |
| <b>CDF</b>         | Cumulative Distribution Function    |
| <b>CDN</b>         | Content Delivery Network            |
| <b>CIDR</b>        | Classless Inter-Domain Routing      |
| <b>CSU</b>         | California State University         |
| <b>CSV</b>         | Comma-Separated Values              |
| <b>DDoS</b>        | Distributed Denial of Service       |
| <b>DoS</b>         | Denial of Service                   |
| <b>GB</b>          | Gigabyte                            |
| <b>HBOS</b>        | Histogram-base Outlier Detection    |
| <b>HTTP</b>        | Hypertext Transfer Protocol         |
| <b>IANA</b>        | Internet Assigned Numbers Authority |
| <b>IP</b>          | Internet Protocol                   |
| <b>ISP</b>         | Internet Service Provider           |
| <b>IXP</b>         | Internet Exchange Point             |
| <b>KNN</b>         | K Nearest Neighbors                 |
| <b>LIR</b>         | Local Internet Registry             |
| <b>MRT</b>         | Multi-Threaded Routing Toolkit      |
| <b>NTT</b>         | Nippon Telegraph and Telephone      |
| <b>RIB</b>         | Routing Information Base            |

|            |                               |
|------------|-------------------------------|
| <b>RIR</b> | Regional Internet Registry    |
| <b>TCP</b> | Transmission Control Protocol |
| <b>UTC</b> | Coordinated Universal Time    |

---

## Acknowledgments

---

First, I would like to thank both of my advisors, Dr. Robert Beverly and Dr. Thomas Krenc. Their patience, feedback, and guidance was invaluable throughout the writing of this thesis. I would like to thank my family, friends, and classmates for supporting me along the way. This material is based upon activities supported by the National Science Foundation under Agreement No 1565443. Any opinions, findings, and conclusions or recommendations expressed are those of the author and do not necessarily reflect the views of the National Science Foundation.

THIS PAGE INTENTIONALLY LEFT BLANK

---

# CHAPTER 1:

## Introduction

---

The Internet that we use today relies on a protocol called the Border Gateway Protocol (BGP) that is over 30 years old [1]. BGP is the Internet’s inter-domain routing protocol. Often called the backbone of the Internet, BGP is used to implement policies between two Autonomous Systems (ASes). ASes use BGP to communicate available routes and to implement policy [2]. Many researchers have studied the protocol as it plays a vital role in how the Internet functions. An attribute of BGP that has seen recent research attention is BGP communities [3].

The BGP community attribute is defined in RFC 1997, however community values are largely undefined and unrestricted [3]. There is little formal specification regarding how the values should be implemented. In addition, the BGP community field has no standardized semantics leaving it up to AS operators how they wish to implement the attribute for that particular AS. Most importantly the BGP community attribute has no form of authentication, meaning both malicious and misconfigured BGP community announcements can potentially spread, and potentially have adverse effects. The “BGP Communities: Even more Worms in the Routing Can” paper has shown many providers transitively propagate Communities due to the lack of understanding the protocol [4].

The BGP community field was originally created to allow AS operators a convenient means to express or change policies on an aggregate of routes. Prior work has shown that malicious or miss-configured BGP community announcements can be used to reroute or black hole traffic as seen in the paper “BGP Communities: Even more Worms in the Routing Can” [4]. These announcements can propagate anywhere if the AS’s policy allows it. In the paper, Streibelt et al. show that if community announcements are not filtered properly they are visible beyond their intended scope. Thus, they can be misused by an AS and lead to unattended or malicious changes in routing or policy.

In this thesis, we will analyze the BGP update message, and will be looking for features indicative of an anomalous community in order to create an anomaly detector. We note that anomalies here are broadly defined, an anomalous BGP community in an announcement

might be due to 1) misconfiguration or an error on the AS's operators part; 2) some form of malicious activity; 3) a unique operation for a particular AS; or 4) some other reason that is unknown or unexplained.

## 1.1 Motivation

While BGP communities have seen some updates since its introduction over 20 years ago, its core function has stayed the same [5]. Little has been done to create a standardized structure across multiple ASes for the BGP community field [6]. There have been attempts to implement some form of authentication, but it will take years before it is widely used and still has its own set of flaws [7]. While authentication would yield some increase in a message's integrity, it does not address the problem of the optional transitive attribute of communities. BGP communities can be used by ASes for many different things including: rerouting traffic, black holing traffic, path prepending, geo-tagging, and many other uses.

Attacks or hijacking using BGP are a common occurrence due to the power BGP has and the lack of security being implemented. It is entirely up to AS operators how they wish to handle BGP community announcements and how those announcements will change their policy or continue to propagate. There are some communities that an AS understands and is defined to implement policy. There are other communities that an AS does not understand and will blindly continue to propagate.

This announcement holds the power to reroute or black hole any traffic that crosses the AS's control plane. An AS may continue to propagate an announcement passing the announcement to all of its peers. This whole process will start over again affecting every AS that listens and has a policy for that particular community string. Streibelt et al. [4] examine BGP community use over time, and discovered the use of communities has increased nearly 300 percent from 2010 to 2018. This increase in use is attributed to both individual ASes using more unique communities and more ASes starting to use communities. Streibelt et al. note that more than 75 percent of BGP announcements contain a community set. With this increase in use, there comes an increase of errors and malicious activity. AS operators are faced with a tough decision to either support BGP communities and take a risk network traffic may be routed incorrectly or ignore all BGP community announcements and lose the ability to fine tune routing adjustments.

More recently, Alex Hardt [8] was the first to examine BGP community behavior over time, where he utilized persistence and prevalence metrics. The persistence and prevalence metrics look at community change over time for a given prefix-ASPATH-peer tuple. Persistence is defined as: given a community for a prefix-ASPATH-peer tuple how much time will pass before that prefix-ASPATH-peer tuple experiences a community change. Whereas prevalence is defined as the probability when a community is seen for a particular prefix-ASPATH-peer tuple how likely are we to see the same community in the future for that same prefix-ASPATH-peer tuple.

In this thesis, we will build off of Hardt’s findings and create an unsupervised anomaly detector. Using the features persistence and prevalence as a starting point for an anomaly detector, we will engineer additional features in order to discover anomalous BGP community announcements.

## 1.2 Scope

We will be looking at different 24-hour periods using BGP updates for this thesis. We will focus on only announcements containing BGP community attributes.

We will start with one BGP collector to perform analysis on and try to determine what normal BGP communities look like. From here, we will perform feature engineering in order to identify potential BGP community anomalies.

Different unsupervised machine learning algorithms will be tested and results will be compared to find anomalous communities identified by multiple algorithms, in order to create an anomaly detector. Ground truth will pose a great challenge for validating results, anomalies identified will be manually validated by using publicly available information found online. Since all the data we are working with is unlabeled data, it will be difficult to tell how well an anomaly detector is working. We will perform unsupervised machine learning, using various algorithms that allow us to find outliers in order to find potential anomalies. Most results will be compared using mean values, along with personal intuition in order to validate announcements marked as anomalous.

Additional vantage points for the same 24-hour period will be used to test the anomaly detector on different data sets. By choosing the same 24-hour period, we hope to see the

same prefix announcement being flagged from different vantage points as anomalous. By doing this we are further validating our results are consistently being flagged as anomalous. It should be noted that our focus is on detecting anomalies and not identifying what caused the anomalous announcement.

Lastly, we want to know how long these anomalous announcements persist. How long before a normal announcement is seen after the anomalous announcement appears.

### **1.3 Primary Research Questions**

- Does unsupervised machine learning clustering algorithms discover anomalous BGP updates with qualitatively different communities and properties?
- How do different unsupervised machine learning clustering algorithms compare for the purpose of community anomaly detection?
- How does the vantage point influence the detection of anomalies?
- When do anomalies occur and how long do anomalies persist for?

### **1.4 Summarized Findings**

- We show that the persistence and prevalence scores for a given prefix are generally high. With a persistence score averaging 87.8% and prevalence score averaging 95.8% across four collectors.
- We notice that the communities associated with many prefixes stay constant for long periods of time and when the communities do change the change is small.
- We find that communities with a high number of foreign ASes are indicative of anomalous behavior. The average foreign percentage for all announcements 16.41% where anomalous community announcements averaged 38.37%.
- We show the number of community sets is generally less than ten where announcements with 15 or more sets are often anomalous.
- We show that anomalous announcements can be seen and detecting from multiple vantage points.
- We show the average anomaly persists for over 30 minutes.

## 1.5 Thesis Structure

- Chapter 1: Introduction and Motivation  
Introduces thesis topic and provides information to its importance.
- Chapter 2: Background Information  
Explains key topics and terms in order to better understand the thesis. Discusses prior and related work done with BGP communities.
- Chapter 3: Methodology  
Visualize our data and how we define how normal communities behave. Discuss which features are good indicators for discovering anomalies and which features should be removed.
- Chapter 4: Analysis  
Compares results from different algorithms and answers key questions we have regarding anomalies.
- Chapter 5: Discussion  
Deep dives into the top anomalies found, and concludes thesis with recommendations for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

---

## CHAPTER 2:

# Background Information

---

In this chapter, we will cover background information including key concepts and prior work that has been done with BGP communities. Understanding these key concepts and terms first will prove helpful in understanding the rest of this thesis.

## 2.1 Key Concepts and Terms

This section will prove useful in understanding terms and concepts used throughout the paper. The bulk of the work done will be discussing BGP communities, but it is only one cog in the complex machine that is the Internet. In order to understand how BGP communities function, one must first understand how BGP functions.

### 2.1.1 BGP

The Border Gateway Protocol (BGP) is an inter-AS routing protocol [1]. The main purpose of BGP is for ASes to communicate reachability information between each other. Essentially this route information tells ASes where to reach different IP addresses. BGP creates a partial map for each AS where to send traffic depending on the IP address. Another way to think of it, is an AS is like a post-office where BGP is the way post-offices communicate what zip codes are in their delivery area. They can also tell them what other post-offices they are connected to. This allows each AS or post-office to know where to forward a packet or letter in order to get the packet closer to its destination.

Every prefix has some route information from BGP, otherwise it would not be reachable across the Internet. A prefix is specified block of Internet Protocol (IP) addresses, later in this chapter we will discuss in-depth what a prefix is. Many prefixes have multiple paths they can be connected through. Depending on policy, traffic may not always take the shortest path.

BGP comes in two forms, iBGP and eBGP. Where iBGP is internal BGP, this is used to share route information between routers in a single AS. eBGP on the other hand is used externally, where route information is shared between routers in two different ASes. This

thesis will focus on eBGP and when referring to BGP messages we are referring to messages between two different ASes.

All BGP messages use Transmission Control Protocol (TCP) on port 179 to communicate between ASes. There are four different types of BGP messages. [X] BGP OPEN messages are used to set up and establish BGP communication between ASes. A BGP UPDATE message allows peers to advertise routes, update routes, or withdraw routes. The NOTIFICATION message lets a peer know there was an error in a previously received UPDATE message. Lastly, the KEEPALIVE message is used to maintain liveness of the BGP connection. If an UPDATE or KEEPALIVE message is not received for a specified time length, the connection can timeout and close.

In this thesis, we focus on BGP UPDATE messages. The UPDATE messages contain the BGP community field, which is what we will be focusing on. We will use the term “announcement” as an UPDATE message received by an AS.

BGP messages have optional attributes that can be both transitive and non-transitive [2]. It is not required for ASes to recognize all optional attributes, it is up to the AS whether or not to implement its use. Optional transitive attributes have a partial bit in the attributes flag set to one. This tells the AS it can pass this message along with the optional attribute intact to its peers. The community and aggregator are both optional transitive attributes for ASes to communicate with all of its peers. Optional non-transitive attributes have the partial bit in the attributes flag set to zero. This means the AS can silently ignore and drop the message if it is unrecognized. Or strip out the optional non-transitive attribute before sending it to its peers. Examples of optional non-transitive attributes include: originator ID, cluster list, and Multi Exit Discriminator.

### **2.1.2 Prefix**

A prefix is a block of IP Addresses, which are often shown in Classless Inter-Domain Routing (CIDR) notation. For simplicity, we will only be looking at IPv4 addresses, which are a 32-bit value containing nearly 4.3 billion addresses. A single IPv4 address could look like 205.15.7.3, representing a single host on the Internet. A prefix in CIDR notation could be 205.15.7.0/24, where the /24 represents how many bits are contained in the network alternatively stated how large of a chunk it is [9]. For 205.15.7.0/24, there are a total of 256

addresses available, 205.15.7.0 - 205.15.7.255 with 254 available addresses for hosts. The smaller the /n where n is the network mask the larger the chunk, meaning more available IP addresses. The larger the /n the smaller the chunk, the less available IP address. A /30 would only yield four available IP addresses, where a /8 would mean 16,777,216 available IP addresses.

Prefixes are a collection of IP addresses in CIDR notation. An AS will announce paths to prefixes within their network or other networks they are connected to.

### **2.1.3 AS**

An Autonomous System (AS) is a collection of routers and prefixes under one routing policy. Each AS can be using its own unique routing protocol within its network, but when talking to other ASes it must use the BGP protocol to communicate route information to other ASes [10]. An AS can be run under one or more administrators working under the same routing policy. AT&T and Comcast are examples of Internet Service Providers (ISPs) ASes, while in this paper, we will look at ASes like Nippon Telegraph and Telephone (NTT) and California State University (CSU). Aside from ISPs, an AS can be a Content Delivery Network (CDN), data center, search engine such as google, or business.

Every AS has its own unique identifying AS number. AS numbers were originally created as a 16-bit value but as the number of ASes has increased, the need for more AS numbers led to the creation of 32-bit AS numbers. An AS number not only identifies which AS owns that number it also identifies which region it is from. Internet Assigned Numbers Authority (IANA) delegates resource identifiers like AS numbers and IP addresses, for distribution to different Regional Internet Registries (RIRs) to distribute to a specific region in the world. Depending on the region this can be further delegated to Local Internet Registries (LIRs) [11]. IANA is a U.S. based non-profit that is in charge of distributing various Internet numbers for the entire globe. For this paper, the majority of our results contain 16-bit AS numbers, for example AS2914 is NTT and AS2152 is CSU.

ASes are selective to which ASes they announce their prefixes to. This decision is not done lightly, as it may have economic impact or impact their customers reachability. An AS will announce prefixes to its customers as it is being paid by its customers. An AS might also have an agreement with certain ASes it peers with to announce routes as a mutual agreement

that benefits both ASes. On the other hand an AS might choose not to advertise to other ASes or providers as it would be costly and not benefit that particular AS.

#### **2.1.4 BGP Communities**

The BGP community field is an optional transitive attribute within BGP UPDATE messages. RFC 1997 defines the community attribute as a means to group destinations that share some common property [3]. Each AS is free to define which communities it belongs to and how it will affect its own policy or routing. AS operators are given an optional transitive attribute, with a field of variable length to implement how they best see fit. The attribute is a set of individual communities, where any AS can manipulate the community set however they choose. An AS is free to add communities, remove communities, change existing communities, or filter out all communities completely depending on what policies an AS has put into place. There are no rules or regulations mandating use of the community field; it is entirely up to each AS how they wish to use or not use the attribute.

BGP Communities was first implemented for use with 16-bit ASes. Later came Extended BGP communities that had 8 octets in total to allow for the support 32-bit AS numbers [5]. In addition, to support 32-bit ASes, the first two octets are used to represent types and sub types. This type field indicates how to partition and read the remaining octets. This additional complexity allows for greater use of extended BGP communities. In this paper, we will only be focusing on 32-bit BGP Communities as the rapid increase of use and anomalies are not fully understood.

Each individual community in the set is a 32-bit value. The first two octets are typically used to represent the AS number, it is known that it is not always the case and will be discussed later in this paper. The last two octets or 16-bits set and implemented by the AS. This number has a unique meaning to each AS and may or may not implement some policy or route change for a given value.

Communities can be used for many different purposes, a couple of commonly recognized types of communities are inbound communities and signaling communities. Inbound communities are used for route tagging with information such as geolocation where the route originated for a given prefix. Signaling communities are prefixes tagged with a community to tell an AS downstream to perform some operation or policy change. In the related work

section of Chapter 2, we will discuss in detail these and other types of communities.

The BGP community attribute, was designed to give more power to the ASes, to fine tune policy and route changes. Then to quickly and easily communicate these changes with other ASes. An example how this could be implemented would be a prefix within a certain AS getting Distributed Denial of Service (DDoS) attack. The AS may be overwhelmed with traffic destined to this prefix. Instead of having the AS operator try to call its peers to stop traffic to that particular prefix, the operator could send a BGP community message. This BGP community message would have the prefix that is being DDoS in its message and in the community field a code that would blackhole or drop any packets with the prefix as the destination. This would stop all the traffic to that prefix allowing the AS to get things back in control and maintain Internet service for the rest of its customers, instead of slowing down traffic or losing connectivity.

As an example, the BGP community value for NTT starts with “2914:X”, where 2914 is NTT’s AS number and the X is some community value. NTT is one of the few ASes that lists all community numbers and their meanings online [12]. An example is “2914:460” which is peer backup. A peer backup is used to advertise alternative or backup routes to avoid loss of connectivity [13]. If a router or cable was to go down causing loss of connectivity, an AS could immediately start sending traffic through this backup route to avoid complete loss of connection. Another example of a community is “2914:1011” that tells the route origin is from Boston, MA. A common community string among multiple ASes is “666”, if NTT was to send a BGP UPDATE message with “2914:666”, all peers with the correct policy would blackhole or drop packets to a given prefix. All BGP UPDATE messages with a community attribute only apply to the given prefix supplied in the UPDATE message.

The term blackhole is widely accepted, but the exact communities used to communicate a blackhole varies between ASes. While many ASes use “x:666” where x is the AS number to blackhole traffic for an advertised prefix, other ASes might have a different number for blackhole. For example AS3356 uses “3356:666” to communicate peer route, where AS3257 uses the community “3257:2666” to communicate blackhole. In this thesis we will treat the community “x:666” as a possible blackhole and potentially anomalous activity. Blackhole communities are very powerful so anytime one is being used it should be investigated and could be the sign of something malicious or possible misconfiguration.

### 2.1.5 AS Path

The term AS path is not only commonly used to describe how to reach a given prefix, but a field within a BGP Announcement. AS paths are used both as metric for determining the path length as well as used for loop detection. An AS path is a set of AS numbers with the last AS number being the AS that advertises that prefix. This advertising AS may be the legitimate owner of the prefix, or may be malicious attempt to hijack the prefix. Using the given AS path in Figure 2.1, “2117 1594 3863”, AS2117 is the first hop to reach a given prefix. A “hop” is used to describe how many ASes must be traversed to reach a given prefix. If we describe this AS path in hops, we would say it is three hops. Any AS connected to AS2117 would prepend their AS number and pass the message along to other ASes it connects with as a way to reach a given prefix. In Figure 2.1, NTT AS2914 is connected to AS2117, if NTT wanted to announce this route to its peers it would perform what is known as path prepending. NTT would prepend AS2914 to the AS path and make an announcement how to reach a prefix through NTT. The new path would now be “2914 2117 1594 3864”, where an AS connected to NTT would now go to NTT as the first hop and AS3864 would be the last hop to reach a prefix. Using path prepending a route can be made less appealing in the path selection process, e.g., setting up a backup route.

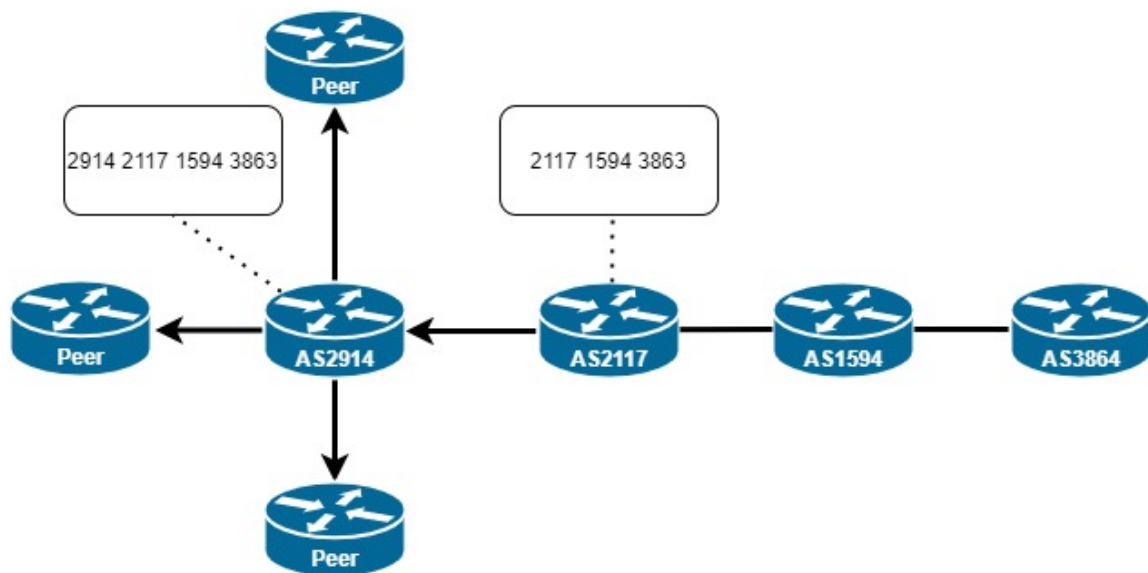


Figure 2.1. AS Path Prepending

### **2.1.6 Peer**

A peer is used to describe what ASes an individual AS is directly connected to. When using the phrase directly connected to, we are not necessarily implying one cable from one router to another. An AS connection might still have to traverse many routers and may have multiple connections to another AS. Directly connected means an AS does not have to go through another AS's network in order to reach its peer. NTT lists they have 65 peers they are directly connected, not including over 1,600 customers they are connected with as well [14]. AT&T is listed as one of those peers. If NTT receives an announcement of a route to a prefix through AT&T, NTT may announce this to all of its other peers by prepending its AS number on the AS path received from AT&T. Depending on policies this peer announcement may continue to propagate or stop right at NTT.

In this thesis we will discuss looking glass peers as where an BGP community announcement came from, but this is not to say where it originated. If NTT receives a BGP community announcement from AT&T, it is possible that the announcement came from any AS on the AS path and it just so happens NTT received through AT&T. This is an important distinction that must be made, when an BGP community announcement is received there may be some intuition from where it came from, but it is unknown who first added the community.

### **2.1.7 Looking Glass**

A tool commonly used to view BGP announcements are BGP looking glasses. A looking glass, also referred to as a route collector, is a software tool for users to remotely access BGP tables and updates from a supporting AS. A looking glass can support two functions, the first is a live tool that allows users to connect directly to a router via telnet or Hypertext Transfer Protocol (HTTP). Once connected, users of the looking glass can query the current routing states and view current BGP tables stored in that router. The second function a looking glass may support is an archive of various snapshots. These snapshots come in two forms Routing Information Bases (RIBs) and Updates. RIB contain snapshots of the router state at set increments, for example Routeviews.org captures ribs every 2 hours [15]. Updates contain the actual update messages a router receives in a given time period, Routeviews.org has Updates available in 15-minute increments. A looking glass is an optional tool that only some ASes support in order to help other ASes trouble shoot problems, or issues an AS may be having with a given prefix or route. It is also widely used for research

purposes, studying changes and used for measurement data. These looking glasses can be geographically located anywhere in the world. Each looking glass offers a unique vantage point from one another based on where and what peers they are connected to. A looking glass can be used to view changes in BGP communities and will be the primary source of data used to build an anomaly detector.

In this thesis we will be examining archived update data sourced from Routeviews.org. Route views offers multiple looking glasses with archived data that goes back many years for many looking glasses. Route views began archiving data in 1997 and has continued this archiving process through present day. Update data from Routeviews.org is available in fifteen minute time intervals to download in the form of a Multi-Threaded Routing Toolkit (MRT) file. These MRT files are raw dumps of BGP UPDATE messages for a certain time period. These files must then be parsed with special tools before being read. We will be looking at 24 hours of data from multiple vantage points for feature engineering and anomaly detection. In addition, we will also look at different 24 hour periods for the same vantage points to help draw and verify conclusions from the data.

### **2.1.8 Machine Learning**

Machine learning will be used in this paper to predict which community announcements are anomalous. This is done by using various algorithms to find outliers in our community announcements data set. We will be performing unsupervised machine learning, this is the process of using unlabeled data to make predictions on. In essence our algorithms will use features we engineer to create clusters. From these clusters, we will identify outliers and these outliers are classified as anomalies. In the methodology section of Chapter 3, we will describe in detail which algorithms we used and how those algorithms work.

## **2.2 Related Work**

### **2.2.1 Early Work**

In an early paper a few years after BGP community attribute was added, the authors Gao, Griffen and Rexford talk about a possible use for BGP communities. Gao et al. discuss implementing policies without modifying BGP, but using BGP communities [16]. They go on to discuss using BGP communities to pass local preference values between peers

and filtering. By assigning local preference values, they can ensure they take care of their customers first and avoid routes that are less economical. Furthermore, they mention using “a special community value could be employed between peers to signal that a route was received from an upstream provider” [16]. Values are assigned in a way that ensure preference for customers over peers and providers. In essence if an AS utilizes the Community value in this way their customers will have a better experience and they will save money by not using certain routes or preventing traffic to traverse through its network to reach certain peers.

In 2001, little was known about the uses for BGP communities. Gao et al.’s ideas for BGP Communities are very similar to how many ASes implement using them today. While the exact schema varies from AS to AS, the concept is the same. One of the way we see BGP communities being used today is setting local preferences and filtering being done. Many ASes also have a community for route origin, letting its peers know where a prefix originated from. This early work, gives us an idea what BGP communities can be used for and what they may look like.

Benoit Donnet and Olivier Bonaventure, take three years of BGP dump tables and measure the use of BGP communities by both the number of ASes and frequency per AS [6]. Additionally, in their paper they propose a BGP community taxonomy for better helping AS operators understand and use BGP communities. Donnet and Bonaventure looked at BGP data from September 2004 to September 2007. They discovered on average more than half of the routes from each vantage point contained a BGP community. In addition, they found that both the number of unique communities had increased and the number of ASes using communities had increased over the three year period. From all but one vantage point their graphs show nearly triple the increase in the number of unique communities being used and triple the number of ASes using communities over three years. These results show a rapid expansion of more unique BGP communities and following the same trend the number of ASes using the BGP community attribute.

Donnet and Bonaventure discuss the taxonomy of BGP communities and define three categories of BGP communities. This is an important discussion in order to better understand the behavior and use of BGP communities. The first category of classification they describe is inbound communities. Inbound communities refers to when a community is added or was received by another AS. They further break down inbound communities into two different

categories; “LocalPref and Routes tagging”. LocalPref or local preference is used to set the “Local Pref” in the AS receiving the route announcement. The example given by Donnet and Bonaventure is for AS174: “174:10 sets the Local Pref to 10”. Where ten represents a local preference value that AS174 recognizes in its policy. The other part of inbound communities is Routes tagging, this is used to indicate where a route was received from the peer AS.

Routes tagging is broken down further into the four most common types; Internet Exchange Point (IXP), Type of Peer, Geographic, and AS. An IXP tagging is interconnection point where the route was learned. The example given: “4589:14901” where the route was learned by AS4589. The Type of Peer category, communicates the relationship between the AS and a particular peer, which eASes router filtering. The relationship types are: “peer, customer, provider, or transit”. The third type of tagging Geographic, indicates the geographic location where a route was received. This can be precise as a city or as vague as an entire continent. Using an example mentioned early “2914:1011” is NTT’s way indicating a route was received from Boston, MA. The last type of tagging discussed is AS, which is simply naming the AS the route was received from. Where the community given specifies a certain AS, but this is not to say the community and the AS share the same number. The example given “15997:1080”, AS15997 is stating the route was received from TELEGLOBE, where TELEGLOBE’s AS number is AS6453 not AS1080.

The second category of classification Donnet and Bonaventure describe in their taxonomy is outbound communities. Outbound communities are used by an AS for traffic engineering, by influencing peers or even further down stream when a route is announced. The AS that is the originator of a route inserts the outbound community into the announcement. This outbound community influences the redistribution of routes by the peers and further down stream. Donnet and Bonaventure break down this redistribution into two sub-classes; “Announcement and AS Path prepending”. An “Announcement” indicates whether or not a route should be advertised to a certain peer or further upstream. The example given: “8938:4000” indicates AS8938 does not want the announcement advertised to any upstream providers. AS path prepending allows an AS to tell its peers to prepend its AS number  $n$  times when announcing the route. This could be to a specific peer or all peers. An AS1234 could send out “1234:1 $n$ ” where  $n$  is the number of times to prepend to the AS path and “where  $1 \leq n \leq 4$ ”. Let’s say  $n$  is 3 and the AS path is “1234 2117 1594 3863”, AS5678

receives the message from AS1234. Before AS5678 announces this route it first prepends its AS number to the path. The new AS path looks like “5678 5678 5678 1234 2117 1594 3863”. This route now has AS path is now artificially increased in length making it look less appealing.

The final category of classification for BGP communities is called blackhole. Blackhole communities are used to by an AS to block packets destined for certain prefixes. Donnet and Bonaventure describe three different types of blackholes. Global blackholing, is where packets are blocked or dropped by all routers within an AS. Where border blackholing is used to block packets at only the border routers for an AS. Lastly, we have upstream blackholing where “packets are blocked by the ISP’s transit provider before they enter its own network”. Blackholes are a powerful tools used to minimize the damage of a Denial of Service (DoS) attack. Donnet and Bonaventure explain blackhole communities should only be used internally and should not traverse the global Internet, in another paper we will discuss this is not always the case.

### **2.2.2 Recent Work**

A more recent paper “BGP Communities: Even more Worms in the Routing Can” by Florian Streibelt, Franziska Lichtblau, Robert Beverly, Anja Feldmann, Cristel Pelsser, Georgios Smaragdakis, and Randy Bush [4], looks at how BGP communities can propagate globally and the effects a rogue BGP community announcement can have on different ASes. Most of the use cases for BGP communities is communication with directly linked peers, therefore BGP communities messages need not to continue propagating. Streibelt et al. discovered more than half of the community announcements traversed more than four hops and with 10 percent traversing over six hops [4]. This indicates that a misconfigured or malicious BGP announcement could potentially propagate on a global scale. A malicious actor would not need to be peering with an AS it is trying to influence and could remotely trigger a policy change from a connection much farther away.

Streibelt et al. use historical BGP data to find the number of unique communities being used per year. They discover that from 2010 to 2018 there is nearly triple the amount of unique communities being used, a 296 percent increase to be exact. This increase in use can be attributed to both more ASes starting to use BGP communities and individual ASes using

communities more frequently. With this increase in use raises concerns of an increase in BGP communities use as an attack vector.

Streibelt et al. continues to discuss how communities can easily propagate globally. They find that even though only “14 percent of transit providers are propagating received communities to their peers,” ASes are so interconnected with many peers communities continue to propagate much further than one would expect [4]. The researchers look at how communities can be leaked to unintended ASes. He refers to this leak as on-path vs off-path. On-path is when a community has an AS number that is on the AS path. For example the community “456:666” an AS path of “123 456 789” is an on-path community, because AS456 is on the AS path. Where the community “1011:666” with the same AS path “123 456 789” is considered off-path, because AS1011 is not on the AS path. This is unusual as the intended target for the community AS1011 is on the path, meaning an AS on the path must have passed this community onto its peers without any filtering. While this may not seem like a big deal, this “leaking community” might have unintended consequences for another AS. This also opens the door for an attack vector to leak specific communities for some desired impact. Streibelt et al. look at some common off-path communities, “666” being one of them. One possible theory why “666” is being propagated off path as ASes may not have a policy for “666” and just pass it along to all of that AS’s peers instead of filtering or removing the community. Later in this thesis, we will look more in-depth into off-path communities and refer to them as foreign communities. Foreign communities could be used as an indicator for potentially anomalous activity.

Streibelt et al. highlighted one of the properties of BGP communities is “Communities can be added, deleted, or set by an AS on prefix origination, ingress, or egress.” [4]. This property of BGP communities can easily be misused both unintentionally and maliciously. This makes it especially difficult to understand who created or added a community in a message as any AS on the AS path could be responsible. This also adds to the difficulty of validating both legitimate and anomalous BGP communities.

In Streibelt’s et al. research, they discovered mixed results for BGP community filtering. Depending on the AS and the number of peers an AS may perform some or no filtering at all. After speaking with the operator community, they discovered nearly every AS has a different perspective on filtering. Some ASes may strip all of the BGP communities before

passing on the route while others will leave the communities untouched. An AS might also selectively choose which peers to pass the route onto with the BGP communities. The big point of this, is there is no general consensus on what to do with BGP communities. Also since most ASes have many peers it only takes a few peers to pass the communities on untouched for them to globally propagate.

The transitive nature of BGP communities allows for a “can of worms” of possible remote attacks. Streibelt et al. discuss and demonstrate three possible remote attacks; blackholing, traffic steering, and route manipulation. An attacker can be anywhere on the AS path and selectively choose an AS to perform blackholing that will prevent traffic going to a given prefix. An attacker can prepend the blackhole community “1234:666” to a route announcement and AS1234 will now drop traffic for the designated prefix. While blackholes are designed to limit the effect of a DoS attack, when used maliciously it is now the means to a DoS attack.

Traffic steering can be performed by two different ways, AS path prepending and local preference. An attacker can use AS path prepending to trigger another AS to add its AS number to a path  $n$  times in order to make its path appear longer. For example “1234:1 $n$ ” where  $n$  is the number of times for AS1234 to prepend to the AS path. Generally traffic will flow to the shorter route, once a route is advertised with this artificially extended AS path the traffic will start flowing to a different shorter route. Local preference sets a given prefix announcement to either be locally preferred or not depending on how the attacker wants to move the flow of traffic. This type of attack could be used for many reasons by steering the flow of traffic. One might be to gain visibility of the traffic in order to gain intelligence or steal information. Another possible advantage with this attack is to then perform a Man-in-the-Middle attack. Now that the traffic is flowing in the right way an attacker could pretend to be the intended destination and again gain intelligence and steal information. Lastly this type of attack could lead to financial advantage having a provider make more money by the flow of traffic or an ISP to incur additional fees.

Route manipulation uses communities with IXPs to trick the IXP to not advertise certain routes. By doing so traffic and route information may never reach its target. When an AS sends a community to an IXP to share route information with its peers. Then a malicious announcement from a different AS connected to the IXP can tell the IXP “not advertise a

prefix to a peer AS” This manipulation is a combination of partial blackholing and traffic steering leading to similar outcomes that are advantageous to an attacker. For all of these attacks the proper conditions must be in place, but Streibelt et al. were able to show these conditions could easily be met and given enough time and resources an attacker could potentially attack any AS remotely from almost anywhere in the world.

Recent work by Alex Hardt’s research looks at BGP community behavior over a years’ worth of time using multiple looking glasses [8]. Hardt looks at characteristics the majority of BGP communities have in order to find what we would consider “normal”. Hardt uses a years’ worth of data from four different vantage points this helps validate what he is seeing and ensures his measurements are not looking glass or time frame specific. Two major characteristics of BGP community behavior Hardt describes as persistence and prevalence. The majority of BGP communities have a high persistence and prevalence score which we will consider as normal behavior. Thus low persistence and prevalence score is uncommon and abnormal. This abnormal community could potentially be an anomaly and because of its rarity we hope by tagging this as an anomaly we will not have a ton of false positives.

To better understand these two characteristics we will look at a couple of examples. An example of persistence given ten BGP community announcements for the same prefix, we ask the question what is the longest streak where the announcements were the same? Say we saw three the same, then a change, then five the same then another change. Five is the longest streak out of the ten BGP announcements, the persistence score would be 5/10 or 50 percent. Where an example of prevalence with the same ten BGP announcements, out of those ten, eight had the same communities. Thus the prevalence score would be 8/10 or 80 percent. These persistence and prevalence score allows us to see how much the BGP communities for a given prefix are changing over time. Hardt discovered that the majority of BGP communities change very little, which leads to fairly high persistence and prevalence scores.

Hardt’s work discusses an important concept that must be highlighted when looking at BGP communities the term prefix-ASPATH-peer tuple. When looking at how BGP communities change or stay constant over time we must ensure we are not only looking from the same vantage point, but that the prefix, AS path, and peer is the same. If any part of this tuple changes we consider this to be a different BGP announcement and cannot compare the

communities for change or consistency. If the prefix changes, then we are referring to a different block of IP addresses. If the AS path changes, then the announcement may be coming from a different direction and any of the ASes on this new path could be the cause for the changes. If the peer was to change, then our collector received the BGP announcement from a different AS and could be the reason behind any changes found in the community field.

Another notable discovery in Hardt's research is the length of the community field does not change significantly for a given prefix-ASPATH-peer tuple. The length of a community field refers to the number of entries or individual communities. For "123:1008 456:17 789:666" the community has a length of 3, we would expect this to go up and down 1 or 2 entries not to jump to a number larger 10 then back down. Additionally, Hardt found "the size of the change is virtually always less than 10 communities" [8]. This suggests that if a community is having large changes in the length of the community it could be potentially anomalous and that we would relatively few anomalies to look at which makes this a viable for feature to have in an anomaly detector. Hardt continues to note that "99 percent of all prefix tuples use less than 10 unique community path," which means the length itself, not just the change in length could also be an indicator of an anomaly.

Hardt discusses the use of foreign ASes in the community path and while it is an irregularity, it varies between vantage points. Thus, its uses as a feature for an anomaly detector may or may not be beneficial. This foreign AS could just be to a flaw within BGP's implementation. As it is an unusual artifact, in this thesis we will look at its use as a feature for different looking glasses to see how beneficial it is.

Much of this thesis will build off of Hardt's discoveries to define BGP community behavior. Hardt's research points out that communities use varies from vantage point to vantage point. This suggests we must train and test on the same vantage point in order to get the best results from an anomaly detector. Hardt's work utilized a large time frame and multiple vantage points to find how the majority of BGP communities behave. We will use this as our starting point with the assumption that other vantage points from different time periods will have similar results.

THIS PAGE INTENTIONALLY LEFT BLANK

---

## CHAPTER 3: Methodology

---

In this chapter, we discuss our methodology and process to detect BGP updates with an anomalous community set. The methodology discussion is broken into the following sections: 1) input data; 2) pre-processing; 3) feature engineering; and 4) machine learning and algorithms.

### 3.1 Data Input

BGP data is sourced from the University of Oregon Route Views Archive Project [15]. Route Views was originally created in 1995 as a tool for AS operators to trouble shoot in real-time. Now Route Views is widely used for research purposes with over 30 collectors around the world and archival data going back over 15 years for some of the collectors. By using Route Views, we are able to choose archived data from multiple looking glasses for the same period. The first looking glass we looked at was Route Views 2 Oregon-ix, this is the first looking glass listed in the archive and contained a large collection of updates. We arbitrarily chose BGP update data from a non-holiday date from the beginning of 2020, January 2, 2020, as input to our anomaly detector. Using January 2 gave us a fairly recent look at BGP communities, but far enough back we could slowly look at more recent dates for comparison.

We use update data instead of RIB data; this allows us to see a finer granularity of change. RIB files are two hour snapshots of route tables, if a change occurred within that two hours and then went back to its previous community we would not be able to see it. By using updates, we see every update message the collector receives. BGP update files are broken into 15-minute intervals and stored in MRT format.

Keeping time constraints and data size in mind we use 24 hours (midnight to midnight) of BGP update data. In addition to this we decided to add the last 15 minutes from the day prior to capture any day-to-day changes; this totaled 97 MRT files. Depending on the looking glass, these 97 MRT files in total ranged in size from 1 to 3 Gigabytes (GBs).

We use the following four looking glasses: Route Views 2, Route Views 3 and Route Views 4 apart of the University of Oregon, and Route Views AMS-IX from Amsterdam's IX. From all four vantage points we looked at the same time period to see how inferred anomalies differed or agreed between multiple vantage points, as well as to test our anomaly detector on different looking glasses as the data sets would vary.

Additionally, we look at different days from the same looking glass for comparison. For Route Views 2 we look at the following four days (January 3 - 6) of update files. We did this to see if there any persisting anomalies existed, and if we could see any patterns at what time anomalies occurred. All five days averaged 8.3 million announcements per day that contained communities as seen in Table 3.1. Knowing that BGP community use has increased rapidly over the past few years we also decided to look at update data from 2016. By looking at older data we knew we would have fewer BGP community updates and expect to see fewer anomalies. The number of updates seen for 2016 was roughly half of what we see today, about 4.2 million updates for a 24-hour period. It will be useful to compare these anomalies to see if the anomalies have changed in the past four years, or if we are seeing similar prefixes and community announcements being classified as an anomaly.

All times referred to in this thesis are in Coordinated Universal Time (UTC) time. This is used across all looking glasses to prevent any time zone differences when comparing data across looking glasses. Table 3.1 will be used as a reference throughout this thesis for total number of updates, as many of our measurements utilize this number. We can see in Table 3.1 that the number of unique communities is a fraction of a percent of the total number of announcements. This means we are seeing many announcements using the same communities over and over. Which gives us hope that we can later define normal communities.

Table 3.1. Looking Glass Input Data

| Looking Glass | Total #<br>MRT Files | Time Frame      | Announcements   | Unique Communities |
|---------------|----------------------|-----------------|-----------------|--------------------|
| RouteViews 2  | 582                  | 1/2/20 - 1/6/20 | 8,323,269 (avg) | 68,852 (avg)       |
|               |                      | 1/2/16          | 4,155,373       | 35,937             |
| RouteViews 3  | 97                   | 1/2/20          | 16,818,673      | 161,940            |
| RouteViews 4  | 97                   | 1/2/20          | 6,063,154       | 47,400             |
| AMS-IX        | 97                   | 1/2/20          | 4,848,228       | 42,425             |

## 3.2 Pre-processing

Each MRT file comes compressed using BZIP2. The file size can vary greatly depending on the looking glass used and time of day. The file size correlates directly with the number of updates that are received in that particular 15-minute time period.

Once uncompressed, each MRT file is parsed to filter for only update messages that contain communities. On average 75 percent of messages contain a community announcement as mentioned in Chapter 1. We use a tool called “BGP Scanner,” that quickly filters and parses MRT files and converts to a regular textual output [17]. The majority of announcements contain IPv4 prefixes with a small percentage being IPv6, this is because of the looking glasses we used. Thus, we primarily focus on IPv4 prefixes in this thesis.

After each MRT file is parsed and converted, we then use a custom script written in C++ to clean and combine all the text files into a single Comma-Separated Values (CSV). Python was initially chosen for its ease in coding, but C++ was chosen over python because of the magnitudes in speed increase for pre-processing. Each update message in the text file is converted to a row in a table, this allows us later to easily query and graph the data. From each message the following Columns were created: Type, Prefix, AS Path, AS Path Length, Peer IP, Peer AS, Communities, Community Length, Community String, and Time Stamp. Prefix is the IP prefix to which the message corresponds. The AS Path is how to reach that prefix, and the AS path length is how many ASes are in the AS Path. Peer IP and Peer AS is what AS passed the message to the collector. Communities is everything in the community

field of the message, and Community length is how many individual communities are in the message. Community String is used to examine the last two octets of each individual community in the community set, by looking at what comes after the “:” for each community in the set. Last is the time stamp when the message was received at the collector. All the individual messages from each of the text files are combined into one CSV file and sorted by time.

We use Python to analyze, graph and eventually perform machine learning. While python is not as fast as C++, it has many libraries available that allow all of the processing to be done with relative ease. In order to quickly access, view, and query the data in python we convert the CSV file to a more efficient file type called feather [18]. Feather is magnitudes faster to load and save, as well as more memory efficient compared to a CSV file. This helps to quickly query or graph data when working with large data sets.

Once in a feather file, we use Python Pandas which is a vectorized data structure [19]. Pandas has many built in functions to quickly query, count, filter, and obtain statistics on the data. Using Pandas we remove any duplicate updates from our data set. We define duplicate updates when two or more rows contain the exact same data for each column. Additionally, we use Pandas to find updates with errors that contained incorrect dates that are outside our specified time frame. We then save to new intermediate feather file with our updated data set.

The number of updates a looking glass receives in one hour can be drastically difference from another hour. Each looking glass offers a unique perspective and will have a different number of updates at different times. We want to see if there are any large spikes in update messages that may lead to large spikes in anomalies. We also want to see if there are any discernible patterns between looking glasses.

In Figure 3.1, we see the number of update messages received by each collector per hour for January 2, 2020. Since each collector receives a different number of total updates per day ranging from four to sixteen million as shown in Table 3.1, we scaled the Y-axis using a min-max scale to normalize our data between zero and one. In this graph we are looking at the frequency of announcements received by each looking glass within 1-hour periods. By scaling the counts we can more easily see any patterns between looking glasses.

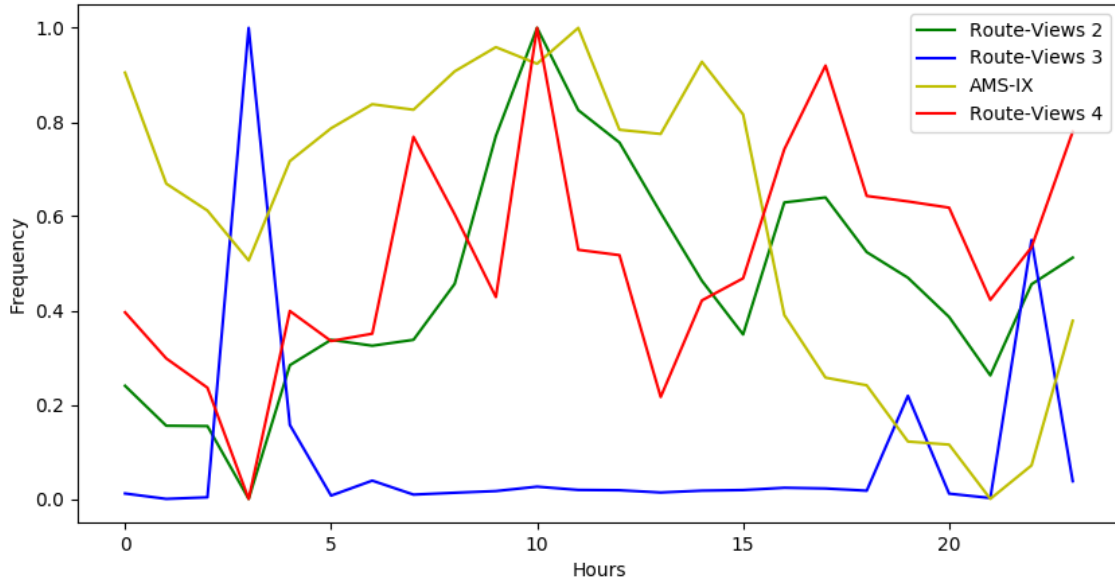


Figure 3.1. Looking Glass Update Frequency Comparison

With the exception of Route-Views 3, the other looking glasses are fairly well distributed across 24 hours with a peak around 10 AM. It should be noted as seen in Table 3.1, Route-Views 3 contains over 16 million updates for this day, which could have some play in the large number of updates between three and four AM. It is also interesting that at the time Route-Views 3 reaches its peak the other collectors go into a valley. Updates can be received from across the globe and maintenance periods can vary between ASes meaning increase in updates at various times. A large spike in updates could be completely normal for a certain time or looking glass, it could also be caused by an attack going on within that hour, a misconfiguration playing havoc, maintenance period, or something unknown. While we were able to see a few common peaks and valleys between looking glasses, without ground truth this maybe a coincidence. We will use these peaks in updates to scale our anomaly data later in this Chapter to rule out the increase of updates increasing the number of anomalies found.

### 3.3 Feature Engineering

In order to find an anomalous community set, we need features that help us identify normal behavior. We are looking at the behavior for a given community set not the individual

communities within that set. This community behavior is on a per-prefix basis, by looking at a community on a per-prefix basis we can see what maybe normal or potentially anomalous for that prefix. Since we do not have labeled data, we are performing unsupervised machine learning using different clustering algorithms. The goal of these engineered features is to give us clearly defined clusters, which we will consider normal communities. The outliers, are communities outside of any one cluster, and we will consider these to be anomalies. These abnormal communities may or may not be a true anomaly, but is exhibiting behaviors that are not commonly found in BGP communities.

The features we engineered use a streaming approach, by processing each update message one at a time in order of when it was received. We keep track of every prefix and community, that we have seen so far and adding new ones as they are received. Each feature is updated as each update message is processed on a per-prefix basis. Thus, if we were to look at all updates received for one prefix, we could see how the feature values changed over time, with each new update being processed.

For all of the features engineered we need to use the prefix-ASPATH-peer tuple. By using the prefix-ASPATH-peer tuple we know we are looking at the community announcement for the same prefix coming from the same peer and took the same path. If a prefix was to change even slightly from say 205.17.11.0/24 to 205.17.11.0/23 while considered a small change this is discussing twice as many hosts and could be attributed to any change in the community set. In addition, it is important that this announcement is coming from the same peer, as a different peer might be adding, filtering, or removing communities and we want as true of a comparison as possible. Additionally, we look at the AS path, if any changes occurred in the AS path, any AS on the path could have made changes to the community set. We define a change in the AS path as when the AS path is not exactly the same as the message that came before with the same prefix and peer. For a given prefix has the following AS path “123, 456, 789” in the first message. Then the second message has an AS path “123, 789, 456”, even though the ASes are all the same in the AS path the order is different. The reordering of AS path can be attributed to any changes to the community set. By looking at all three parts of this tuple we are removing as many variables as possible that could influence community changes. This allows for more consistent metrics for whether or not a community is changing.

All feature engineering was done using python Pandas and python dictionaries. By using a vectorized data structure and dictionaries we could keep track and quickly loop through all the data in order to create features. The following Table 3.2, presents the features we engineered with a brief description, and their range of values.

Table 3.2. Engineered Features

| Features                | Description                                                                                                                             | Range                |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| Time Since Change       | Captures length of time in seconds between updates that have had a change in the community set.                                         | 0 - 87,299           |
| Prevalence              | Update count of most frequent community for prefix divided by total the number of updates received.                                     | 0.0 - 1.0            |
| Persistence             | Count of the longest streak of consecutive updates with the same community for a prefix divide by the total number of updates received. | 0.0 - 1.0            |
| Foreign                 | Percentage of off path communities in the community set.                                                                                | 0.0 - 1.0            |
| Change                  | Percentage of communities that have changed within a community set.                                                                     | 0.0 - 1.0            |
| Seen Before             | If the community in an update has been seen before for that particular prefix, AS, or has ever been seen before.                        | 0 or 1               |
| Blackhole               | If the community set contains a community with the “X:666” blackhole community.                                                         | 0 or 1               |
| Graceful Shutdown       | If the community set contains a community with the “65535:0” Graceful Shutdown community.                                               | 0 or 1               |
| Community Length Change | How much the community length has changed from the previous update for the same prefix.                                                 | $-\infty$ - $\infty$ |
| RIR                     | The region authority for the peer AS in the update.                                                                                     | -1 - 5               |
| Granularity of Change   | The numerical difference between individual communities that have changed as an absolute value, then summed for the set.                | 0 - $\infty$         |

Two types of graphs will be used in this section, frequency graphs over time and Cumulative Distribution Function (CDF) graphs. A CDF graph is an easy way to visualize the distribution of values for a particular feature. The CDF is the probability that a variable takes a value equal or less than  $x$  [20]. The X-axis represents values of  $x$  and the Y-axis represents percentage values from 0% to 100%. The CDF graph also provides us with the median and quantile values, which allows us to better understand the distribution of values.

In the following Table 3.3, we are examining the mean values of the features we have engineered. These mean values appear to be fairly consistent across multiple looking glasses, which helps verify our concept of normal is universal across multiple looking glasses. In the remainder of this chapter, we are going to refer to these values and elaborate in detail to each respective feature.

Table 3.3. Mean Values Looking Glass Comparison

| 24hr Means            | Route-Views 2 | Route-Views 3 | Route-Views 4 | AMS-IX  |
|-----------------------|---------------|---------------|---------------|---------|
| Time Since Chg(Secs): | 21750.62      | 27336.49      | 24516.56      | 20809.1 |
| Prevalence:           | 93.40%        | 96.50%        | 96.30%        | 97.20%  |
| Persistence:          | 81.30%        | 89.40%        | 87.90%        | 92.60%  |
| Foreign:              | 9.80%         | 35.30%        | 12.90%        | 7.60%   |
| Change:               | 15.30%        | 11.60%        | 10.10%        | 11.10%  |
| Community Len Chg:    | 0.00028       | 1.72E-06      | 0.000256      | -0.0031 |
| Granularity of Chg:   | 2589.339      | 1426.324      | 2214.591      | 1092.85 |
| Community Len         | 4.893         | 5.324         | 4.928         | 5.844   |

### 3.3.1 Time Since Change

We tracked the time since we saw a change in the community for each prefix-ASPATH-peer tuple. Many prefix-ASPATH-peer tuples will send out the same community, by tracking the time we see between changes we can get an idea how long the community persists for each prefix-ASPATH-peer tuple. We believe that each prefix-ASPATH-peer tuple may change at different time intervals, and this time length is considered normal for that particular prefix-ASPATH-peer tuple. With every update received with no changes in the community,

the *Time Since Change* is updated to the difference between the time the current update was received to the time of the last changed update was received. Our hope is by keeping track of this time, we can get an idea how long communities persist for each prefix-ASPATH-peer tuple.

In order to better identify anomalous communities we want to identify how often communities change or rather how long they persist for. Hardt's work highlight that the majority of communities persist for quite some time before experiencing a change [8]. We wish to know specifics for each looking glasses. In order to do this we will take a look at the mean values for each looking glass and look at CDFs to get a better feel of what the majority of the *Time Since Change* values look like.

The *Time Since Change* feature has some variance between looking glasses in Table 3.3, but tells us the average community for a given prefix-ASPATH-peer tuple stays static on average over six hours. After roughly six hours that prefix-ASPATH-peer tuple then experiences a change in the community attribute. While a change less than six hours does not guarantee an anomaly, as this could be attributed to a policy change or maintenance period. Based on this information alone, we may conclude a prefix-ASPATH-peer tuple that is experiences frequent community changes in minutes or seconds is a potential anomaly, but the mean value for *Time Since Change* does not capture enough information. From here we will look at the distribution of values in a CDF graph to better understand the *Time Since Change* values.

In Figure 3.2 we can see a consistent curve across all four looking glasses with the *Time Since Change* feature. The X-axis in Figure 3.2 represents time in seconds since the last change in community for a prefix-ASPATH-peer tuple. Depending on the looking glass, 20-40% of the communities change within a second yielding a *Time Since Change* of zero. With the exception of AMS-IX, about 50% of communities have a *Time Since Change* of 10,000 seconds or less. All four looking glasses have roughly 80% of communities have a *Time Since Change* of 50,000 seconds or less. The *Time Since Change* feature has a wide range of values, we could equally say a large percentage of communities have a *Time Since Change* of zero, and say a large percentage of communities have a *Time Since Change* more than 10,000 seconds. We believe that looking at the timing of changes is still a useful feature, although it is difficult to say what exact value for *Time Since Change* is anomalous.

When combined with the other features we have engineered we feel it will have an added benefit.

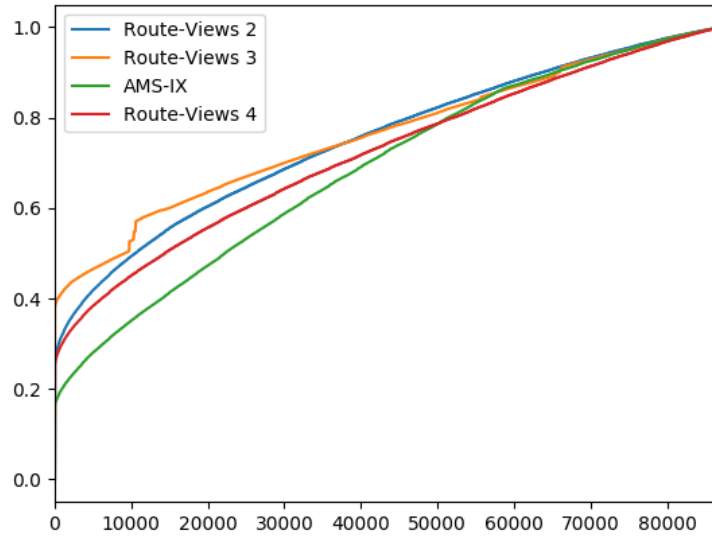


Figure 3.2. CDF: Time Since Change

### 3.3.2 Prevalence and Persistence

As discussed in Chapter 2, Alex Hardt discovered the majority of prefix-ASPATH-peers have had high community prevalence and persistence scores. We kept a running tally of how many updates we have seen for each prefix-ASPATH-peer tuple. For every update received we updated the total number of updates we have seen for that particular prefix-ASPATH-peer tuple at that moment in time. We then counted the communities that were the same, the highest count that were the same was then divided by our total number of updates seen for the prefix-ASPATH-peer tuple at that moment in time. This gave us our *Prevalence* score, where prefix-ASPATH-peer tuples that experienced frequent changes in the community attribute would have a low score. This high frequency of change could be an indicator on anomalies occurring.

*Persistence* score allows us to keep track of the longest running streak, where the community hasn't changed for that prefix-ASPATH-peer tuple. Generally if a community changes it

will have one change then consistently send its prior community thereafter leading to long running streaks of the same communities. A prefix-ASPATH-peer tuple that is sending out many updates but has low streak of the same communities is not common and could be another indicator of an anomaly.

The *Prevalence* and *Persistence* features play a key role in capturing how communities persist over time. Prior work by Alex Hardt has shown these both to be quite high, we are interested to see the overall distribution of scores [8]. We want to see if we can see a clear majority values that we will consider normal, and rare values that could indicate an anomaly.

*Prevalence* and *Persistence* scores are high across all looking glasses as seen in Table 3.3. These high scores tell us for all prefix-ASPATH-peer tuples there communities change very little, and have a consistent “normal” community for each prefix-ASPATH-peer tuple. By looking at the distribution of values for *Prevalence*, in Figure 3.3 we can see roughly 10% of announcements have a *Prevalence* score of 80% or less. The remaining community announcements have a *Prevalence* greater than 80%. This distribution tells us that the majority of *Prevalence* scores are high and only a small percentage of scores are low, meaning a low *Prevalence* score is potentially anomalous and will not be very common. This is a good indication that *Prevalence* will work well as feature for an anomaly detector and should minimize false positives.

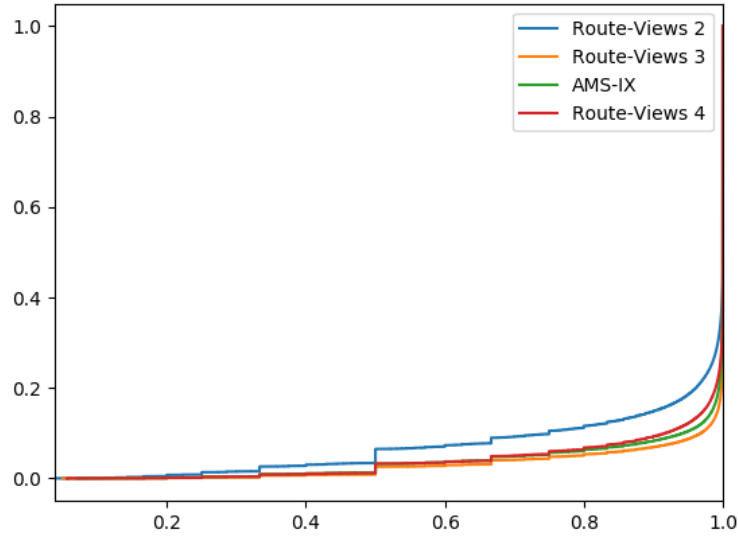


Figure 3.3. CDF: Prevalence

Similar conclusions to *Prevalence* feature can be made with the *Persistence* feature, by looking at Figure 3.4. With the exception of Route-Views 2 averaging slightly less, the rest of the looking glasses have 20% of *Persistence* scores of 80% or less. The remaining 80% of *Persistence* scores are roughly 80% or more. This distribution tells us that the majority of *Persistence* scores are high and only a small percentage of scores are low, meaning a low *Persistence* score is potentially anomalous and will not be commonly seen. This is a good indication that the *Persistence* feature will work well for an anomaly detector and should minimize false positives.

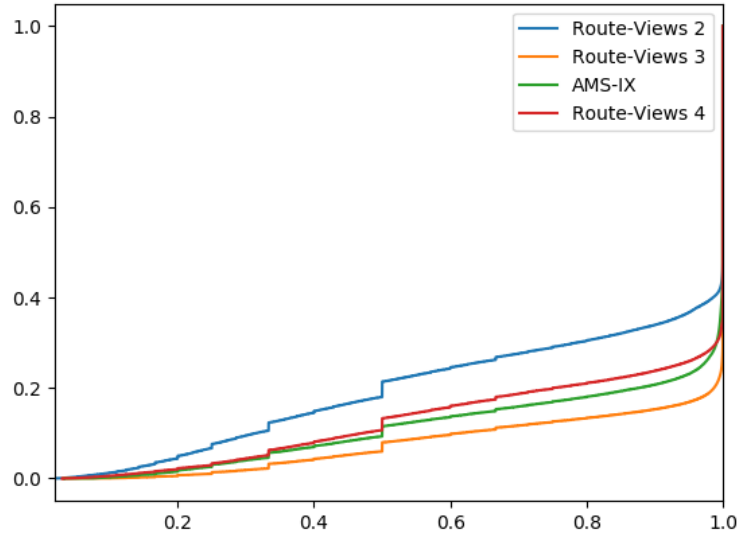


Figure 3.4. CDF: Persistence

### 3.3.3 Foreign

As described in Chapter 2, foreign communities are off-path communities that correspond to an AS number that is not on the AS Path. This can occur for a number of reasons, first it could be improper filtering on a ASes part. If an AS continues to pass an announcement to its peers that may not be the target or no longer on the same AS path without filtering out the correct communities, then communities are leaked to other ASes. Second, an AS may add a community with an AS number that is not on the AS path, this could be due to a misconfiguration or some other policy reasoning. Lastly it is using communities in a different way that no longer follows the generalized agreement for communities. We expect to see “AS number : X” where X is some 16-bit value. Instead, an AS is using a different value for the first two octets not representing an AS number, e.g., “0:123”. Where 0 is not a valid AS number, but is used for some custom policy that AS has created. This *Foreign* community might not always be anomalous as it could be due to how BGP messages propagate and could just be a misconfiguration. This could be used maliciously or have unintended consequences and would be then categorized as anomalous. We will look at the first part before the “:” of each individual community in the community set, then compare

each of these with the AS path. We will tally up the number of *Foreign* communities and divide them by the total number of individual communities. This gives us a percentage of *Foreign* communities in the community set, the higher the percentage the more the *Foreign* communities. Where zero would mean no *Foreign* communities and one would be all of the communities are *Foreign*. Our intuition by having this variable percentage of *Foreign* we can see the community with a finer granularity and perhaps help identify anomalous communities.

As indicated in Chapter 3, due to the transitive nature of communities and the lack of standardized implementation of communities, off-path communities can appear within the community set. These *Foreign* communities are an artifact of how BGP policy is implemented by an AS not filtering out communities with AS numbers not in the AS path, before continuing the propagation of an update. Thus, it may or may not be interpreted as an anomaly, but due to the power a *Foreign* community can have, we wish to still examine these communities as a potential anomaly. Using the Table 3.3 and the Figure 3.5 we wish to discover how prominent *Foreign* communities are.

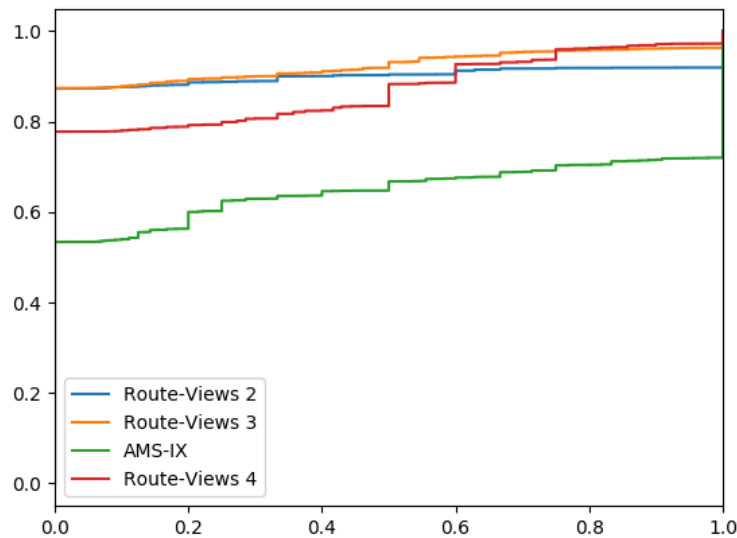


Figure 3.5. CDF: Foreign Communities

Looking at Figure 3.5, we can tell that for all looking glasses over half of the announcements contain 20% or less foreign communities. For all three Route-Views looking glasses at least 10% of announcements contain 50% or more *Foreign* communities. Examining mean values in Table 3.3, the percentage of *Foreign* communities is quite small between 10-15%, with the exception of Route-Views 3. Route-Views 3 has a much higher *Foreign* community percentage of 35.3% as seen in Table 3.3. This could be due to an AS Route-Views 3 peers with either consistently leaks off-path communities or has a policy that extensively uses off-path communities. We believe this is due to a unique policy as the community “0:28917 666:666 41441:41441:41441” was found thousands of times for multiple prefixes, and many were classified as anomalies. If this was an off-path leak we would expect to see off-path AS numbers, but instead we see: ‘0’ a non-valid AS number, ‘666’ the blackhole community, and the six octet or three tuple “41441:41441:41441” community, which would be found in an extended community announcement.

The first three ASes on the AS path are consistently the same “202365 206499 43727”, but unfortunately none of ASes have any BGP community information online. The last community “41441:41441:41441” gives us a possible clue what may be happening here. The number “41441” is a valid AS number, representing Speed-IX out of the Netherlands. According to their website “0:peer-AS” tells an AS not to announce the prefix in the announcement to a specific peer [21]. While they have no mention of a policy for “666:666”, they do mention the use of six octet or double “:”. Although there are no specific matches for the community “41441:41441:41441”, all the uses for the double “:” revolves around advertising or not advertising to a certain peer or all peers.

Speed-IX has some nontraditional BGP communities, yet they are not on the AS path. We believe because they are nontraditional many ASes do not know what to do with them, so instead of filtering them out they pass them along. Thus they are most likely a leak because of this unusual community use. In Chapter 5, we will discuss this in more detail as well as how foreign communities affect our results from our anomaly detector.

Between Table 3.3 and the Figure 3.5, we can see significant use of *Foreign* communities. It is important to note that this use varies between prefix-ASPATH-peer tuples, where high *Foreign* use may be normal for some prefix-ASPATH-peer tuples and may be potentially anomalous for others. We find the *Foreign* feature is beneficial as these are unusual

announcements and would be interesting to learn more about.

### 3.3.4 Change and Granularity of Change

*Change* looks at the previous community announcement for a prefix-ASPATH-peer tuple and compares it to the current announcement. Each individual community in the announcement is compared to the previous announcement, while keeping track of how many of the individual communities changed. We then divide the number of communities that have changed by length of the community set in the previous announcement. If only communities are added to the set and none of the original communities have changed, they will still reflect in the *Change* percentage. For example, if we see a community with the length of ten, then the length is increased to twelve with no changes to the original ten, the *Change* would be  $2/10$ . Similarly, if the length went from ten to eight with no changes in the individual communities, the *Change* would also be  $2/10$ . This yields us a *Change* percentage, that allows us to see how much change has occurred. A low percentage or a small *Change* could be normal for a given prefix-ASPATH-peer tuple. While a high percentage or high *Change* for a given prefix-ASPATH-peer tuple might be an indicator of an anomaly.

We already have a feature that tells us what percentage of communities have changed, but it would be beneficial to see a finer *Granularity of Change*. For example, if community “1234:437” changes to “1234:439” compared to “1234:437” changing to “1234:2730”. The first change “437”  $\rightarrow$  “439” only has a difference of two. Where “437”  $\rightarrow$  “2730” has a difference of 2,293. The *Granularity of Change* feature would capture this difference as an absolute value, then sum up any other differences from communities that changed within the set. Resulting in a number greater or equal to zero if no change occurred. The intuition here is a change with a small number might be normal fine tuning of routes or policies. Where a change with a large number might indicate a more drastic change in policy or routing and potentially more anomalous behavior.

There are limitations to the *Granularity of Change* feature as implemented that we would like to highlight. The *Granularity of Change* feature presents a matching problem, and depending on methodology generates different results. For example with our methodology, if given the community set “1234:437”, then in the next update seen for the same prefix-ASPATH-peer tuple the community set is “1234:439 1234:500”, the “1234” would match

on the first instance “1234:439” for a difference of 2. The “1234:500” is an addition to the community set and not reflected in the *Granularity of Change*. This may not be the perfect approach, as there are many ways this *Granularity of Change* could be implemented, this is the way we chose how to use it. In the future work section of Chapter 5, we will discuss other possible ways it might be implemented.

We measure change in two ways to help better understand how communities change. By looking at how communities change we want to discover how much change is normal and if the change features will help identify potential anomalies. We use *Change* to capture the percentage of individual communities that have changed within the community set, where *Granularity of Change* captures the numeric difference between communities that have changed to see the magnitude of the change.

In Table 3.3 the low *Change* percentage across all four looking glasses shows us that when a community does change for a prefix-ASPATH-peer tuple that only a small number of individual communities change while the rest remain the same. For example Route-Views 4 has almost exactly 10% *Change*, thus for a community with the length of ten only 1 of those communities has change while the rest remain the same. If community for a prefix-ASPATH-peer tuple experiences 90% *Change* then it is probable this could be an anomaly. Looking at the distribution in Figure 3.6, about 80% of updates have less than 30% *Change*. If we look at the curve for all four looking glasses we can see a small percentage of updates experience a *Change* greater than 50%. This small percentage of updates with a high percentage of *Change* will work well as an indicator that *Change* will be a good feature for finding anomalies.

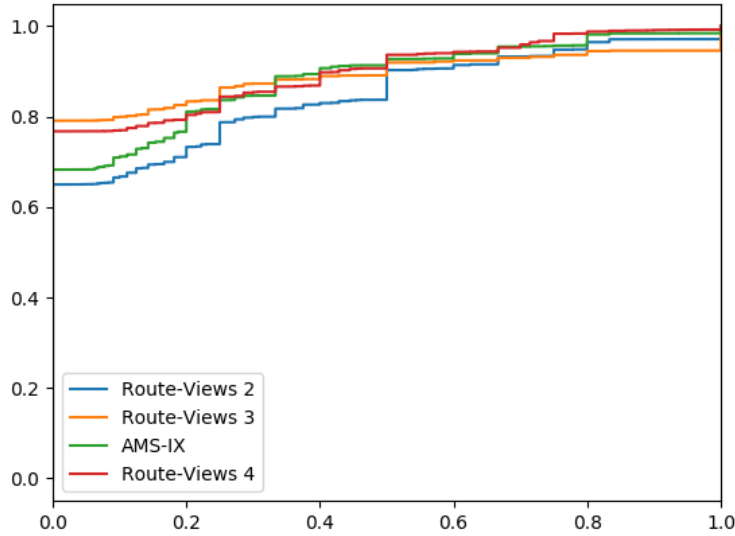


Figure 3.6. CDF: Community Change

The *Granularity of Change* for Route-Views 2 averages roughly 2,589 in Table 3.3. Meaning if an prefix-ASPATH-peer tuple experiences a change on average that change is an increase or decrease across one or more communities of 2,589. In order to better understand this number we must consider the scale of this value, a two octet community value can range between 0-65,535. This means each individual community in the community set can have a difference of 65,535, the absolute value of each difference is then summed up to give us the *Granularity of Change* score. Thus, the fact multiple individual communities can change at the same time, the limit of *Granularity of Change* is equal to the length of communities times 65,535. Looking back at the Route-Views 2 *Granularity of Change* of 2,589 can be interpreted as relatively small change given the scale of *Granularity of Change*.

In Figure 3.7, we plot a CDF graph with the X-axis in log form to better differentiate change and see the lines for each looking glass. We discover all four looking glasses have between 60-80% of the updates have *Granularity of Change* score of zero and 2-5% have a change of 1. The *Granularity of Change* grows rapidly to over 100,000, this minority of updates makes up roughly 20% to 30% for all looking glasses and values ranging anywhere from 10 to over 100,000. We believe this feature might have some indication of a possible anomaly

but by itself would not filter enough announcements out leading to many false positives.

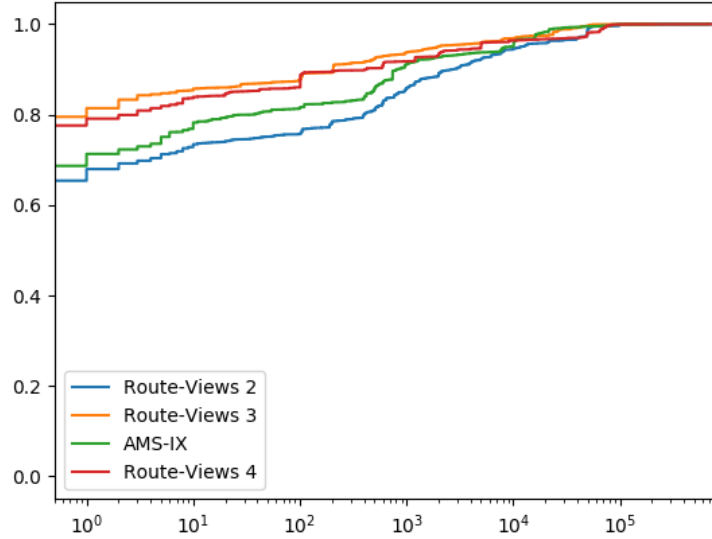


Figure 3.7. CDF: Granularity of Change

### 3.3.5 Seen Before

We want to know if the each individual community in the community set, has been seen before or if it has never been seen before now. By looking at the individual communities, we are able to see how an individual community may be new to a particular prefix-ASPATH-peer tuple, but maybe not new for that prefix, AS, or looking glass. If it is something that has been seen before it might be normal behavior for that prefix-ASPATH-peer tuple or even normal for that AS. If it has never been seen before it could potentially be an anomalous community. We will break down this seen before into three categories Prefix, AS, and Ever.

The first category is *Seen for Prefix*, meaning all the times we have seen this prefix have we ever seen this community in an announcement. While not the full prefix-ASPATH-peer tuple it still indicates if a community may be normal for that particular prefix if it has been seen before.

The next category *Seen for AS*, represents if the peer AS who sent the announcement has ever

sent an announcement containing this individual community for any prefix. The intuition here is if this AS has sent out the announcement before it is not an unknown community announcement and perhaps normal for that particular AS. Where as if it hasn't been seen for a particular prefix or AS then it has more potential to be an anomaly.

The last category *Seen Ever*, checks a community to see if it has ever been seen before for any AS. This final check is based off the looking glass, for all the announcements a looking glass has received has it ever received an announcement containing this particular community. If this last check fails then this individual community has never been seen before for this collector, a rarity like this could be indicative of an anomaly.

The *Seen Before* features give us an idea of how new a particular community is for a prefix, AS, or looking glass. This is done by keeping track of individual communities on a per prefix, per AS, and per looking glass perspective, opposed to the entire set on a prefix-ASPATH-peer tuple perspective. For example, we could identify a new community for a prefix, but this individual community may not be new for that AS or looking glass, this gives us an idea how new an individual community might be and possibly filter out potential false positives. We want to find out how often a new individual community is introduced, and the *Seen Before* features allow us to do that. By looking at the percentages for each looking glass we hope to see a small number of announcements that contain individual communities that have never been seen before that will yield us less potential anomalies. We also wish to discover if this is looking glass dependent, or if this is consistent across all the looking glasses we used.

The *Seen Before* features have values either 0 or 1, we choose to examine these results in a table rather than a CDF. The percentages in Table 3.4 are the percentage of updates received by the looking glass where all of the individual communities were seen before for the prefix, AS, or Ever. The last row shows us the number of updates that contained one or more individual communities never seen before. In Table 3.4 we can see consistency across multiple looking glasses, with slightly lower percentage for *Seen for Prefix* and *Seen for AS* for Route-Views 3. The *Seen Ever* feature is 99.9% for all four looking glasses. This tells us that in 24 hours less than .1% of updates contains an individual community that has never been seen before for each looking glass. The *Seen Ever* feature has potential to filter out a large percentages of announcements, but this does not necessarily correlate as an anomaly.

As anomalous communities may be a rare occurrence, but could have possibly been seen before. Thus, it is difficult to tell how beneficial the *Seen Before* features will be. Our hope is by being able to tell if it has been seen for the prefix, AS, or looking glass, our clustering algorithm can glean more information to make a better decision.

Table 3.4. Seen Before

| Seen Before        | Route-Views 2 | Route-Views 3 | Route-Views 4 | AMS-IX |
|--------------------|---------------|---------------|---------------|--------|
| Seen for Prefix:   | 82.6%         | 67.4%         | 82.9%         | 87.9%  |
| Seen for AS:       | 92.2%         | 80.9%         | 92.1%         | 93.6%  |
| Seen Ever:         | 99.9%         | 99.9%         | 99.9%         | 99.9%  |
| Never Seen Before: | 3841          | 10,777        | 4612          | 4270   |

### 3.3.6 Blackhole

As described in both Chapters 1 & 2 blackhole communities are powerful and we expect them to be an uncommon occurrence due to this power. When we do see a blackhole community we believe this may not always be an intentional use, but perhaps malicious or accidental due to a misconfiguration. Knowing that the blackhole community is not universally agreed upon among ASes, we still wish to know when the community “X:666” is being used as many ASes still follow this implementation. We are treating the *Blackhole* feature like a flag and anytime a community set contains “X:666” we will note it with this feature. We believe because of the power the blackhole community has, it will make for a strong indicator for a potential anomaly.

The *Blackhole* feature works by flagging any announcement that contains the blackhole community, we want to get an idea of how often the blackhole community is being used and if it varies between looking glasses. By figuring out how often the blackhole community is being used we hope to determine the how common or rare an update with a blackhole community is. This allows us to get a better idea how the black hole communities are getting used in day to day operations.

The Table 3.5 shows percentages of announcements containing the blackhole community for each looking glass. The percentages are fairly small for all looking glasses, but the number

of announcements using the blackhole community ranges from 133,989 to 1,392,413. As mention in Chapter 2, AS3356 uses “x:666” to communicate a peer route, meaning many of these announcements may be used for a different purpose than blackholing traffic. The community “666:666” is found thousands of times, but because of the nature of BGP communities it is near impossible to say who added it. We examined a few of these announcements and looked at each AS on the AS path but none of the ASes listed their policy for BGP communities online. As mentioned earlier under the *Foreign* communities section, AS41441 Speed-IX was suspect based on its policy for other *Foreign* communities. Speed-IX may be contributing to the use of the blackhole community, but does not list any policy using “666:666” community. Due to the number of updates using the blackhole community, we feel many ASes are using them in there day to day operation, not only for blackholing traffic, but for other purposes as well.

Table 3.5. Blackhole

| Looking Glass | % of Updates Containing the Blackhole Community | Number of Updates Contain- ing the Blackhole Community |
|---------------|-------------------------------------------------|--------------------------------------------------------|
| Route-Views 2 | 6.50%                                           | 577,522                                                |
| Route-Views 3 | 8.30%                                           | 1,392,413                                              |
| Route-Views 4 | 2.20%                                           | 133,989                                                |
| AMS-IX        | 4.90%                                           | 241,353                                                |

### 3.3.7 Graceful Shutdown

*Graceful Shutdown* is a standardized community from RFC 8326 that informs other ASes a path is about to shutdown [22]. This is done to prevent any packet loss or connection delay. The community “65535:0” communicates a path is being shut down deliberately perhaps for maintenance. Since this community is telling other ASes to take a different path and is well defined it has the potential to steer traffic. Similarly to blackholing *Graceful Shutdown* is a powerful community and we believe it could be an indicator for anomalous BGP community activity. If the *Graceful Shutdown* community has never been seen for a prefix or AS and has suddenly appeared in an update message, this could be a potential indicator of an anomaly.

We expected *Graceful Shutdown* feature to be similar to the *Blackhole* feature, but we found no announcements that contained this feature. We did find some use of the first part “65535:”, but not an exact match of “65535:0”. Even though it is defined it is not being implemented or at least not frequent enough to be a beneficial feature, thus we decided to drop it.

### 3.3.8 Community Length Change

In Hardt’s research as discussed in Chapter 2, the length of the Community set does not change significantly. An AS along the AS path may add or remove one or two communities every once in a while, but to see an AS add 15 communities would be abnormal. We believe if we see a large decrease or increase in the number of communities in community set, then it has a higher probability of being an anomaly.

We are interested in knowing the average length of community sets, and to see if this length varies between looking glasses. The length of communities appears to be very consistent in Table 3.3 across multiple looking glasses averaging roughly a *Community Length* of five. This data is consistent with Hardt’s findings [8]. Thus, for each prefix-ASPATH-peer tuple only five individual communities are generally used and if a prefix-ASPATH-peer tuple were to say have fifteen it would be highly unusual and potentially anomalous.

While previous work has highlighted that the community length is generally less than ten, we are interested in knowing just how often the length of the community changed and by how much it changed. We want to know if this change in length is universal across multiple looking glasses or specific to each glass. We believe it is important to find the change in length in defining how normal communities behave, in order to help identify abnormal community behavior.

In Table 3.3 all four looking glasses had a very small change in *Community Length Change*. Route-Views 2,3,4 all had a very small increase in length, with AMS-IX having a slight decrease in community length. Meaning when a prefix-ASPATH-peer tuple has a change in its community the length generally does not change. Thus, communities having their length drastically increased or decreased is rare and could potentially be an anomaly.

In Figure 3.8, we plot the X-axis in log form to better differentiate change and see the

lines for each looking glass. Roughly 90% of communities experience a *Community Length Change* of one or less. A very small percentage of communities experience *Community Length Change* greater than two and the scale goes over a hundred. In Chapter 5, we will discuss anomalies where the community length increased by over a hundred. This drastic *Community Length Change* is rare, but will work as a good indicator for anomalies and should yield little to no false positives. If we solely looked at *Community Length Change*, we may miss some anomalies as this change in length is such a rare occurrence.

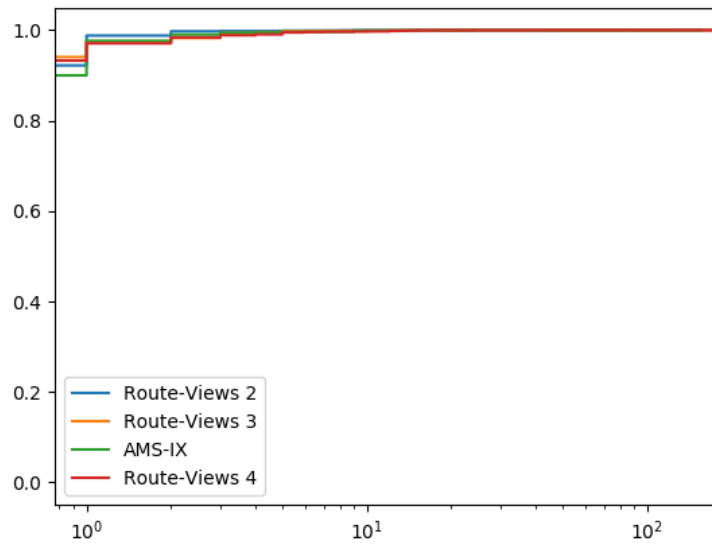


Figure 3.8. CDF: Community Length Change

### 3.3.9 RIR

We believe it would be beneficial to see what RIR the peer AS is a part of. By knowing what region the peer AS is from, we might be able to tell what is considered “normal” communities for that region. While knowing the region is not directly an indicator of an anomaly, combined with the other features it could potentially indicate what is normal or abnormal for a certain region. Thus, we believe it will be a good supportive feature in our anomaly detector.

Each looking glass is connected to different peers from different regions. The *RIR* feature

looks at what region these peers are a part of, or if it is a 32-bit AS, which is a limitation in our in our model that we will expand on this in our future work section of Chapter 5. Our hope is that if certain regions or 32-bit AS peers use unusual communities that it may be normal for that particular region. The Table 3.6 shows the distribution of peers for each looking glass by the *RIR*. We can see each looking glass has a unique distribution of announcements received from peers in different regions. Due to the lack of ground truth, we cannot definitively say how beneficial the *RIR* feature is. While it may give our non-supervised clustering a bit more information, our methodology has potential flaws that we will discuss in the future work section of Chapter 5.

Table 3.6. RIR

| <b>RIR</b> | <b>Route-Views 2</b> | <b>Route-Views 3</b> | <b>Route-Views 4</b> | <b>AMS-IX</b> |
|------------|----------------------|----------------------|----------------------|---------------|
| ARIN:      | 59%                  | 54%                  | 18%                  | 31.9%         |
| RIPE:      | 31%                  | 18%                  | 21.3%                | 44%           |
| APNIC:     | 10%                  | 7%                   | 60.3%                | 0.01%         |
| LACNIC:    | 0%                   | 0%                   | 0%                   | 0%            |
| AFRANIC:   | 0%                   | 0%                   | 0%                   | 0%            |
| 32-bit AS: | 0%                   | 21%                  | 0.4%                 | 24%           |

### 3.4 Machine Learning and Algorithms

The unlabeled data we are working with, and the sheer number of updates that occur, make the task of creating an anomaly detector particularly challenging. Using labeled data would make this task easier, but labeled data indicating an attack or misconfiguration with BGP communities is nearly non-existent. Even if we could obtain data and use our engineered features, this example would only be beneficial for where ever that looking glass was. This is because of each looking glass has such a unique perspective. While some engineered features could be universal to multiple looking glasses not all could be used. An anomaly detector would have to be trained and tested on the data from the same collector in order to achieve the best results.

The other major challenge is training on a data set for just a single day has nine million rows

using roughly a dozen columns in order to create a model takes hours or even days to process. As ASes and communities change over time this model would have to be retrained over time as well to keep up with these changes, in addition to each collector having to have its own model as well. While we know only a small percentage of community announcements will be anomalous, because of how many announcements are received by each collector the number of potential anomalies quickly adds up. Each collector we look at averages nearly nine million announcements per day. Even if a tenth of a percent or .1% were flagged anomalous it would yield around 9,000 anomalies.

All of the algorithms we use have a contamination variable or outliers fraction that can be set to filter the anomalies only picking the top n% of anomalies, in our case we set our outliers fraction equal 0.0001. While this may not be exactly correct, our hope is we are only looking at the top percent of anomalies even though we may be missing some. With a setting of 0.0001 outliers fraction, 300 to 800 announcements are tagged as anomalies depending on which algorithm is used for Route-Views 2. The combined total of all algorithms results in 3,124 unique anomalies as seen in Table 4.1.

In this thesis we use several unsupervised clustering algorithms, but because of the data being unlabeled it is difficult to say if one is better than another. We do not have metrics like accuracy, precision, or recall to use to compare these algorithms. Our plan is to combine all of the results tagged as an anomaly and examine the anomalies that were found by the most algorithms. By doing this, we will reduce the number of anomalies we have to examine. Additionally, with no hard metrics to easily prove one algorithm is working better than another, we are essentially taking a majority vote between algorithms. While we could manually go through each anomaly from each algorithm, this would be extremely time consuming given a few thousand anomalies. By looking at the anomalies found by the most algorithms, we have a good starting point and ordering to manually analyze. While we do not have time to go through each anomaly, we will at least be able to look at the most prominent anomalies. We are using an ensemble method with our unsupervised machine learning, if a community is found to be anomalous by multiple algorithms it is more likely to be a true anomaly.

We will attempt to contact via email the peer ASes where the anomaly was received from, but we might not get a reply back. Even if we do get an email replying to our question

about an anomalous community from the peer AS they may not be the AS that added the anomalous community, and not be able to give us any helpful information. Leaving us to Google for possible meanings of anomalous BGP communities. Some ASes put a list of all the communities they support on their website which makes it easy to identify and understand what the community is doing. Unfortunately many ASes do not do this, making it even more difficult to understand what the community or anomaly means. Lastly some ASes may post status updates online to their status which may tell us of a possible misconfiguration or malicious activity.

In this thesis we use the python library PyOD to perform outlier detection. PyOD is a powerful toolkit that offers many different algorithms for detecting outlying objects in multi-variant data [23].

### **3.4.1 Angle-based Outlier Detector (ABOD)**

The Angle-based Outlier Detector (ABOD) algorithm uses angle measurements in combination with distances between two points. This is done for each for each dimension/feature in the data set. By taking the variance of angles between pairs of dimensions it can help discern between points that are similar to one another and points that we would consider to be an outlier [24].

### **3.4.2 Cluster-based Local Outlier Factor (CBLOF)**

Cluster-based Local Outlier Factor (CBLOF) uses a clustering algorithm to classify clusters as small or large. Then the “anomaly score is then calculated based on the size of the cluster the point belongs to” and “the distance to the nearest large cluster” [25]. The smaller the cluster and further the distance the more likely it will be classified as an outlier.

### **3.4.3 Feature Bagging**

The feature bagging algorithm uses sub-sampling to build detectors on randomly selected features. Then uses averaging and a combination of other methods to prevent over-fitting and improve predicting. The sample size chosen is always the same when building a model, but the number of features randomly selected can vary. By averaging the maximum of all the base detectors, a prediction score can be generated [26].

### **3.4.4 Histogram-base Outlier Detection (HBOS)**

Histogram-base Outlier Detection (HBOS) assumes feature independence similar to Naïve Bayes. It calculates the degree of outlying-ness by building histograms. Each feature is converted into a histogram with either fixed or dynamic bin widths. All histograms are then used to compute an anomaly score for predicting outliers [27].

### **3.4.5 Isolation Forest**

The Isolation Forest algorithm randomly selects a feature and then randomly selects a value for that feature to split on between the minimum and maximum values. This process is repeated on a forest of random trees. Normality is measured by the average path length across all trees. When a particular sample produces a shorter path length it is highly probably to be an anomaly [28].

### **3.4.6 K Nearest Neighbors (KNN)**

The K Nearest Neighbors (KNN) algorithm uses a density measurement to calculate outliers. KNN measures the length to the kth nearest neighbor to determine the outlying score. Where k is a positive number, the algorithm will look at five different k neighbors for comparing distance. The largest score out of the five is then taken and used as the distance score [29]. The larger the distance score the more likely the particular sample is an anomaly.

### **3.4.7 Average KNN**

The average K Nearest Neighbors (average KNN) works similarly to regular KNN. Once all five different k neighbors are scored instead of taking the largest score, all five scores are then averaged [29]. Then this average is used as the distance score for comparison and outlier detection.

### **3.4.8 Algorithm Comparison**

In this section, we will compare algorithms we used to see how they perform differently. Without ground truth, there is no way to truly measure how much better one algorithm performed compared to another. Thus, we choose to look at other characteristics of the algorithms such as processing time, and number of anomalies found compared to accuracy,

precision, and recall if we had labeled data. Before we run any algorithm on our data set we first scale all of our features individually using a min/max scaling. This gave all of our features values between 0 and 1. By doing this we help remove possible weighting of features due to the changes of values they may have. This allows us to have more consistent results as our the values of our data set change.

We initially ran all seven algorithms on a 8-hour data set to get an idea of how the algorithms would perform. It quickly became apparent, that two of the algorithms were not performing the same. The angle-based algorithm ABOD was much slower to process than the rest of the algorithms and yielded no anomalies found. Without ground-truth we cannot say for certain that there was simply no true anomalies found. Given that the other algorithms were finding hundreds of anomalies in the same data set, we believe the algorithm was unsuited for our data type. Additionally, it took multiple times longer to process than the rest of the algorithms.

The other algorithm that we found to be under performing was feature bagging. While feature bagging found anomalies it was taking magnitudes longer in time to run, and the majority of the anomalies it found were also found in the other algorithms. In order for us to process 24 hours of data, it would have taken multiple weeks to run. Where the other algorithms could finish in a few days, or even a few hours. Between the long processing time, and little added benefit of its results we decided not to use feature bagging. This allowed us time to look at additional looking glasses and multiple days of data.

Processing time for HBOS, Isolation Forest, and CBLOF took a few hours each depending on the number of announcements for a 24-hour period. On the other hand, both KNN and average KNN took a few days to process for the same data set. If we were to look at longer periods of data such as weeks instead of days, we might consider dropping the KNN algorithms. Again, given our lack of ground truth it is near impossible to tell how beneficial having the KNN algorithms are.

---

## CHAPTER 4:

### Analysis

---

In this Chapter, we will compare the results of the algorithms that we chose to use. We will look at the number of anomalies found by each algorithm, and the number of anomalies that were found by multiple algorithms. We will then perform data analysis to answer some key questions regarding anomalies. All questions work under the assumption that any anomaly found by our detector is a true anomaly. This is because without ground truth we cannot verify if the anomalies found are false positives or if the algorithms miss false negatives.

### 4.1 Algorithms

In order to compare the algorithms, we will look at the number of anomalies found by each looking glass for each of the five algorithms. We also want to know how many anomalies are found by more than one algorithm.

#### 4.1.1 Limitations

While this study discovers interesting events relating to BGP communities in the wild, we emphasize several limitations. First, without labeled data we are forced to perform unsupervised learning and empirically analyze the results. Further, without strong examples of normal and anomalous events, we design and engineer features that we believe will capture behavior and allow for discrimination. However, since we chose and designed these features, it is unsurprising that the resulting anomalies reflect anomalies in these same features.

The lack of labeled data further highlights that an “anomaly” will typically represent one or more features that statistically differ from their normal values. This, however, is not to say that they represent a particular type of behavior; indeed the anomaly may represent a misconfiguration or bug as easily as a malicious attack.

In future work, we hope to obtain a corpus of labeled data. However, with few ways to compare algorithms without labeled data, we will validate anomalies by seeing if more than one algorithm found that particular anomaly.

### 4.1.2 Number of Anomalies Found

The Table 4.1 shows the number of anomalies found by each algorithm for each looking glass and the number of anomalies found by more than one algorithm. Referring back to Table 3.1, Route-Views 3 had over 16 million announcements which led to issues processing the KNN algorithms, thus given time constraints we only ran the other three algorithms. We can see in Table 4.1, the majority of the algorithms found a similar number of anomalies, with the exception being average KNN finding less than half of the anomalies the other four algorithms found. Without ground truth or labeled data we cannot say if average KNN has more false negatives or if the other algorithms are having more false positives. In Table 4.1, we can see HBOS, Isolation Forest, and CBLOF are all very close in the number of anomalies they find. The KNN algorithm trails these three algorithms with roughly 20% less anomalies found.

Table 4.1. Algorithm Comparison

| Looking Glass:                              | Route-Views 2 | Route-Views 3 | Route-Views 4 | AMS-IX |
|---------------------------------------------|---------------|---------------|---------------|--------|
| Histogram (HBOS):                           | 860           | 1674          | 605           | 483    |
| Isolation Forest:                           | 879           | 1678          | 593           | 485    |
| Average KNN:                                | 345           | N/A           | 195           | 158    |
| Cluster-based (CBLOF):                      | 879           | 1628          | 607           | 643    |
| K Nearest Neighbors (KNN):                  | 708           | N/A           | 504           | 404    |
| Anomalies Found by More Than One Algorithm: | 485           | 591           | 290           | 272    |
| Total Number of Unique Anomalies:           | 3124          | 4441          | 1956          | 1828   |

Looking at the last row in Table 4.1, the number of anomalies found by more than one algorithm appears to be consistent across all four looking glasses given the number of announcements received by each looking glass. We found only a handful of announcements

were found by all five algorithms, but seeing a few hundred found by two or more algorithms shows some consistency across algorithms.

In Figures 4.1, and 4.2, we use a CDF plot to better visualize the distribution of the number of algorithms detecting each anomaly. In both figures, the X-axis is the count of algorithms, with the Y-axis being the percentage of anomalies found. Figure 4.1, shows algorithm count by each looking glass, where Figure 4.2 shows the algorithm count by each day. Both figures show very similar lines, meaning no matter the looking glass or day, the percentage of algorithms finding the same anomaly is nearly the same. Roughly 85% of anomalies are only seen by one algorithm, showing the majority of anomalies found by each algorithm is unique from other algorithms. On the other side to this, about 15% of anomalies are seen by 2 or more algorithms. Our unlabeled data presents a large challenge; having anomalies found by multiple algorithms give us hope to be true positive anomalies.

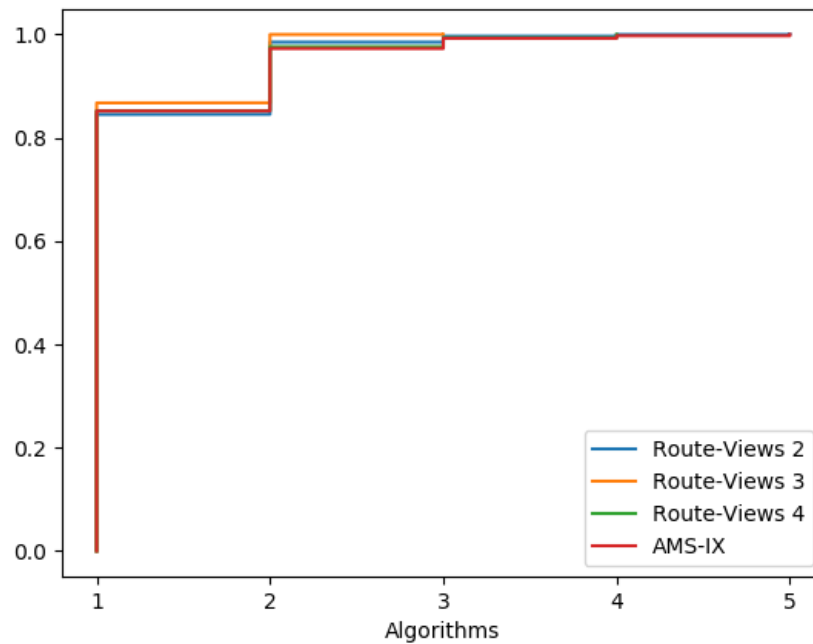


Figure 4.1. CDF: Algorithm Count by Looking Glass

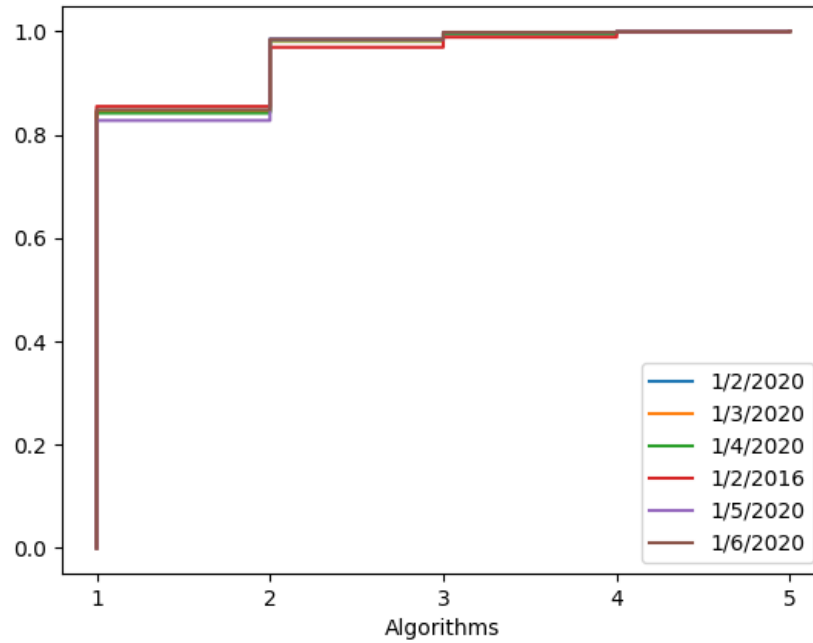


Figure 4.2. CDF: Algorithm Count by Days

### 4.1.3 Anomaly Distribution

We want to see how distributed the anomalies are for each algorithm for a 24-hour period. We are looking to see if we can find any similarities or differences in what time anomalies are found for each algorithm. By making this comparison, we hope to see how consistent each algorithm is with one another. Generally, if a community is classified as an anomaly there are more announcements with that same community. One algorithm might see only one of those announcements as an anomaly where other algorithms sees several of those announcements as an anomaly. We expect there to be some differences between each algorithm given the information from Table 4.1.

In Figure 4.3 below we are looking at the number of anomalies found for each algorithm on 1/2/2020 for Route-Views 2. The Y-axis represents the number of anomalies found within a given hour on the X-axis. The Y-axis shows the anomaly count, thus we see average KNN below the rest of the algorithms. While some algorithms align over a peak or valley, overall

no two algorithms consistently aligned on the same peaks and valleys. The KNN and average KNN have many differences in peaks and valleys, but upon closer examination all 345 anomalies found by average KNN were also found with KNN. Thus, average KNN has little added value to our anomaly detector. Looking at Figure 4.3, we can see a few times where all or nearly all of the algorithms had low or high number of anomalies found. Around 3 AM we see all but average KNN going into a valley and average KNN having a decline in the number of anomalies found. Shortly after 5 AM we see a peak of anomalies found for all algorithms, with the exception of average KNN shortly after the other four. This consistent peak could be the same anomalous community seen in different announcements. We observe all algorithms show high variations in the number of anomalies found from hour to hour. Specifically, HBOS exhibits between 20 to 50 anomalies over the day, with maximum of 80 seen at (UTC) midnight. In conclusion, the number of anomalies that occur during certain hours appears erratic across all of the algorithms used.

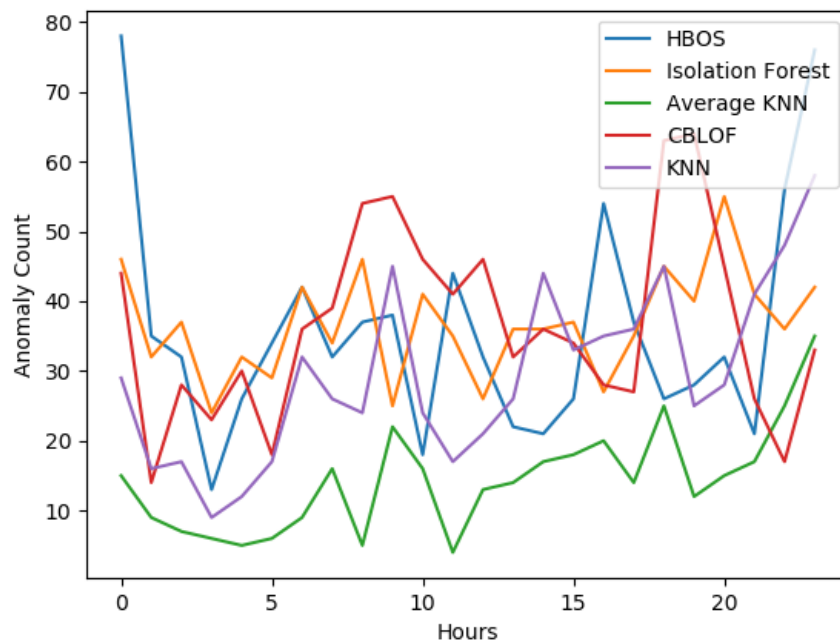


Figure 4.3. Algorithm Time Comparison (Route-Views 2)

The Figure 4.4 below shows the cumulative number of anomalies found ordered by the total found by each algorithm. This will serve as reference when comparing the cumulative number of anomalies found by more than one algorithm. Figure 4.5 shows us the number

of anomalies that one algorithm found that was also found by another algorithm.

This gives us an idea how many different anomalies each algorithm is finding, as well as how many anomalies they are finding that is the same. As noted earlier average KNN did not find any different anomalies than KNN, while this does not affect our total number of unique anomalies found by all algorithms, it does affect our cumulative graph and thus we decided to remove average KNN to get a better comparison with the other algorithms. While CBLOF found the most anomalies, in Figure 4.5 we can see the majority of the anomalies it found were different than the rest of the algorithms. We see in Figure 4.5 that there were approximately 25 anomalies found by at least one of the other three algorithms. The HBOS algorithm was in the middle for finding anomalies, but found the most anomalies at approximately 165 that were also found by one of the other three algorithms. In the middle, the KNN algorithm found approximately 65 and the Isolation Forest algorithm found about 130 that were found by at least one of the other three algorithms. We can conclude two different things from Figure 4.5, that CBLOF found the most anomalies different from the other algorithms, and HBOS found the most anomalies also found by other algorithms.

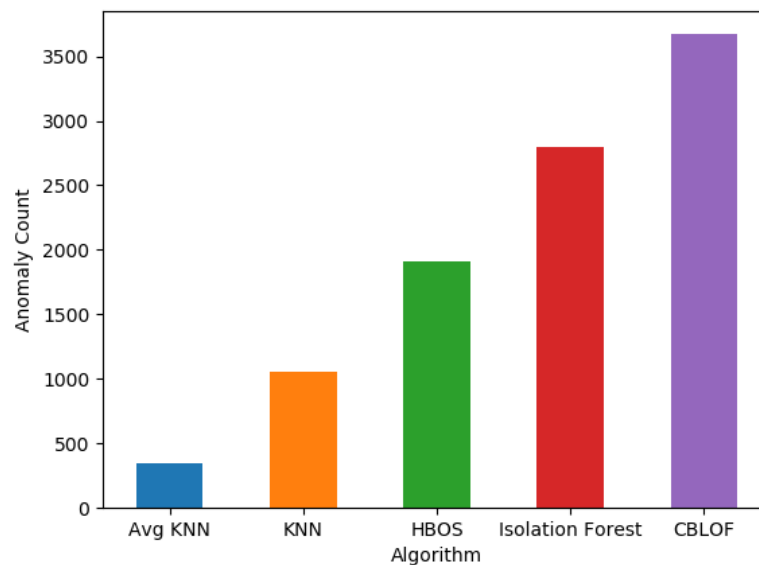


Figure 4.4. Cumulative Anomaly Count by Algorithm (Route-Views 2)

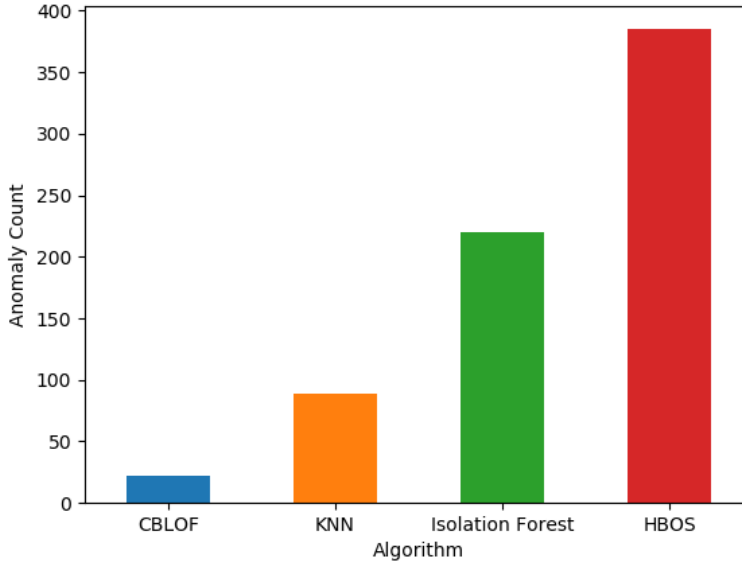


Figure 4.5. Cumulative Anomaly Count for Same Anomalies Found (Route-Views 2)

## 4.2 Collectors and Time

Next, we wish to answer the following initial questions from our data. In Section 4.1 we focused on comparing the differences and performance of the algorithms we used. In this section we will focus on the different looking glasses we used and how they compare.

### 4.2.1 How many announcements contain anomalous communities?

Table 4.2 shows us the total number of anomalies found by each looking glass, aggregated over all algorithms, on January 2, 2020. The percentage of anomalies found is fairly consistent with all four looking glasses with a slight drop for Route-Views 3, but it still yields us a few thousand anomalies to examine. While we have no way to tell how many false positives or false negatives we have, the number of anomalies found by our detection methodology is manageable to parse through to get an idea of what the anomalous communities we found look like. Depending on the looking glass, the algorithms find between a hundred to over a thousand anomalies each. The total shown in Table 4.2, is the number of unique anomalies

found by all algorithms. Given the low percentage of anomalies found, our anomaly detector is shown to be more useful by minimizing the number of false alarms. This sensitivity is partially attributed by how we parameterized our algorithms and the features used. Thus, changes could be made to adjust this sensitivity to better suit the looking glasses needs.

Table 4.2. Anomaly Counts

| Looking Glass:                 | Route-Views 2 | Route-Views 3 | Route-Views 4 | AMS-IX    |
|--------------------------------|---------------|---------------|---------------|-----------|
| Anomaly Count:                 | 3124          | 4442          | 1957          | 1829      |
| Anomaly Percent:               | 0.04%         | 0.03%         | 0.03%         | 0.04%     |
| Total Number of Announcements: | 8,799,328     | 16,818,673    | 6,063,154     | 4,848,228 |

#### 4.2.2 Do more anomalies occur at a certain time of day?

Next, we want to understand when anomalies occur and if there is any patterns to when they occur. By understanding this we might be able to discern what is causing the anomaly or even have the ability to predict when future anomalies will occur. We should note BGP collectors sees updates from around the world, so the local time at the BGP collector may not be that important and we would expect to see anomalies throughout the day. To get a better idea of when anomalies occur we graphed the frequency of anomalies found by all algorithms on 1/2/2020 for each looking glass. For each hour, we plot the number of anomalous communities found scaled by the number of updates found in that same hour. By scaling this way we can rule out an increase in a number of anomalies due to an increase in the number of updates.

Taking a look at Figure 4.6, we see a large spike of 0.12% anomaly share shortly after hour 15. All time is in UTC time, but AMS-IX is located in another country and thus BGP updates and anomalies may occur at different times. With the exception of some spikes, all looking glasses exhibit an anomaly share of 0.02% to 0.04%. We can speculate that during this time AMS-IX had a maintenance period, but something malicious may have also occurred. The rest of the graph is fairly flat, with one common peak for all four looking glasses shortly after 5 AM. This is potentially an anomaly seen by all four looking glasses

in some form. Meaning whatever event happened at that time, was able to be seen by all four looking glasses. This event could potentially be an attack or just an AS performing a maintenance period seen by all looking glasses. The edges of the graph at hours 0 and 23 have large increases in anomalies, but we feel this is only an artifact of our algorithm and data. In hour 0, our model first has to build its database, everything is initially considered a new community as the database is initially empty, as it builds we see the decrease in anomalies. Hour 23, has a similar problem, most likely this increase is due to our data set time frame starting at 23:45 of the day before. Overall there seems to be no significant differences across multiple looking glasses.

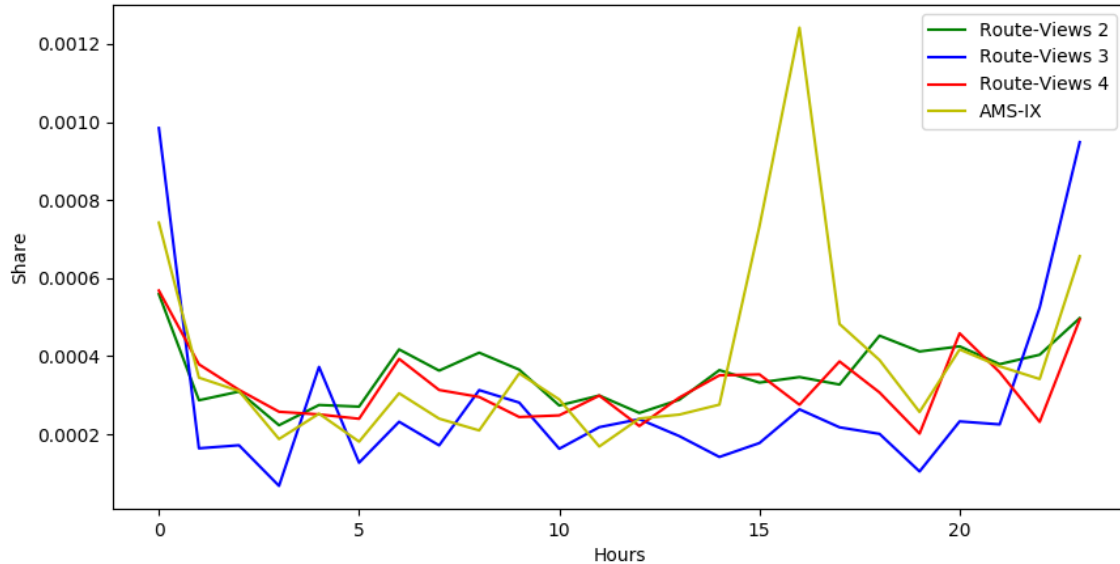


Figure 4.6. Looking Glass Anomaly Frequency Comparison

### 4.2.3 Do anomalies occur at the same time every day?

To see if there is any anomaly consistency across multiple days, we examined five consecutive days of data for Route-Views 2, as well as one day from 2016. In Figure 4.7 we notice a large spike for 1/4/2020 a shortly after 10 AM, with 1/6/2020 also having a large spike shortly thereafter. We can speculate that this is due to a policy change, misconfiguration, or malicious attack, as any one of these events could trigger a response from multiple ASes causing multiple anomalies from just one anomalous announcement.

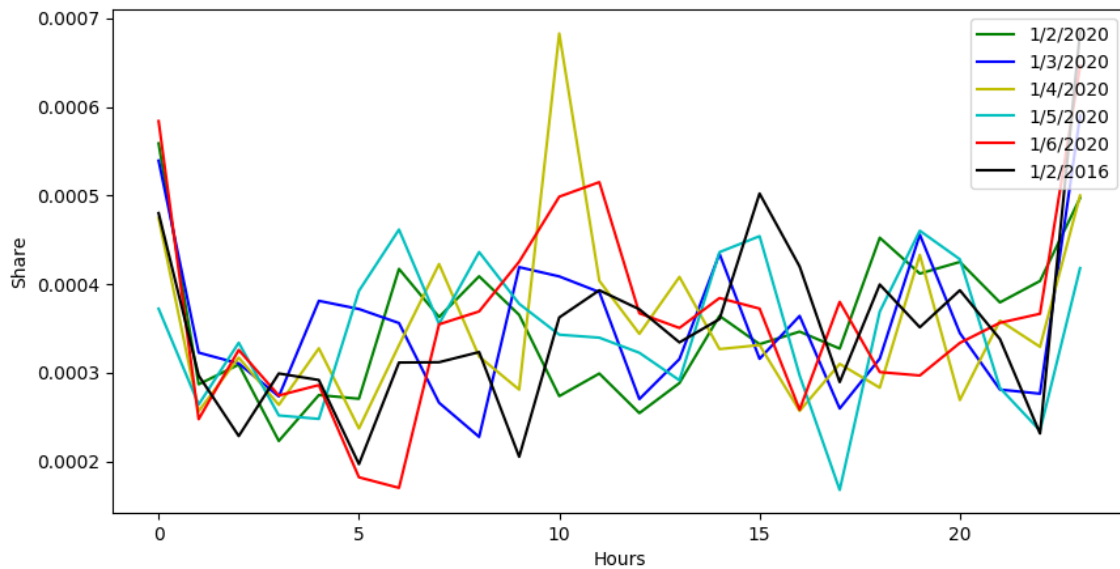


Figure 4.7. Day Comparison (Route-Views 2)

With the exception of 1/2/2016, the peak around 2 AM seems to line up with all of the days from 2020. While not a large peak on the graph, it is interesting that all the days from 2020 seem to line up as if something is consistently happening around 2 AM. This could be a maintenance period for some ASes thus triggering this consistent peak in anomalies around that time. With additional information, we may be able to use this peak in the future to predict an increase in anomalies are going to occur around 2 AM. We should note that there may be some correlation here, but without any additional information, there is too little to deduce any causality.

The valley that occurs around hour 17 is interesting as it is consistent with all but 1/4/2020 and 1/6/2020. This consistent valley shows us there may be a slow period for any policy changes or community changes that result in an anomaly around this time. Perhaps this is due to peak use or some other reason for this lack of anomalies.

While a few hours in the day showed some consistency between days, the majority of the time we see variation between days. We can conclude that most anomalies occur at different times during the day. There is some consistency with the total percentage of anomalies occurring each day, as each day aligned very closely with each other day even though every day was on its own scale.

#### 4.2.4 Can the same anomaly be seen from multiple collectors?

We want to see if an anomaly can be seen from multiple looking glasses. This is more a test of anomaly propagation than testing our anomaly detector. This is because any AS along the AS path can drop or change the communities. Thus, while an anomaly can be seen by one looking glass, by the time that same announcement reaches another looking glass an AS may have changed or dropped the anomalous community.

We compared anomalies found on Route-Views 2 against Route-View 4 joining on the prefix and community field. We decided to join on the prefix to ensure we were looking at the same announcement. If we were to join on other columns such as peer, AS path, and time stamp, all of these would be completely different from another looking glass as they change with each hop. We were able to find 33 anomalies that can be seen from both looking glasses, almost 2% of the anomalies found at Route-Views 4. We can see the time stamps happened within a second or less of each other which helped us confirm we were looking at the same announcements. We believe this low percentage is due to a small change in the community set. We match on both the prefix, and the community set to ensure we are looking at the same update, because of this any small change to the community set will result in not matching the other collector. Given more time we believe it would be interesting to examine an anomaly seen in one collector and not another and try to understand what changed. One of these anomalous communities can be seen many times and has a length of 134. The first five communities are “0:42 0:109 0:209 0:703 0:714”, in Chapter 5 we will examine this anomaly and other in more details.

We also compared anomalies found on Route-Views 2 against AMS-IX joining on the prefix and community field. We were able to see eight anomalies seen at both looking glasses with less .5% of anomalies found at AMS-IX. The time stamps on anomalies found with each looking glass were within a few seconds of each other. The time stamps and the number anomalies found goes with the fact that the two looking glasses are 5,000 miles away and most likely have to traverse more ASes than Route-Views 2 compared to Route-Views 4. A quick summary of these anomalies finds a range of feature values, with community lengths from 5-23, *Prevalence* and *Persistence* scores from 14-79%, *Foreign* from 0-51%, and *Change* from 13-100%. Given time constraints and the number of anomalies to analyze, we decided to focus on the top anomalies found by multiple algorithms, and will leave these anomalies for future work.

In Figure 4.8, we show the cumulative anomaly account for all four looking glasses, which are ordered by the number of anomalies found at each looking glass. Route-Views 3 has the most anomalies found by a good margin, but when we compare this Figure, to Figure 4.9 we see it does not have the most anomalies found by other looking glasses. A cumulative total of 320 anomalies could be seen by more than one looking glass. The looking glass Route-Views 4 had the second least amount of anomalies, yet has the most anomalies found in other looking glasses. We find AMS-IX has both the least amount of anomalies and the least amount found by one of the other three looking glasses. As mentioned earlier AMS-IX is positioned much further away and most likely has more ASes in between it compared to the rest.

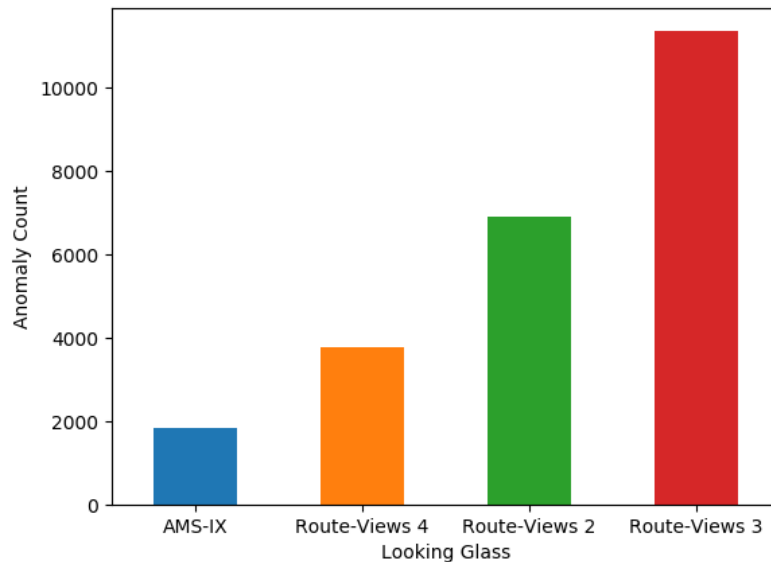


Figure 4.8. Cumulative Anomaly Count by Looking Glass

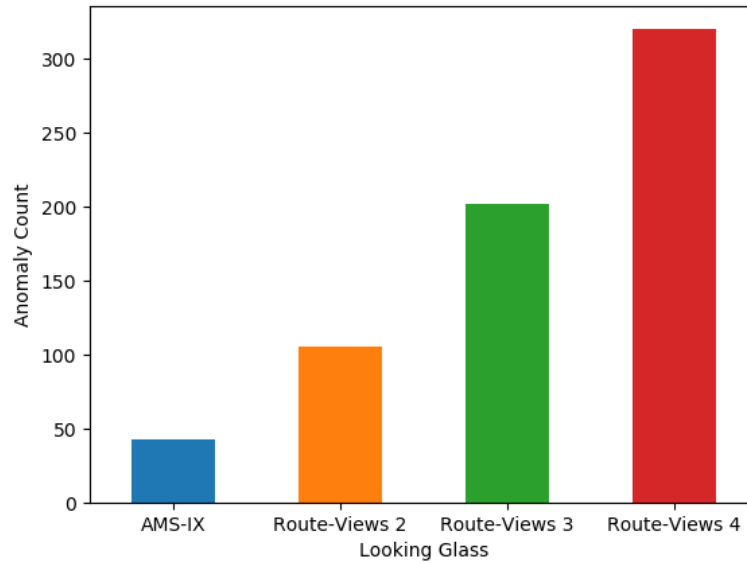


Figure 4.9. Cumulative Anomaly Count by Looking Glass for Same Anomaly

By finding the same anomalies from different looking glasses, we have some level of reassurance that our anomaly detector is consistent across different looking glasses. If labeled data was obtained this consistency could be further tested.

We have discovered anomalies are not limited to a single looking glass, showing that some anomalies can propagate to multiple looking glasses. While it is already known communities can propagate much farther than their intended target, this reinforces the impact an anomalous community can have across the internet and the importance of an anomaly detector.

#### 4.2.5 Can the same anomaly be seen on different days?

We compared anomalies found on 1/2/20 and 1/3/20 for the looking glass Route-Views 2, joining on the prefix and community field. We found nearly 500 anomalies with the same prefix and community from both days. These anomalies were found at different times each day throughout the day. This shows the same anomalies consistently appear across multiple days and some may persist for hours or possibly days. We found a wide range of different

communities and feature values. The “0:42 0:109 0:209 0:703 0:714...” community found in the join from Route-Views 2 and Route-Views 4 was also present. In addition to this community, we will also do a deeper analysis in Chapter 5 on the community “3549:2018 3549:30840 3356:3 3356:86 3356:575...”, which is a top community found by multiple algorithms. As mentioned before due to constraints we are not able to analyze all the anomalies found.

In Figure 4.10 below, we show the cumulative anomaly count across five consecutive days and include one from 2016. The ordering of days is by the amount of anomalies found on that day, as we can see 1/2/2016 has considerably less anomalies found. Moving on to Figure 4.11, we can see the cumulative anomaly count for anomalies found in more than one day, with a cumulative total of 4,782 anomalies. Most notably, no anomalies with the same prefix, and community found on 1/2/2016 were found on any other day. As mentioned in this thesis before, community use changes over time leading to different anomalies. The only change we see in order is the swap between 1/2/2020 and 1/3/2020, meaning the day 1/3/2020 had the most anomalies also found in other days. We also notice a larger gap between 1/5/2020 and 1/6/2020, with 1/6/2020 having a significant increase in anomalies. This data show us, that many anomalous communities continually appear across multiple days.

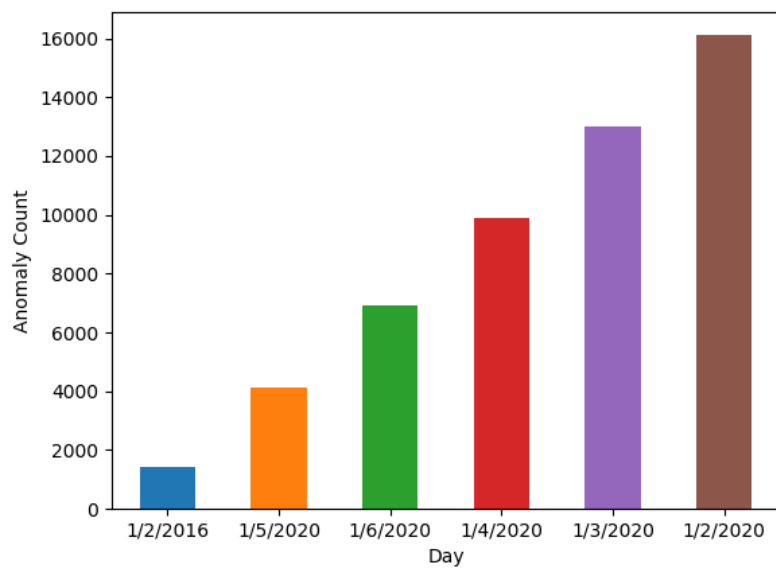


Figure 4.10. Cumulative Anomaly Count by Day

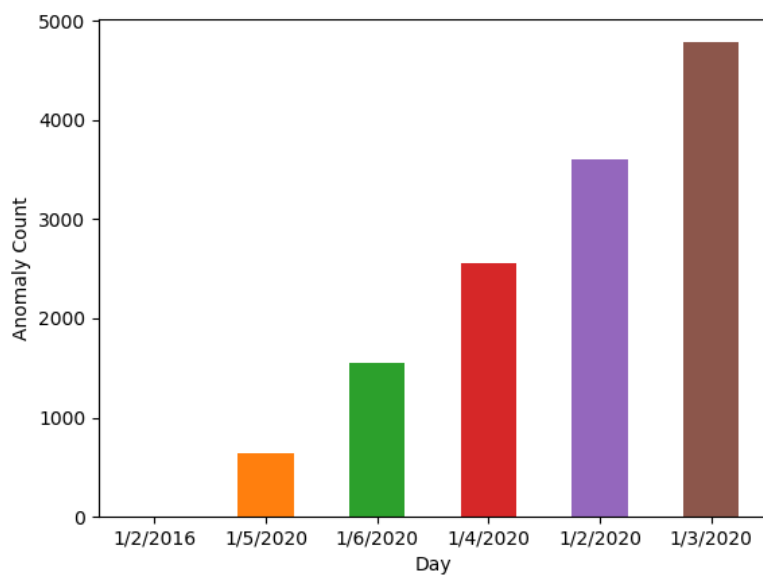


Figure 4.11. Cumulative Anomaly Count by Day for Same Anomaly

## 4.2.6 How long do anomalies persist for?

Once an anomaly is found we then look forward in time at the same prefix-ASPATH-peer tuple to see if the community in the anomalous announcement changed. If a change in communities after the anomalous announcement was discovered we then take the difference in timestamps to see how long the anomaly persisted for. In Table 4.3 we show how long anomalies persist for each looking glass on 1/2/2020. We discovered the average time an anomaly persists for is 30 to 50 minutes depending on the looking glass. We also discovered 15% to 41% of anomalies either never changed the community after the anomalous community was first announced or persisted past the 24-hour time frame of our data set.

Table 4.3. Time Length Anomalies Persisted

| Looking Glass:                            | Route-Views 2 | Route-Views 3 | Route-Views 4 | AMS-IX  |
|-------------------------------------------|---------------|---------------|---------------|---------|
| Average Time Persisted (Secs):            | 1870.72       | 3001.91       | 2663.91       | 2100.48 |
| % of Anomalies Persisted past Time Frame: | 15.4%         | 20.7%         | 41.4%         | 38.7%   |

To better understand the distribution of times, we plot the times in a CDF graph in Figure 4.12. The X-axis is in log form to better differentiate changes in time and see the lines for each looking glass. It should be noted, Figure 4.12 does not include the percentage of anomalies that persisted after the 24-hour time frame and the max time in this figure is artificially capped because of our time frame. Looking at just the anomalies that changed within the 24-hour time frame, about 70% persisted for less than 10 or 100 seconds depending on the looking glass before changing. Thus, based on the anomalies that do change within our time frame the majority of the anomalies only persist for less than a 100 seconds. Additionally, 10-40% persisted for a second or less depending on the looking glass.

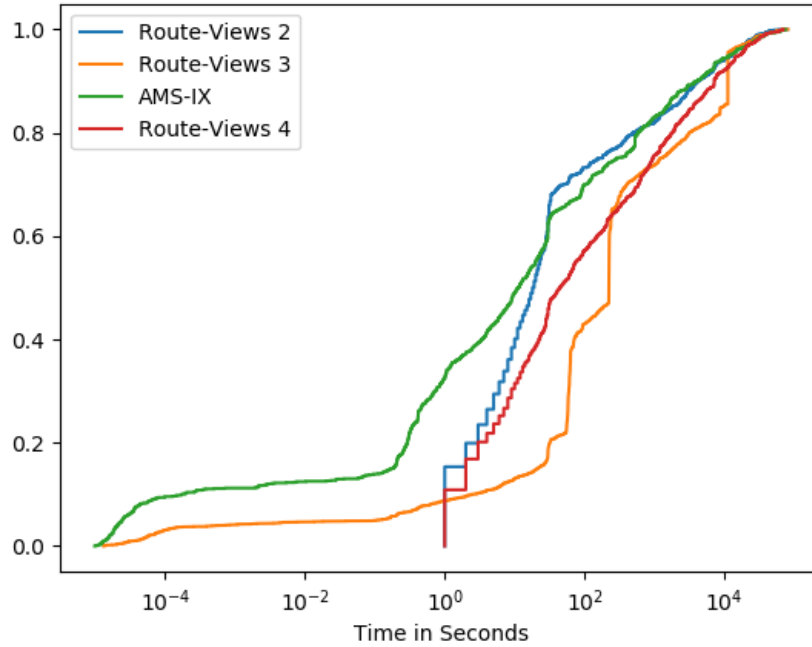


Figure 4.12. CDF: Time Length Anomalies Persisted

Comparing the anomaly persistence time to the *Time Since Change* feature discussed in Section 3.3.1, we can see a similar percentage of updates that persisted for a second or less. In contrast the average *Time Since Change* is significantly longer time, than how long anomalous communities persist before changing. In comparison between the two, we would conclude on average an anomaly will persist for a much short time then the time for a non-anomalous community, but with a clause that up to 40% of anomalous and non-anomalous updates may persist for second or less.

The percentage of anomalies that persisted past our time frame could be for a variety of reasons including new communities being seen for the first time, a policy change resulting in an abnormal community change, an artifact of our time frame, or some other reason. We expect some anomalies will always persist outside a given time frame, but this does not explain the variation from 15% for Route-Views 2 and 41% for Route-Views 4. In the future work section of Chapter 5 we will talk about how we would like to explore how long anomalies persist for.

THIS PAGE INTENTIONALLY LEFT BLANK

---

## CHAPTER 5:

### Discussion

---

In this chapter, we will take the anomalies found by multiple algorithms, and look at the top few anomalies in common among all algorithms. Our intuition, is that if multiple algorithms find an anomaly, then it is more likely to be a true anomaly. We will perform a deep analysis on anomalies that have uniquely different communities. We refer to uniquely different as community sets don't share individual communities or the communities themselves are only different by a small difference of one or two. The community sets we focused on have both different individual communities and almost all of the feature values are different.

At the end of this chapter we will summarize our findings and discuss the contributions made. We will then conclude this thesis with our recommendations for future work.

### 5.1 Anomaly Analysis

In this section, we look at the top few anomalies found for Route-Views 2 on 1/2/2020. When we refer to the top anomalies we are referring to identical anomalies found in common by the most algorithms. The set of top anomalies are different from one another, but represent the same anomaly detected by multiple algorithms. It should be noted, that only the features engineered were used in the machine learning process for all algorithms. Thus, the community set itself was not an input to the algorithms.

The following Table 5.1, will be used for reference when analyzing anomalous communities. The first row is only the first five communities as many of the anomalies found had more than 135 unique communities in the community attribute. We chose the following three anomalous communities to examine because they were both the top anomalies found and contained unique differences from one another.

Table 5.1. Top 3 Unique Anomaly Comparison

|                               |                                                                                |                                                                                              |                                                                                                |
|-------------------------------|--------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <b>Community:</b>             | <b>0:42</b><br><b>0:109</b><br><b>0:209</b><br><b>0:703</b><br><b>0:714...</b> | <b>0:0</b><br><b>1299:2873</b><br><b>1299:2883</b><br><b>1299:5001</b><br><b>1299:5873..</b> | <b>3549:2018</b><br><b>3549:30840</b><br><b>3356:3</b><br><b>3356:86</b><br><b>3356:575...</b> |
| Algorithm Count:              | 5                                                                              | 4                                                                                            | 4                                                                                              |
| Time stamp:                   | 1/2/2020<br>12:15                                                              | 1/2/2020<br>2:57                                                                             | 1/2/2020<br>15:30                                                                              |
| Time Since Change(Secs):      | 0                                                                              | 0                                                                                            | 32403                                                                                          |
| Prevalence:                   | 33.33%                                                                         | 50.00%                                                                                       | 16.67%                                                                                         |
| Persistence:                  | 33.33%                                                                         | 50.00%                                                                                       | 8.33%                                                                                          |
| Foreign:                      | 94.78%                                                                         | 67.44%                                                                                       | 15.79%                                                                                         |
| Change:                       | 98.51%                                                                         | 9.30%                                                                                        | 15.79%                                                                                         |
| Seen For Prefix:              | No                                                                             | No                                                                                           | Yes                                                                                            |
| Seen For AS:                  | No                                                                             | No                                                                                           | Yes                                                                                            |
| Seen Ever:                    | No                                                                             | Yes                                                                                          | Yes                                                                                            |
| Blackhole:                    | No                                                                             | Yes                                                                                          | Yes                                                                                            |
| Community Length Change:      | 126                                                                            | 1                                                                                            | 0                                                                                              |
| RIR:                          | ARIN                                                                           | RIPE                                                                                         | ARIN                                                                                           |
| Granularity of Change:        | 0                                                                              | 100,233                                                                                      | 136,516                                                                                        |
| Time Anomaly Persisted(Secs): | 13,284                                                                         | N/A                                                                                          | 28                                                                                             |

### 5.1.1 First Anomaly

The first anomaly we will take a look at was found by all five algorithms. The first ten communities in the anomaly are “0:42 0:109 0:209 0:703 0:714 0:1273 0:1828 0:2497 0:2559 0:2635”. We only show the first ten as this community attribute has a length of 134 communities. The length of this community attribute increased by 126 over the previous announcement. The first two things that stand out by this anomalous community is the sheer length of the community which is rare. The second is the use of “0:X” which contributes to

the 94.78% *Foreign* score.

If we look at just the feature values it is easy to tell why it stood out as an anomaly. In addition to the *Foreign*, and *Community Length Change* we see both a low 33.33% *Prevalence* and *Persistence* scores in Table 5.1. Meaning for this prefix-ASPATH-peer tuple the community has changed frequently and has a low persisting community. We can see the *Time Since Change* is zero, meaning that the announcement before this was less than a second before this anomalous announcement was received. This anomalous announcement then persisted for 13,284 seconds or 3 hours and 41 minutes before another change occurred. The *Seen Ever* feature is zero, meaning for all the announcements this looking glass has received in our data set before this announcement, it contains an individual community within its community set that it has never before. The *Granularity of Change* is zero and the *Change* is 98.51%. This means that none of the communities in the first announcement had the second part after the “:” change, either all of the change was due to the additional communities or the entire individual community changed.

When analyzing a community, we first look at the peer to see if they have any BGP community policy, and then we work our way down the AS path. Our goal is to try to figure out who may have added the anomalous communities and what their purpose is. The peer sending the update for this first anomaly is AS2914 NTT, which lists all of the BGP communities they use in their policy. The “0:X” is not listed as one of them. The next AS on the AS path is AS20473 Choopa LLC also lists their BGP community policy, but does not use the “0:X”. The final AS on the AS path is AS208393 Dedcloud Network, which does not list any policy information online for BGP communities.

Now that we have gone through the entire AS path with no leads, we will take a closer look at the individual communities in the community set for any clues. The second to last community looks promising as it does not start with zero and is not an AS in our AS path. The community is “24115:24115”, the “24115:” is a valid AS number and is AS24115 Equinix. Equinix does not list their BGP community policy on their website, but through some digging we were able to find some slides from 2018 showing their BGP community policy [30]. The “24115:24115” means announce to all. The “0:X” means except to X where X is the AS number you don’t want to announce to.

The only direct peers with Equinix is AS20473 Choopa, but this is not to say the communities

could have been leaked through other peers, or AS208393 Dedcloud Network is using a similar policy as Equinix [31]. There are many possibilities due to the transitive nature of BGP communities. None the less this does give us more information to better understand what the anomalous BGP community string is doing in terms of policy. We can say it is most likely non-malicious and not a misconfiguration unless we classify leaked BGP communities as a misconfiguration.

### 5.1.2 Second Anomaly

The second anomaly we chose to look at was found by four algorithms. Only two anomalies for Route-Views 2 were found by all five algorithms. One shown in the section above, the other being very similar to the first anomaly we analyzed. Thus, we decided to skip it and look at this community “0:0 1299:2873 1299:2883 1299:5001 1299:5873 1299:5883 1299:7009 1299:7873 1299:7883 2223:36351”, which is the first ten communities in the set. The length of this community is considerably less than the first anomaly analyzed, but at the length of 43 it is still much longer than the majority of communities attributes commonly seen. The community set has only increased by one meaning the community attribute in the announcement before this one had a length of 42 for this same prefix-ASPATH-peer tuple. A unique community in the set that first stands out is the “0:0” community. While in the first anomaly analysis we looked at communities that started with “0:X” we did not see a community where X was zero. Thus, in this case “0:0” may have a different use than in the first anomaly.

Looking at Table 5.1, we see the *Time Since Change* score of zero, telling us this announcement followed the previous announcement in less than a second. The *Prevalence* and *Persistence* scores are both fairly low sitting right at 50%, we can speculate and say this may have been the second announcement seen for the prefix-ASPATH-peer tuple. None of the first ten communities are on the AS path, and this holds true for most of the community set giving us a *Foreign* score of 67.44%. The *Change* percent is low at 9.30%, yet the *Granularity of Change* is 100,233. This tells us that only a few communities in the set changed, but numerical difference in the change was quite large. The *Seen Ever* flag is yes, while this community has never been seen for this prefix or AS the looking glass has seen this community before. This community set contains use of the *Blackhole* community, used in this community “3223:666”, where AS3223 is on the AS path. The *Time Anomaly Persisted*

is marked “N/A”, meaning from the time this announcement was seen at 2:57 AM it either did not change and persisted or another announcement for the same prefix-ASPATH-peer tuple was not seen inside our 24-hour time frame.

Starting with the peer on the AS path we look at AS3303 Swisscom, to look for BGP community policies. Swisscom does not have any publicly available information regarding their policies for BGP communities. The next AS on the AS path is AS2914 NTT, which lists their policy online, but only explains these four communities “2914:410 2914:1201 2914:2202 2914:3200”. This does not help explain the anomalous communities in the set that we are more interested in. These communities are just information that the route is a customer route and its origin. The next AS on the AS path is AS3223 Voxility, which does not list any information online for their BGP community policy. The final AS on the AS path is AS134548 DXTL Tseung Kwan O Service, which has its AS number prepended three times at the end of the AS path. The DXTL Tseung Kwan O Service AS is based out of Hong Kong and does not list any information about its organization online.

Now that we have looked through the entire AS path we will search through the community set for answers. The “1299:X” community is seen several times, and is AS1299 Telia Carrier. Telia Carrier lists their BGP community policy, and goes with some of what we are seeing. The community “1299:2873” is for China Unicom path prepending where the last number in the second tuple is how many time to prepend, in our case three [32]. This information helps explain the “1299:X” communities, but does not explain how it got there or who may have added it as it is an off path community.

The next community we want to take a look at is the “6762:X” community which is seen five times. The AS6762 TELECOM ITALIA SPARKLE, lists all of their BGP communities. The community “6762:18500” means the route was learned in Hong Kong [33]. This goes with the last AS on the AS path originating in Hong Kong. Again we have some explanation as what the individual communities mean, but not why they are in the community set or who may have added them.

The next community we want to examine is “22223:X” which is seen five times. Unfortunately the AS2223 MVTW Wireless does not list their policy for BGP communities. This leads us to the next community “22229:X”, which is found seven times in the set. The AS22229 WORLDPAC does not list any BGP community policy online. It should be noted

that we looked at the first part of the community tuple as an AS number as this is traditionally how it is used. It could also be the case that both “22223:X” and “22229:X” communities are not referring to AS numbers, even though those happen to be legitimate AS numbers.

We have an idea what most of the individual communities mean for this second anomaly analysis, but we do not know why all of these communities are present. Our theory is some peer or AS on the AS path is leaking these off path communities rather than stripping them out or dropping the announcement. This anomalous community is more difficult than the first to say why it is an anomaly. It could be a misconfiguration, malicious, routine maintenance, or something else.

### 5.1.3 Third Anomaly

The third anomaly we choose to look at was found by four of the five algorithms. The first ten communities in the community set are “3549:2018 3549:30840 3356:3 3356:86 3356:575 3356:666 3356:2011 3356:11918 3491:400 3491:412 ”. The anomalous community has a length of 19, and has a *Community Length Change* of zero. All of the first ten communities have an AS on the AS path, which continues for most of the community set yielding a low 15.79% *Foreign* score.

This anomalous community has a very low *Prevalence* score of 16.67% and a *Persistence* score of 8.33%. This shows the community for the prefix-ASPATH-peer tuple is experiencing a lot of changes with no clear dominant community. Looking at Table 5.1, we can see the *Time Since Change* is 32,403 seconds or about 9 hours. Then it only lasted for 27 seconds before some other change occurred. This community has been seen for all of the *Seen For* features, thus it may not be a rare community for this prefix-ASPATH-peer tuple. The *Change* score is 15.79% with a *Granularity of Change* of 136,516. Given the length only of the community set, we can say three of the communities experienced a large change. The *Blackhole* community is also seen in this set as “3356:666”. As mentioned in Chapter 2, we discovered AS3356 uses “666” to communicate a peer route.

The individual communities in this announcement are not what makes in an anomaly, as we will show most of these communities can be explained. What makes this BGP community announcement an anomaly is how often we are seeing it change and how erratic the timing is of these changes appears to be. The behavior of the community is not what we typically

see. The peer AS3549 GBLX has their BGP community policy listed online. We looked at the first two communities “3549:2018” and “3549:30840”. We don’t see a policy listed for the “3549:2018”, but we see “3549:30840” means the route is sourced from the U.S. [34]. The next AS on the AS path is AS3356 Level3, which as mentioned before lists their policy online. We can use this policy to get an idea what the next few communities in the set “3356:3 3356:86 3356:575 3356:666 3356:2011 3356:11918” mean. Most of these communities are discussing the routes origin, with the exception of the “666” meaning peer route and the “11918” not being listed [35]. The next AS on the AS path AS3491 PCCW Global lists its policy online as well. Which we can use to decipher the following communities “3491:400 3491:412 3491:62210 3491:62220”. This time only the “400” had a policy listed as a customer in the Asia region [36].

The last four ASes on the AS path did not have a policy listed, for which we could use to better understand the follow communities “9498:1 9498:91 9498:9333 9498:58717”. This leaves us with foreign communities “34111:9498 34911:9498 40512:9498”, none of the ASes that had their policy listed had a policy for these three communities. We tried looking up the first part of the tuple as an AS as we did for the first two anomalies, but none of these ASes had policies listed. While we are able to explain many of the communities used in this community set, but due to the lack of information we cannot explain all of the communities. Without labeled data, we cannot say if it is a true anomaly, but we can say it is exhibiting abnormal behavior based on the features we have engineered. We could speculate what is the cause of this announcement, but like the first two anomalies without ground truth there could be a variety of reasons causing its behavior.

## 5.2 Conclusion

Our main goal of this thesis was to determine the feasibility of an anomaly detector. During this process we studied how BGP communities are being used from multiple vantage points across multiple days. We examined various statistics and engineered multiple features to help understand BGP community behavior. We discovered:

- There is significant use of *Foreign* communities, which we show ASes using *Foreign* communities in their policy as routine operations. This illustrates the lack of standardized implementation and evolving use of BGP communities.

- The majority of communities do not change, however when a change does occur on average only 10-15% of the communities in the community set change. This change to all individual communities is on average a total difference of one to two thousand.  
j
- The *Blackhole* community is found in 2-8% of updates received. Many ASes are using the *Blackhole* community in their day-to-day operation, not only for dropping traffic to a specified prefix, but for other purposes such as communicating a peer route.

### 5.2.1 Contributions

In the process of creating an anomaly detector we discovered the number of updates that contain anomalous communities in Section 4.2.1. This section also, highlights the percentage of anomalies across multiple looking glasses, and the consistency of anomalies from different looking glasses.

In Section 4.2.2, we examined the frequency of anomalies across multiple looking glasses and days for a single looking glass. We found that there are no significant differences across multiple looking glasses. We did discover some consistency across multiple days, but not enough consistency without ground truth to show correlation.

In Sections 4.2.4 and 4.2.5 we examine anomalies from two different looking glasses, and separately examine two different days for the same looking glass. We have shown, that the same anomaly could both be seen from two different looking glasses and found by our anomaly detector. Additionally, we discover many anomalies with the same prefix and communities could be seen across two days for the same looking glass.

In our final question, we look at how much time do anomalies persist for in Section 4.2.6. We discover 15-41% of anomalies either never change after being found as anomalous or persist outside of our 24-hour time frame. This information also tells us the time that anomalies persist for outside the 24-hour time frame varies significantly between looking glasses. Up to 40% of the anomalies that do finish within the time frame only persist for a second or less. The majority of anomalies for all looking glasses persist for less than 100 seconds.

Lastly in Sections 3.4.8 and 5.1, we are able to show an anomaly detector is not only feasible, but we were able to create a working model. We found many anomalies with

multiple unsupervised clustering algorithms, and used an ensemble method to help improve our unknown accuracy. We performed case studies on the top three unique anomalies, showing their unique differences and why they were classified as an anomaly.

## 5.3 Future Work

While studying BGP communities and creating an anomaly detector, a few questions were raised that we would like to investigate in the future. We have questions on BGP behavior, and different implementation of the features we engineered. Our first question we are interested in, regards the use of the blackhole community. After discovering that some ASes use the blackhole community for purposes other than dropping traffic, we would like to know how common of practice this is. The use of BGP communities is constantly evolving, where the blackhole community was once considered by many to be used to drop traffic, now it maybe evolving in to another use. We wish to see the use of the blackhole community over a few years, to see if the use of the blackhole community is increasing given its current use.

Similarly, to the blackhole communities the *Foreign* communities are being used by ASes to implement policy in a nontraditional way. We are curious to know how many *Foreign* communities are due to intentional policy use or due to lack of policy filtering off-path communities. We also would like to examine the use of *Foreign* communities over time to see if has an increase in use. We want to study both blackhole communities and *Foreign* communities to help better understand BGP community behavior, and get a better idea of what is normal.

As mentioned in Chapter 3, the *Granularity of Change* feature presents a matching problem, and has limitations. For our implementation we matched on the first community with the same numbers before the “:”. Other implementations might match on the last community found, or match on every community found with the same first part of the tuple. Another implementation would be to take the average of all matches. In the future we would like to try various methods and test them to see their results. In our implementation we summed the absolute difference; we are curious if it would be better to take the average difference or take only the largest difference. Our main goal is to find the best way to capture exactly how much change has occurred at the finest granularity possible.

We captured *Time Since Change* in seconds, but given how quickly update messages can be

received, this may not be a fine enough granularity. We saw many updates with a *Time Since Change* of zero seconds – this could be attributed to two things. First this is the first time we have seen this community and or the first time we have seen this prefix-ASPATH-peer tuple resulting in zero time because it is the first appearance. The second contributor, may be a change is taking place in a fraction of a second, and our implementation is not capturing it. We would like to test this timing further, while we can not say if it will have much impact on our anomaly detector, it would help us better understand BGP behavior.

We found our *RIR* feature had more than one limitation, and it is hard to say how beneficial the feature is. Before just dropping the feature we would like to test the following changes. First, rather than looking at the peer, we would like to look at the origin of the route, the last hop on the AS path. While the origin may have not added anomalous communities, based on its region other ASes may have tagged communities on along the way. Second, because our implementation groups 32-bit ASes together, we would like to have finer granularity and have 32-bit regions. Finally, we feel it would be interesting to perform a deeper analysis comparing communities from different regions. If no differences can be found, then this feature does not have any added benefit and can be removed.

Given the number of unique anomalies, we were constrained to look at just a few prominent ones. A paper alone could consist of just case studies on different anomalies as there are so many of them. Given additional time, we would like to compare the top anomalies from different looking glasses. We have shown the same anomaly can be seen from two different looking glasses and we are curious if there are anomalous communities unique to each looking glass. To explain further, we are interested if there is an anomalous community that is seen in one looking glass and the community set is unique such that another looking glass has never seen a similar community set.

In the future we would like to examine a larger time frame, such as a week or even a month. Our implementation takes some initial time to build its data set of communities, prefixes, and ASes it has seen before. We would like to see, over a larger period of time, if we could improve our accuracy and see a slight decrease in anomalies found after the first day or two. We also want to get a better idea of how long anomalies persist for, as we feel having a longer time frame would give us a more accurate idea, as well as monitor other behavior of anomalous communities over time.

---

## List of References

---

- [1] Y. Rekhter and K. Lougheed, “Border Gateway Protocol (BGP),” RFC Editor, RFC 1105, Jun 1989, [Online]. Available: <https://tools.ietf.org/html/rfc1105>
- [2] Y. Rekhter and K. Lougheed, “Border Gateway Protocol (BGP),” RFC Editor, RFC 1163, Jun 1990, [Online]. Available: <https://tools.ietf.org/html/rfc1163>
- [3] R. Chandra, P. Traina, and T. Li, “BGP communities attribute,” RFC Editor, RFC 1997, Aug 1996, [Online]. Available: <https://tools.ietf.org/html/rfc1997>
- [4] F. Streibelt, F. Lichtblau, R. Beverly, A. Feldmann, C. Pelsser, G. Smaragdakis, and R. Bush, “BGP Communities: Even more worms in the routing can,” in *Proceedings of the Internet Measurement Conference 2018*. Boston MA USA: ACM, Oct. 2018, pp. 279–292, [Online]. Available: <https://dl.acm.org/doi/10.1145/3278532.3278557>
- [5] Y. Rekhter and S. R. Sangli, “BGP extended communities attribute,” RFC Editor, RFC 4360, Feb 2006, [Online]. Available: <https://tools.ietf.org/html/rfc4360>
- [6] B. Donnet and O. Bonaventure, “On BGP communities,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 55–59, Mar. 2008, [Online]. Available: <https://doi.org/10.1145/1355734.1355743>
- [7] R. Bush and R. Austein, “The Resource Public Key Infrastructure (RPKI) to Router Protocol,” RFC Editor, RFC 6810, Jan 2013, [Online]. Available: <https://tools.ietf.org/html/rfc6810>
- [8] A. Hardt, “Characterizing BGP community irregularities toward an anomaly detection engine,” M.S. Thesis, Dept. of Comp. Sci, NPS, Monterey, CA, USA, Dec. 2019, [Online]. Available: <https://calhoun.nps.edu/handle/10945/64179>
- [9] V. Fuller and T. Li, “Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan,” RFC Editor, RFC 4632, Aug 2006, [Online]. Available: <https://tools.ietf.org/html/rfc4632>
- [10] J. Hawkinson and T. Bates, “Guidelines for creation, selection, and registration of an Autonomous System (AS),” RFC Editor, RFC 1930, Mar 1996, [Online]. Available: <https://tools.ietf.org/html/rfc1930>
- [11] IANA, “Number Resources,” Jan 1983, [Online]. Available: <https://www.iana.org/numbers>

- [12] NTT, “Routing,” Jan. 2020, [Online]. Available: <https://www.gin.ntt.net/support-center/policies-procedures/routing/>
- [13] Cisco, “IP Routing: BGP Configuration Guide, Cisco IOS Release 15M&T - BGP Best External [Cisco IOS 15.4M&T],” Sep. 2019, [Online]. Available: [https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute\\_bgp/configuration/15-mt/irg-15-mt-book/irg-best-external.html](https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/15-mt/irg-15-mt-book/irg-best-external.html)
- [14] ASRank, “AS Rank: AS2914 (NTT America, Inc.),” July 2020, [Online]. Available: <https://asrank.caida.org/asns/2914/as-core>
- [15] RouteViews, “About – Routeviews,” Jan. 2020, [Online]. Available: <http://www.routeviews.org/routeviews/index.php/about/>
- [16] L. Gao, T. Griffin, and J. Rexford, “Inherently safe backup routing with BGP,” in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, Apr. 2001, vol. 1, pp. 547–556 vol.1, iSSN: 0743-166X.
- [17] L. Cogotti, “New MRT-BGP reader six times faster than its predecessors,” Nov. 2018, [Online]. Available: <https://blog.apnic.net/2018/11/29/new-mrt-bgp-reader-six-times-faster-than-its-predecessors/>
- [18] I. Zaitsev, “The Best Format to Save Pandas Data,” Mar. 2019, [Online]. Available: <https://towardsdatascience.com/the-best-format-to-save-pandas-data-414dca023e0d>
- [19] PyData, “Intro to data structures — pandas 1.1.0 documentation,” Jan. 2008, [Online]. Available: [https://pandas.pydata.org/docs/user\\_guide/dsintro.html](https://pandas.pydata.org/docs/user_guide/dsintro.html)
- [20] NIST, “1.3.6.2. Related Distributions,” July 2020, [Online]. Available: <https://www.itl.nist.gov/div898/handbook/eda/section3/eda362.htm>
- [21] Speed-IX, “BGP community filtering,” July 2020, [Online]. Available: <https://speed-ix.net/bgp-community-filtering/>
- [22] B. Decraene, K. Patel, C. Pelsser, C. Filsfils, and P. Francois, “Graceful BGP Session Shutdown,” RFC Editor, RFC 8326, Mar 2018, [Online]. Available: <https://tools.ietf.org/html/rfc8326>
- [23] Y. Zhao, Z. Nasrullah, and Z. Li, “Pyod: A python toolbox for scalable outlier detection,” *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019, [Online]. Available: <http://jmlr.org/papers/v20/19-011.html>
- [24] H.-P. Kriegel, M. Schubert, and A. Zimek, “Angle-based outlier detection in high-dimensional data,” aug 2008, pp. 444–452.

- [25] PyOD, “pyod.models.cblof — pyod 0.8.1 documentation,” 2019, [Online]. Available: [https://pyod.readthedocs.io/en/latest/\\_modules/pyod/models/cblof.html](https://pyod.readthedocs.io/en/latest/_modules/pyod/models/cblof.html)
- [26] PyOD, “pyod.models.feature\_bagging — pyod 0.8.1 documentation,” 2019, [Online]. Available: [https://pyod.readthedocs.io/en/latest/\\_modules/pyod/models/feature\\_bagging.html](https://pyod.readthedocs.io/en/latest/_modules/pyod/models/feature_bagging.html)
- [27] M. Goldstein and A. Dengel, “Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm,” aug 2012.
- [28] PyOD, “pyod.models.iforest — pyod 0.8.1 documentation,” 2019, [Online]. Available: [https://pyod.readthedocs.io/en/latest/\\_modules/pyod/models/iforest.html](https://pyod.readthedocs.io/en/latest/_modules/pyod/models/iforest.html)
- [29] PyOD, “pyod.models.knn — pyod 0.8.1 documentation,” 2019, [Online]. Available: [https://pyod.readthedocs.io/en/latest/\\_modules/pyod/models/knn.html](https://pyod.readthedocs.io/en/latest/_modules/pyod/models/knn.html)
- [30] W. Naing, “BGP Large Communities in IXPs,” Feb. 2018, [Online]. Available: <https://www.slideshare.net/apnic/bgp-large-communities-in-ixps>
- [31] BGPview, “AS24115 Equinix BGP Network Information - BGPView,” Aug. 2020, [Online]. Available: <https://bgpview.io/asn/24115#peers-v4>
- [32] Telia, “BGP Communities,” Aug. 2020, [Online]. Available: <http://www.teliacarrier.com/our-network/bgp-routing/bgp-communities.html>
- [33] Sparkle, “BGP Community support for AS6762 Customers,” Aug. 2020, [Online]. Available: <https://etabeta.noc.seabone.net/communities.html>
- [34] Onestep, “as3549 | One Step,” Aug. 2020, [Online]. Available: <https://onestep.net/communities/as3549/>
- [35] Onestep, “as3356 | One Step,” Aug. 2020, [Online]. Available: <https://onestep.net/communities/as3356/>
- [36] Onestep, “as3491 | One Step,” Aug. 2020, [Online]. Available: <https://onestep.net/communities/as3491/>

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