Bigfoot: A Geo-based Visualization Methodology for Detecting BGP Threats

Meenakshi Syamkumar
ms@cs.wisc.edu
University of Wisconsin - Madison

Ramakrishnan Durairajan
rkrish@cs.wisc.edu
University of Wisconsin - Madison

Paul Barford
pb@cs.wisc.edu
University of Wisconsin - Madison and comScore, Inc.

Abstract

Studies of inter-domain routing in the Internet have highlighted the complex and dynamic nature of connectivity changes that take place daily on a global scale. The ability to assess and identify normal, malicious, irregular and unexpected behaviors in routing update streams is important in daily network and security operations. In this paper we describe *Bigfoot*, a Border Gateway Protocol (BGP) update visualization system that has been designed to highlight and assess a wide variety of behaviors in update streams. At the core of Bigfoot is the notion of visualizing the announcements of network prefixes via IP geolocation. We investigate different representations of polygons for network footprints and show how straightforward application of IP geolocation can lead to representations that are difficult to interpret. Bigfoot includes techniques to filter, organize, analyze and visualize BGP updates that enable characteristics and behaviors of interest to be identified effectively. To demonstrate Bigfoot's capabilities, we consider 1.79B BGP updates collected over a period of one year and identify 139 candidate events in this data. We investigate a subset of these events in detail, along with ground truth from existing literature to show how network footprint visualizations can be used in operational deployments.

Keywords: BGP Security; Routing anomalies; Visualization.

Index Terms: •**Human-centered computing** → **Geographic visualization;** •**Information systems** → *Location based services; Geographic information systems;* •**Computing methodologies** → *Anomaly detection;*

1 Introduction

The transmission of reachability information (updates) between autonomous systems (AS) via the Border Gateway Protocol (BGP) is one of the most basic aspects of Internet operation. BGP updates convey information about IP address space availability, AS path and other associated attributes. The *AS path* attribute lists the ASes traversed to reach the announced address space. These updates enable policy-compliant routes between networks to be established and maintained. However, the enormous volume and diversity of BGP updates, the possibility of malicious behavior and the lack of global coordination between participants present significant challenges to network operations on a daily basis.

To build a foundation for understanding and improving BGP, update streams have been analyzed extensively in prior work. Early studies such as [40] helped to elucidate latencies in routing convergence after failures, while [48] examined the impact of route updates on network traffic. Prior studies have also developed methods for automating analysis of update streams. For example, a number of studies have described tools that assist in the process of identifying reachability problems and determining root-causes for BGP routing changes (*e.g.*, [36, 49, 57]). While such tools are useful in network operations, there continue to be outages, misconfigurations and attacks that can have a significant impact on service quality. This calls for new methods for monitoring and analysis that can reveal both expected and unexpected behaviors in BGP update streams.

Graphical visualization is a well known method for assessing large, complex data sets. Visualizations are particularly useful for outlier detection. In this paper, we present a new method for visualizing and analyzing BGP updates. The goal of our work is to develop a capability that can be used in both research and operations to identify and assess normal and anomalous activity in BGP updates. The requirements for the system include the ability to ingest a large volume of updates, winnow activity to specific networks or regions of interest and highlight a variety of behaviors that are of interest in security and operations (*e.g.*, hijacks, black holes, DDoS attacks, etc.).

At the core of our approach is the application of *IP geolocation* to the network address prefixes that are included in BGP updates. Our hypothesis is that polygons that emerge from drawing a border around geolocated IP addresses projected on a world map are intuitive representations that provide insights on a range of behaviors that are important to network operations. At first glance, the task of drawing such polygons might seem to be straightforward. However, we show that simple approaches to drawing polygons result in irregular shapes that preclude applications in target use cases.

We address this challenge by developing a methodology for visualizing network address space in 2-D polygons projected on world maps, which is the first contribution of our work. Our approach is based on drawing convex hulls around a subset of IP address locations associated with a given prefix. The objective of subset selection is to produce polygons are the unique and include characteristics that are relevant to target use cases. Since geolocation of a given IP address can be inaccurate [35]), the subset selection process remove outliers that would otherwise inappropriately distort the resulting polygon. However, this must be done with care so that meaningful points are not inadvertently removed.

We implement our network polygon generator in a system that we call *Bigfoot*[1], which is the second contribution of this paper. Bigfoot includes three main components. The first is a data processing engine that ingests static files or streaming BGP updates and generates geopositioned 2-D polygons that represent network address prefixes. The second is the visualizer that is based on ArcGIS [7] and that projects the 2-D polygons onto world maps. The third is an analysis engine that is designed to identify unexpected or unwanted behaviors in update streams. The analysis engine is designed to be flexible. It can include anomaly detection algorithms that are based on specific behaviors in update streams (*e.g.,* [42, 52]) or on relative characteristics of the generated polygons.

We demonstrate the potential utility of Bigfoot in network security and operations through a series of case studies, which is the third contribution of our work. The studies are focused on both "normal" BGP update activity and on anomalous behaviors that have been described in prior work. Specifically, we show examples of network address space footprints from BGP updates that have a predictable structure and that are useful as a baseline for recognizing anomalous behaviors. We tune the analysis engine in Bigfoot to identify several classes of network anomalies and then apply it to 1.7B BGP updates

[1] Bigfoot is deployed in Internet Atlas[33] and will be openly available to the community at the time of publication along with all the case studies.

978-1-5090-1605-1/16/$31.00 ©2016 IEEE

collected over a period of one year from the RouteViews [23] and BGPmon project [58]. We identify 139 candidate events in this data. We evaluate a subset of these events by hand, along with ground truth information from existing literature, to show how the resulting visualizations effectively highlight unwanted behaviors in the update streams.

2 Visualizing Locations of Network Prefixes

In this section, we describe the basic approach for visualizing the geographic footprint of IP network prefixes. We begin with a naïve approach, and provide an illustrative example that motivates the need for a more robust and consistent geographic representation of prefixes. For the remainder of the paper, we assume IPv4 address prefixes, although there is nothing inherent in our methods that preclude their application to IPv6 address prefixes.

2.1 Naïve Visualization Approach

Our simple approach to generating geopositioned polygons for IP address prefixes that are included in BGP updates takes two inputs: *(1)* a stream of network prefixes, and *(2)* a percentage (p) that specifies the number of IP addresses to select for geolocation within a given prefix. We assume the availability of an IP geolocation service that is highly scalable and that can provide responses to queries in a timely fashion [38]. There are many such commercial geolocation services *e.g.,* from Akamai, Maxmind, and others. We also assume that only a subset of addresses within a prefix are required to convey sufficient information for operational purposes. Optimizing p based on prefix size, performance requirements and resultant polygons is a subject for future work. For analyses presented in this paper, we set p = 100% *i.e.*, we geolocate all IP addresses in a prefix.

Using the specified inputs, the simple approach for generating visualizations is as follows. *(1)* Select the specified percentage p of IP addresses from each prefix (*e.g.,* in a uniform random fashion). *(2)* Geolocate the selected IP addresses. In this study we use Maxmind's IP Geolocation service [12]. *(3)* Project the geographic locations of the selected IP addresses onto a world map. *(4)* Enclose the projected points in a 2-D polygon. What we show below is that simple methods for drawing polygons (step #4) can result in shapes that convey little or no useful information. This is the main motivation for our Bigfoot methodology (described in §3).

Given a set of points on a world map that can act as anchors, there are many possible ways in which polygons that enclose these points could be drawn. We examine three simple methods for drawing lines between consecutive points that form the polygons in the following section: unsorted (random), latitude-sorted and longitude-sorted.

2.2 Illustrative Example

To analyze the naïve approach methods, we used prefixes of different sizes from network prefixes in different parts of the world. As noted above, in each case we use p = 100% so that all IP addresses are geolocated. However, we only project *unique locations* on the map and we only use those unique locations as anchors for drawing polygons.

Figure 1 shows the network polygons for the Fairpoint Communications network (a /16 prefix). Our observations of prominent features in the polygons include *(i)* identical coverage (*e.g.*, north-eastern regions in all three polygons), *(ii)* absence of coverage (*e.g.*, north and north-west regions are missing in latitude-sorted polygon compared to the other two polygons), *(iii)* sharp spikes (*e.g.*, all over latitude-sorted polygon) and multiple overlapping spikes (*e.g.*, unsorted polygon), and *(iv)* dense coverage (*e.g.*, north-east and northern regions of longitude-sorted polygon).

2.3 Discussion

- Using naïve methods one might expect polygons with the number of edges proportional to size of the network prefix due to the number of cities represented in Maxmind [39]. However, our visualizations suggest that the block-level aggregates are much less diverse. We hypothesize that since prefixes are either grouped or divided into super- or sub-prefixes respectively, polygons created using simple methods actually highlight geolocation inaccuracies [47]). This is counter to our objective of providing visualizations that are useful in an operational setting.
- We observe inconsistencies in polygons of all of the prefixes we examined using naïve methods. Omission of one or two points can cause significant changes in shapes. The result is that no intuition is conveyed in these representations making meaningful assessment more difficult and precluding decision support in an operational setting. This calls for context-aware methods to visualize diverse and dynamic network data.

3 The Bigfoot Framework

To create the ability to produce visualizations of network address prefixes in BGP updates we developed *Bigfoot*. The objectives of this system are to produce visualizations that eliminate the inconsistencies and artifacts illustrated in §2 and to provide visual clues to behaviors of interest in BGP updates.

Bigfoot is designed as a modular system in which each component is open and configurable with a minimal set of interfaces. This approach enables the system components to be fine-tuned, extended, or replaced without affecting the rest of the framework. To maximize the utility of the system, Bigfoot can send *alerts* about threat events to analysts based on the network- and IP prefix-specific subscriptions they create.

3.1 Core Components

Bigfoot consists of three components: the *inconsistency solver*, the *anomaly detector* and the *visualizer*, and we describe each part of this modular system below.

Inconsistency Solver. Bigfoot begins by ingesting BGP data in batch or streaming mode and generates geocoded 2-D polygons.[2] First, we start by extracting the announced address prefix and AS path information from the data and create an in-memory hash of the updates. This step is executed in parallel on a 32-core machine. Next, we generate IP addresses contained in every prefix using p=100% and simultaneously geolocate them using MaxMind geolocation service.

The inconsistency solver borrows graphical techniques including alpha shape creation [1] and convex hull-based boundary creation [6] to produce consistent 2-D representations of network prefixes from the geolocated addresses. Both of these options can be invoked from the visualizer depending on the requirement. The inconsistency solver acts as a post-processing step to the basic naïve approach described in the prior section, and takes two inputs including a *(1)* list of IP-geolocated coordinates/points for every prefix and *(2)* a threshold type for outlier elimination to produce shape files containing individual polygons for geolocated coordinates in an input prefix list.

Specifically, given a set of (4 or more) unique IP-geolocated coordinates for a prefix in the prefix list, the convex hull estimation method produces a polygon that encloses the given points. For a network prefix with only one available coordinate, a circle with that point on the center is generated with a radius relative to the size of other polygons, and for a prefix with two points, two circles are created. A network prefix with three points is simply represented by a triangle, which forms the base case for our polygon construction. Similarly, the alpha shape creation method takes a set of (4 or more) geographic coordinates and creates Delaunay triangles [31] to establish connections between nearby points. In the process, any outlier points classified based on their inter-vincenty distance [27] exceeding a given threshold are removed, where the threshold is computed automatically using the input threshold type.[3] Our design choice of

[2] All results in this paper are based on batch mode operation but capabilities are in place to operate in streaming mode.

[3] Bigfoot currently supports three threshold types: mean, median and mode. In this paper, mean of the all inter-vincenty distances is calculated and is used to winnow points whose neighbor-vincenty distances exceed the

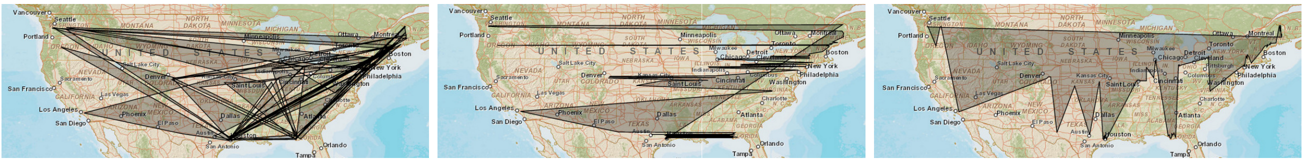

Figure 1: **Network footprint of Fairpoint Communications (AS 32645, 216.227.0.0/16) produced using unsorted (left), latitude-sorted (center), and longitude-sorted (right) geolocation information for p=100%.**

automatically calculating threshold value to winnow points based on input threshold type, and not based on directly passing a threshold value, is based on the fact that multiple IP addresses geolocate to a single geographic location and carelessly chosen values increase the chances of removing such important cliques in the polygon.[4]

Anomaly Detector. In addition to producing visually consistent polygons of network prefixes in an update stream, Bigfoot includes an anomaly detection component. The objective of this component is to enable behaviors of interest to be highlighted by the system. These include classes of anomalies that have been considered in other studies as well as those that relate to relative shapes and positions of network polygons.

Algorithm 1: Bigfoot Anomaly Detector

```
   input : timeOfInterest
   input : k
   input : UpdateStream
1  foreach update U in UpdateStream[prev(k)] do
2      aggregates = combinePrefixes(prev(k));
3      basePolygons[U] = generate2DPolygon(aggregates);
4      baseASPaths[U] = getASPaths(U);
5  foreach update U in UpdateStream[timeOfInterest] do
6      f = compareASPaths(U, baseASPaths[U]);
7      if f == “match" then
8          newPolygon = generate2DPolygon(U);
9          i = comparePolgons(newPolygon, basePolygons[U]);
10         if i == “match" then
11             normalUpdate(U);
12         else
13             newASPath = augmentASNumbers(U);
14             j = compareASPaths(newASPath,
               baseASPaths[U]);
15             if j == “match" then
16                 normalUpdate(U);
17             else
18                 anomalyUpdate(U);
19     else
20         asPath = findClosestASPath(U, baseASPaths[U]);
21         $P_{hijacked}, P_{hijacking}$ = getAttackInfo(asPath, U);
```

Consistent with standard approaches to anomaly detection, this component establishes a baseline for “normal" behavior and uses thresholds to identify anomalous behavior. Algorithm 1 shows the key steps of our method that considers both AS path and network polygon shape characteristics, and it works as follows. To establish the baseline for update behavior, the anomaly detector uses updates from previous k days and aggregates the subnet information announced in a particular AS path together to generate consistent 2-D polygons (steps 1 to 4). In our evaluation, to derive k, we use a simple linear model on BGP update churn per day to obtain a value of 4. These polygons are the *base case* polygons for detecting behaviors of interest in the BGP update stream. We show that this simple approach is effective in our case studies but plan to consider alternative approaches for future work.

Next, a window of time (*e.g.*, hours, days, weeks) over which updates will be analyzed is selected (steps 5 to 21). This window of interest can range from a few hours to many months.[5] The anomaly detector compares the AS path included in each update with the base case AS path (step 6). If they match, the subnet polygons in the update are compared with the base case polygons by using Convex Hull intersection comparison (specifically using *equals* and/or *contains* comparison methods described in [34, 51]) which are available in Python's shapely library [15] (step 9). In a similar vein, the number, area and geography of the base case polygons are also compared with the polygons produced from the updates. In case of previously observed AS path, the threshold criteria for anomaly detection is currently set as a perfect mismatch from base case polygons in all the comparisons. The criteria is set to perfect mismatch in order to detect changes in geographical footprint of well known AS paths, which in turn is used by the anomaly detector to find redirection attacks and router misconfigurations. Based on this criteria, if there is perfect match in all the comparisons, the anomaly detector classifies events as normal updates (steps 10 and 11). When a mismatch is determined using the intersection comparison methods, the anomaly detector, for each update, augments AS numbers with subnet information by looking up each of the AS numbers in Prefix2AS dataset [21] and checks if the subnet in the update is associated with any one of those AS numbers present in the AS path of the base case (steps 13 and 14). If it is, then the update is classified as a normal update. In our design, lowering the threshold criteria from perfect mismatch, to say 95% mismatch, enhances Bigfoot's ability to detect events at finer granularities (*e.g.*, to individual subnets.)

If there is a mismatch in the initial AS path comparison, the anomaly detector compares the subnet information to the base data to determine the previously observed AS paths. Among the previously observed AS paths, it determines the closest matching AS path(s) and performs polygon equals and contains comparison with the base polygon to identify the misbehaving AS number(s) and the victim subnets (steps 19 to 21). In case of previously unobserved AS path, the threshold criteria for anomaly detection is currently set to perfect match with base case polygons, which enables detection of potential victims of man-in-the-middle attacks. The intuition behind having a different threshold criteria for AS path mismatch case is that, for subnets with well established AS paths, only when there is a routing leak, the AS paths in the updates change. By doing this, it produces two different types of information in the anomalous updates: *(i)* networks for which the traffic is being hijacked and/or leaked, and *(ii)* networks involved in the hijacking/leaking (step 21).

Visualizer. All visualizations shown in this paper are generated from the third component of Bigfoot called the Visualizer. Visualizer was developed as an extension of Internet Atlas [33], which is built on top of ArcGIS. The GIS foundation enables polygons to be displayed on maps of various types. Input to the visualizer are the shape files (including .shp, .shx, .dfb, .cpg) and the associated meta data generated by the inconsistency solver. Output from the visualizer is available through the Internet Atlas web-based UI.

The visualizer includes usability features that enable it to operate

calculated mean value.

[4] There are actually multiple points in the extreme north of Canada in Figure 2-(left).

[5] We show the application of different time scales in §4.1 and §4.2.

on update streams from multiple sources and to navigate through updates (both spatially and temporally) and corresponding maps in various ways. For example, we include the ability to highlight sender/receiver characteristics in updates, which can be helpful in identifying certain types of behaviors as illustrated in §4. This is enabled in part by the ArcGIS tracking analyst [2] available as part of Internet Atlas, which dynamically generates the corresponding shape files in a computationally efficient fashion. While we emphasize basic visualization techniques in this paper, the visualizer can produce a variety of other representations (*e.g.*, heatmaps).

Summary. The main features of Bigfoot include: *(1)* a method for ingesting and grooming BGP update data that produces geopositioned polygons for network prefixes, *(2)* a filtering capability for identifying anomalies in the update data, and *(3)* a visualization component that can produce outputs in several standard formats (*e.g.*, .shp, .png). The system is modular and configurable in the sense that each component can be *tuned*, *modified* and/or *replaced* independently. The modular design offers an opportunity to enhance situational awareness and configure for specific use cases as needed with minimal modifications to the system.

3.2 AS Visualizations

In this section, we demonstrate Bigfoot's ability to produce consistent 2-D polygons for various networks, ranging from large tier-1 to regional networks with varying prefix sizes, by visualizing their prefixes chosen from the CAIDA's AS ranking project [4]. Figure 2-(left) shows the network footprint of XO Communications, a tier-1 network, consisting of 7M addresses. Coverage of XO is large and a simple consistency check shows that the polygons produced are consistent with their network map [29]. 2-D polygons produced using Bigfoot for Fairpoint Communications, containing 83K addresses, are shown in Figure 2-(center). The polygons produced are consistent when compared with the naïve approach (see Figure 1). Figure 2-(right) shows the network updates from Syringa, a regional service provider in Boise, ID, containing 47K addresses. These examples illustrate Bigfoot's ability to produce geographically consistent 2-D visualizations of diverse networks with varying prefix sizes and geographic affinity.

4 Demonstrating Bigfoot

In this section, we demonstrate how Bigfoot can be used to provide insights on routing errors that can lead to transient outages, and eventually to inconsistent network state. Given our geographical approach for prefix visualization, Bigfoot is particularly effective at highlighting behaviors that involve sudden changes in geographic footprints. We developed two approaches to test and evaluate Bigfoot: *(i)* based on isolated network events documented in prior work [5, 26], and *(ii)* using archives of BGP updates from RouteViews [23] and BGPmon [58] and validating with ground truth information [22, 25].We discuss results from processing the BGP update data and classifying those updates that do not have ground truth for verification. Finally, we compare against RIPEStat [18] and VIS-SENSE [30], to demonstrate the utility of Bigfoot.

4.1 Isolated Network Event Analysis

We begin assessment of Bigfoot by extracting BGP updates from archived data that includes two documented events of interest [5, 26]. These are presented as use cases 1 and 2 (below). We then provide this data as input to Bigfoot and show the how visual representations effectively highlight the target behavior.

Use case 1. In this scenario, we consider BGP updates from a single day (5th of August 2014) that contained a routing leak based on peering agreement between Vimpelcom (AS 3216) and China Telecom (AS 4134) as described in [5]. BGP update data from the target prefixes is extracted from archives and used as input to Bigfoot.

Figure 3-(left) shows the base case polygons including the subnets and AS information belonging to Vimpelcom. Both the subnets and AS path information of the base case data are available at [33]. Figure 3-(right) illustrates the impact of routing leak by showing victim subnets (Vimplecom) in red polygons and the misbehaving subnets (China Telecom) in black polygons. These subnets are detected using the polygon comparison method described in §3. On the day of the routing leak, the AS paths associated with Vimplecom subnets did not match with any of the AS paths in the base case data. By setting the criteria for anomaly as *perfect matches with base cases* using polygon comparison methods, the previously observed AS path information is extracted from the base data. From the closest matching AS path, the misbehaving AS is identified as AS 4134 and the prefixes associated with AS 4134 are obtained from Prefix2AS dataset [21] and are automatically generated and displayed as black polygons.

Use case 2. The ability to monitor updates from all peers is important in network operations. Problems related to performance and security that can arise from unmonitored peers are described in [26]. In this example, we focus on update data from May 22, 2014 that represents misbehaving peer events and use Bigfoot to visualize the impact.

Figure 4-(right) shows the impact of Beltelecom (AS 6697) creating a detour of traffic from Yandex (AS 13238). These detours result in degraded performance as described in [26]. This was due to the effect of bogus routes announced by Beltelecom (black polygons) on its peer, *i.e.,* Yandex (red polygons). Bigfoot effectively highlights these bogus route events by providing the required filtering and monitoring capability, and the ability to compare this against normal/expected behavior (Figure 4-(left)).

4.2 Examination of Update Streams

Next, we demonstrate Bigfoot's capabilities in operational deployments by considering two archives of update data from RouteViews [23] and BGPmon [58] project for a period of six months each, from February 2013 to July 2013 (D1) and January 2015 to June 2015 (D2). These archives included 699,091,503 and 1,091,009,199 BGP updates respectively. In this data corpus, we found 73 candidate events from D1 and 66 candidate events from D2 by applying our anomaly detector. From the first 73 candidate events, we select two events[6] (as shown in use cases 3 and 4 below) and validate with ground truth information [22] to show how anomalous prefixes in update streams can be highlighted. Similarly for the remaining 66 candidate events from D2, we validate one event with ground truth information [25]as shown in use cases 5.

Use case 3. BGP prefix hijacking is a well known threat in interdomain routing and can occur at varying time scales, typically lasting from few minutes to several hours. In this scenario, we describe a prefix hijacking event that we found using Bigfoot.

Figure 5-(left) shows the footprint for base case updates in Guadalajara, MX to Washington, DC. During the hijack event, instead of the standard AS path, a new AS path emerges that is directed to the Belarusian ISP GlobalOneBel (black polygons). This causes traffic to flow to Russia via Europe and then back into the US. The hijacked prefixes are clearly illustrated in Figure 5-(right) We also validated this hijacking event using the AS numbers and subnet information given in [22]. This particular hijack happened throughout the month of February 2013; for the purpose of succinct representation, we show only the anomalies detected for three days of archived BGP data. This demonstrates how Bigfoot can be employed to provide visual cues on prefix hijacking, independent of the time scale at which they occur.

Use case 4. Bigfoot detected another prefix hijacking event described in [22] where traffic was being rerouted out of the US to Iceland by Icelandic ISP Opin Kerfi (AS 48685). Figure 6-(left) depicts the regular flow of updates where the announcements stay within US. The divergent paths taken by updates after the leak are

[6] Remaining 71 events warrant careful investigation with similar ground truth information. We are working on a feature that enables an operator to flag events as false positives.

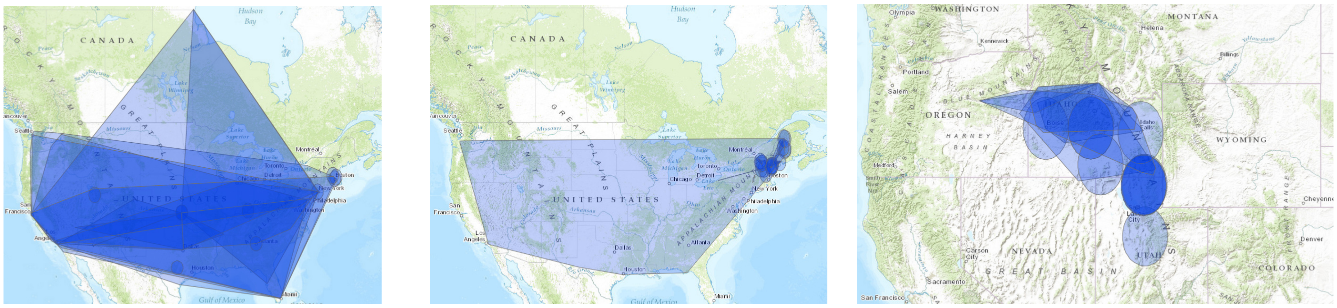

Figure 2: **(Left) - Polygons produced by Bigfoot for AS 2828 and AS 7014 assigned to XO Communications. (Center) - Network footprint of AS 32645 assigned to Fairpoint Communications. (Right) - Network footprint of AS 15305 assigned to Syringa Networks.**

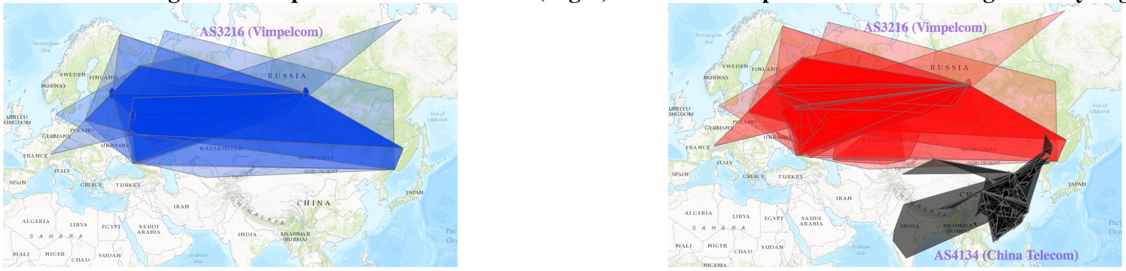

Figure 3: **Footprint of BGP update events produced before (left) and during (right) routing errors scenario described in [5].**

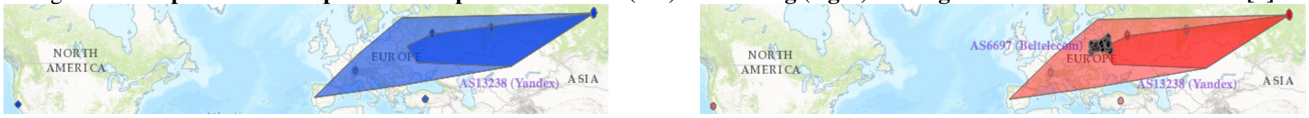

Figure 4: **Footprint of BGP update events produced before (left) and during (right) misbehaving peering scenario described in [26].**

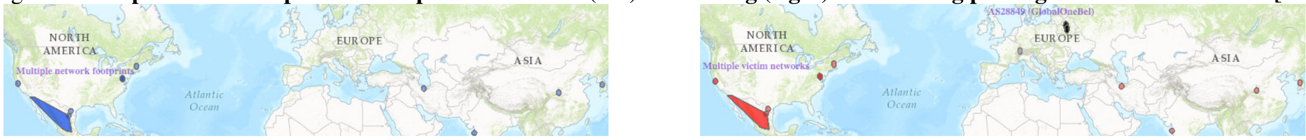

Figure 5: **Footprint of BGP update events produced before (left) and during (right) prefix hijacking (Belarus) described in [22].**

shown in Figure 6-(right) and the misbehaving AS is identified as Opin Kerfi (AS 6677). Bigfoot's ability to provide a visualization of the difference between normal and hijacked routes should aid network operators in identification and remediation of these kinds of events.

Use case 5. Bigfoot also highlights a BGP routing hijack event across different continents, which is overlooked by other methods. Per the routing event explained in [25], traffic from the UK-based customers of British Telecom, including Atomic Weapons Establishment, alone are hijacked to Ukraine. Surprisingly, using Bigfoot, we found that the impact of hijack event to be *more* serious than what has been reported. Specifically, apart from the UK-based customers reported in [25], we found that the traffic of customers from other continents, including USA and India, were also hijacked during the event as shown in Figure 7. In addition to validating the hijack explained in [25] as shown in Figures 7-(top left) and -(top right), Figures 7 depicts Bigfoot's ability to identify the footprint of additional prefixes from other continents including countries like USA (bottom left) and India (bottom right) that were also hijacked.

These use cases highlight how network operators can use Bigfoot to identify the geographical region(s) affected by normal and anomalous event updates. Bigfoot visualizations can also convey additional metadata such as the AS path observed under normal operations, the AS path observed during the anomaly, the timestamps associated with the updates along with prefix details.

4.3 Validation of the Remaining Events

Validation of *all* candidate events identified in D1 and D2 is problematic because of the general lack of ground truth. To cope with the lack of ground truth information, we adopt a classification-based approach and manually examine *(1)* the meta-information including AS path, prefix allocation, and duration of the attacks, and *(2)* geographic characteristics of the remaining candidates in D1 and D2. The idea is to compare the characteristics of candidate events to events for which we have ground truth to determine their consistency with our definition of anomalies. We plan to reach out to the network operators to further validate these events in future work since we cannot say with certainty that they are anomalies that required action.

C1: Classification using meta-information. We start with examining various meta-information of the prefixes and classify them into three categories: *(i)* $C1_A$, where the hijacking AS inserts itself or replaces one or more of the ASes in the AS path; *(ii)* $C1_B$, where the prefixes announced in the update forms an entirely new AS path; and *(iii)* $C1_C$, where the prefixes announced in the update belongs to a different address registry.

Table 1: **Classification of remaining candidate events using meta-information.**

$C1_A$		$C1_B$		$C1_C$	
D1	D2	D1	D2	D1	D2
25	23	44	41	9	7
U1, U3		All except U2		-	

Table 1 shows the classification of the remaining candidate events in D1 and D2 based on meta-information of the prefixes. On manual examination, we found that 25 candidates from D1 and 23 candidates from D2 either inserted itself or replaced the AS paths, similar to use cases 1 and 3 (*i.e.*, U1, U3). Similarly, 44 event from D1 and 41 events from D2 exhibited properties similar to all other use cases, except U2, where a new path is seen in the updates. Finally, 16

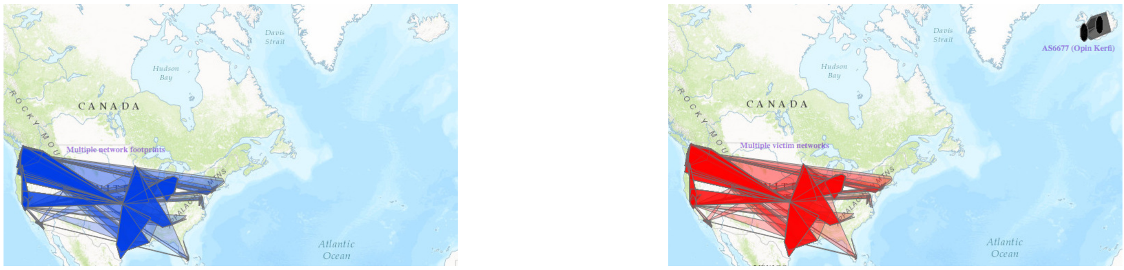

Figure 6: **Footprint of BGP update events produced before (left) and during (right) prefix hijacking (Iceland) described in [22].**

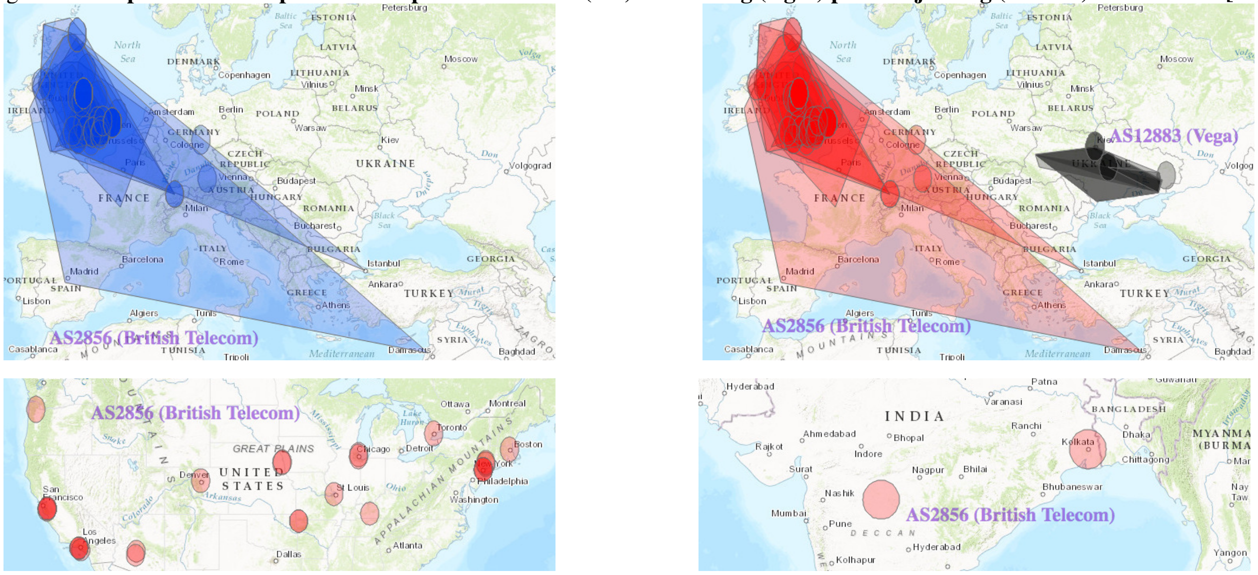

Figure 7: **Footprint of BGP update events produced before (top left) and during (top right) prefix hijacking (Ukraine) described in [25]. Customers from other continents including countries like India (bottom right) and USA (bottom left) were also attacked.**

events from D1 and D2 did not exhibit characteristics as those for which we had ground truth. We are currently investigating these events in further detail.

Table 2: **Sub-classification of candidate events using meta-information and time duration of the events.**

	$Sub-C1_A$		$Sub-C1_B$		$Sub-C1_C$		
	D1	D2	D1	D2	D1	D2	
Short	10	9	9	17	2	0	U5
Moderate	13	10	16	11	1	1	U1, U2
Long	2	4	19	13	6	6	U3, U4

Next, we further classify candidates that are shown in Table 1, based on time duration of the events, into *(a)* short, where the events lasted for less than 12 hours; *(b)* moderate, where the duration of the events were greater than 12 hours but lesser than 12 days; and *(c)* long, where the events existed for more than 12 days. We show the sub-classification of C1 (*i.e.*, $Sub-C1$) for all three categories (explained above) in Table 2. From the table, we find that the classification of candidates using time duration and meta-information is quite effective for characterizing events for which we do not have ground truth. For example, using U3 and U4 as seeds, we manually examined the aforementioned properties of events in D1 and D2 for all three categories and found that 6 events lasted for more than 12 days and the hijacking ASes replaced itself in the update AS path. Similarly, for 26 events, the time duration was under 12 hours and the announced AS paths were completely new. These 26 events were similar to use case 5.

C2: Classification using geography. Finally, we analyze the geographical characteristics of the events and classify them based on prefixes that are *(1)* distributed across different continents ($C2_A$); *(2)* spread across different countries ($C2_B$) but within the same continent; and *(3)* regional, $C2_C$, where the prefixes get geolocated to different regions in the same country.

Table 3: **Classification of remaining candidate events using geography of the prefixes.**

$C2_A$		$C2_B$		$C2_C$	
D1	D2	D1	D2	D1	D2
36	26	31	37	6	3
U1, U3, U4		U1, U5		-	

Table 3 shows the classification of the remaining candidate events in D1 and D2 based on geography of the prefixes. We found that prefixes of 36 candidates from D1 and 26 candidates from D2, similar to use cases 1, 3 and 4 (*i.e.*, U1, U3 and U4) were geographically distributed across different continents. Similarly, 31 event from D1 and 37 events from D2 exhibited properties similar to other use cases (*i.e.*, U2 and U5), where the prefixes were geolocated within the same continent but across different countries. Lastly, 9 events from D1 and D2 did not exhibit characteristics as those for which we had ground truth, which warrants further investigation.

4.4 Comparison with other tools

To provide further perspective on Bigfoot, we compare it to other tools for BGP update visualization and threat detection including RIPEstat [18] and VIS-SENSE [28]. We do not report comparisons to BGPlay [32], HERMES [10] and VAST [44] since those tools *(i)* are *only* used for visualizing the *AS interconnections and paths* extracted from BGP update streams and *(ii)* do not have anomaly detection capabilities. Further, none of the existing tools have the capabilities to visualize network prefix geographic footprints.

RIPEstat. RIPEstat [18] is a widely used web-based interface for prefix and AS visualizations. For this comparison, we focus on RIPEstat's *Observed Network Activity* widget [19], a heat map based visualization tool. To create heat map based visualization functionality on Bigfoot, we plugged in a new heat map rendering engine comprising of 106 lines of JavaScript code—created using heatmap.js [9]—into the Visualizer. Figure 8 compares the footprint of one of the prefixes of Fairpoint Communications (*i.e.*, `216.227.0.0/16`) produced using RIPEStat (left) and Bigfoot (right). We note that, apart from identifying locations that are identified by RIPEStat (left), Bigfoot identified additional footprint (right) of Fairpoint Communications as seen in [8] highlighting Bigfoot's ability to effectively and exhaustively identify a network's footprint. Bigfoot also provides support to visualize multiple network prefixes simultaneously, unlike the RIPEstat widget.

VIS-SENSE. The VIS-SENSE [28] project has produced visualization technologies for identification and prediction of abnormal behavior. Specific to BGP monitoring and hijack detection, VIS-SENSE has capability similar to VisTracer[37], BGPfuse[46], BGPViewer[45]. Since, no public version of VIS-SENSE tools are available, we focus on the attack scenario explained in Biersack *et al.* [30] for this comparison. In that scenario, VIS-SENSE provides visual analysis of a prefix hijack event where a spam attack was orchestrated by the attacker and false ownership claim to the victim network—Link Telecom (AS31733)—was achieved via Internap (AS12182). Figure 9 shows the detection and visualization of the scenario using Bigfoot, where apart from reproducing the results shown in Figure 4 of [30], Bigfoot also highlights the scope of attack events by providing geographical footprints of the victim and the attacker networks. We also note that VIS-SENSE's anomaly detection requires inputs from both active measurements via a traceroute-based tool called *Spamtracer* and a model that represents the normal and expected behavior of the network. Unlike VIS-SENSE, Bigfoot does not require any traceroute-based inputs to detect network threat events.

5 Related Work

Visualizations can be useful for analyzing large, complex network data sets. A common technique for visualizing the spatial relations between IP address blocks is a Hilbert representation [11, 13]. Other interesting approaches for unraveling BGP's complexity are to plot AS relationships in a radial space [3], or to plot IP prefixes in a quadrant-based 2D space [54]. BGPlay [32] creates graphical representation of AS paths seen in BGP updates. Similar to BGPlay are HERMES [10] and VAST [44] which are used for exploring and visualizing the ASes and their interconnections. Elisha [53] and Event Shrub [55] detect origin AS change events and anomalies based on historical data respectively. Commercial tools (*e.g.,* [24]) use similar techniques to understand the anomalies and attacks. Lad *et al.* creates a ranking scheme for visualizing AS paths called Link-Rank [43] and design PHAS [41] alerting system that builds upon Link-Rank for alerting prefix hijacks. TAMP [56] and PGPeep [50] are employed to detect router misconfigurations, flaps and other routing anomalies.

A community effort that bears the strongest resemblance to ours is the RIPEStat [18], which offers representations of IP address space as heat maps. Another effort that is similar to ours is `VIS-SENSE`, a visual analytics project for detecting prefix hijack attacks [30]. Bigfoot goes well beyond RIPEStat and VIS-SENSE by including: *(1)* consistent geographical representations, *(2)* addresses visual outliers, and *(3)* offers viable use cases. To the best of our knowledge, Bigfoot is the first effort to establish a Geo-based visualization of network footprints that includes capabilities to layer sizable network updates on top of a map of the physical Internet [33].

6 Summary and Future Work

The reachability information exchanged in BGP updates is the basis for establishing routing between networks and as such has direct implications on a wide range of characteristics including security, performance and robustness. Prior empirical studies on BGP update behavior highlight the large volume and diverse characteristics as well as a range of risks and threats. Visualizations offer an opportunity to unravel complexity and to bring attention to key events of interest in large data sets such as BGP update streams.

In this paper we describe a new approach for visualizing BGP updates that we call Bigfoot, which is based on generating 2-D polygons of the geographic footprint of network prefixes included in updates. We show how simple approaches to generating footprints result in polygons that are inconsistent and thus not applicable in target use cases. In contrast, Bigfoot generates polygons based on convex hulls and using context sensitivity that enable the generation of consistent representations. We demonstrate Bigfoot's general capability and how it can be used in specific use cases described in prior work.

In ongoing work, we are focused on expanding and enhancing Bigfoot's applicability to a broader range of security (*e.g.*, remotely triggered black-holing for DoS attack mitigation [16, 17]) and operational (*e.g.*, unauthorized traffic redirection due to misconfiguration [14]) use cases. In particular, we are focused on enabling Bigfoot to produce visualizations that highlight a broader range of anomalous and attack conditions and to potentially play a role in mitigation of these situations. We also plan to consider ways in which network footprint visualizations can be used in network planning and risk assessment. Future efforts will also include the enhancement of Bigfoot's anomaly detector to conduct a range of analyses on false alarm rates for different configurations of thresholds.

Acknowledgements

We thank RouteViews and BGPmon projects for providing BGP datasets. This material is based upon work supported by the DHS grant BAA 11-01 and AFRL grant FA8750-12-2-0328. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not reflect the views of the DHS or AFRL.

References

[1] Alpha Shapes. http://en.wikipedia.org/wiki/Alpha_shape.

[2] ArcGIS Tracking Analyst. http://www.esri.com/software/arcgis/extensions/trackinganalyst.

[3] CAIDA: IPv4 Address Space Utilization. http://www.caida.org/research/topology/.

[4] CAIDA's ranking of Autonomous Systems. http://as-rank.caida.org/?mode0=as-ranking.

[5] Chinese Routing Errors Redirect Russian Traffic. http://research.dyn.com/2014/11/chinese-routing-errors-redirect-russian-traffic/.

[6] Convex Hull-based boundary creation. http://blog.thehumangeo.com/2014/05/12/drawing-boundaries-in-python/.

[7] ESRI ArcGIS. http://www.esri.com/software/arcgis/.

[8] Fairpoint Communications coverage map. http://www.fairpoint.com/cmsimages/Service-Area-Map_tcm12-18519.png.

[9] heatmap.js - Dynamic Heatmaps for the web. http://www.patrick-wied.at/static/heatmapjs/.

[10] HERMES: Visualization of ISP Relationships. http://www.dia.uniroma3.it/hermes/.

[11] Hilbert Curves. http://datadrivensecurity.info/blog/posts/2015/Jan/mapping-ipv4-address-in-hilbert-space/.

[12] Maxmind: IP Geolocation Database. https://www.maxmind.com.

[13] Measuring the use of IPv4 space with Heatmaps. http://www.caida.org/research/traffic-analysis/arin-heatmaps/.

[14] On-going BGP Hijack Targets Palestinian ISP. http://research.dyn.com/2015/01/going-bgp-attack-targets-palestinian-isp/.

[15] Python Shapely library. https://pypi.python.org/pypi/Shapely.

[16] RFC 1997. https://tools.ietf.org/pdf/rfc1997.pdf.

[17] RFC 5635. https://tools.ietf.org/pdf/rfc5635.pdf.

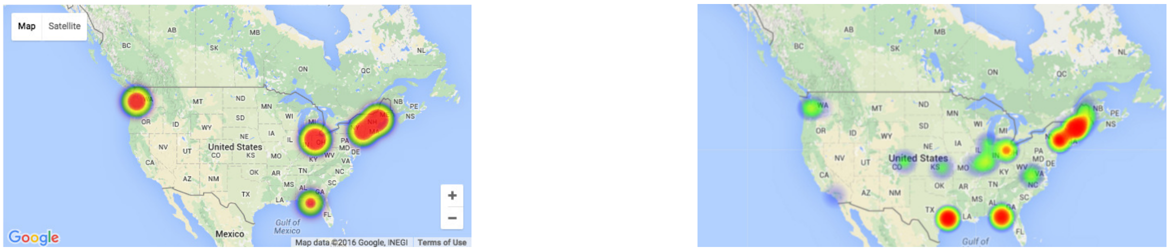

Figure 8: **Comparison of address prefix footprint for `216.227.0.0/16` used by Fairpoint Communications. Visualizations are created using RIPEStat (left) [20] and Bigfoot (right).**

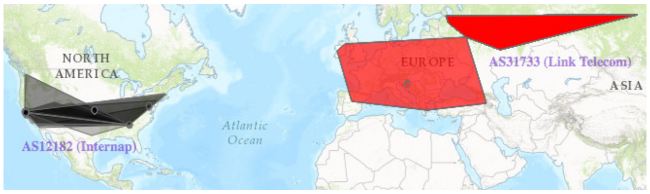

Figure 9: **Footprint of BGP update events produced during (bottom) Link Telecom hijack scenario described in [30].**

[18] RIPEstat Network Activity widget. `https://stat.ripe.net/widget/network-activity`.

[19] RIPEstat scenario. `https://stat.ripe.net/widget/network-activity#w.resource=193.0.0.0%2F10`.

[20] RIPEStat Visualization for `216.227.0.0/16`. `https://stat.ripe.net/widget/network-activity#w.resource=216.227.0.0/16`.

[21] The CAIDA UCSD Prefix to AS dataset. `http://www.caida.org/data/routing/routeviews-prefix2as.xml`.

[22] The New Threat: Targeted Internet Traffic Misdirection. `http://research.dyn.com/2013/11/mitm-internet-hijacking/`.

[23] The Route Views Project. `http://www.routeviews.org/`.

[24] Thousand Eyes Vizualizing and Troubleshooting Tool. `https://www.thousandeyes.com/lp/bgp-webinar?utm_source=blog&utm_medium=cta&utm_campaign=webinar`.

[25] UK traffic diverted through Ukraine. `http://research.dyn.com/2015/03/uk-traffic-diverted-ukraine/`.

[26] Use Protection if Peering Promiscuously. `http://research.dyn.com/2014/11/use-protection-if-peering-promiscuously/`.

[27] Vincenty Distance. `http://en.wikipedia.org/wiki/Vincenty's_formulae`.

[28] VIS-SENSE. `http://www.vis-sense.eu/`.

[29] XO Communication Assets. `http://www.xo.com/why/the-right-network/assets/`.

[30] E. Biersack, Q. Jacquemart, F. Fischer, J. Fuchs, O. Thonnard, G. Theodoridis, D. Tzovaras, and P. Vervier. Visual Analytics for BGP Monitoring and Prefix Hijacking Identification. *IEEE Network*, 2012.

[31] B. Delaunay. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934.

[32] G. Di Battista, F. Mariani, M. Patrignani, and M. Pizzonia. Bgplay: A system for visualizing the interdomain routing evolution. In *International Symposium on Graph Drawing*, 2003.

[33] R. Durairajan, S. Ghosh, X. Tang, P. Barford, and B. Eriksson. Internet Atlas: A Geographic Database of the Internet. In *ACM HotPlanet*, 2013.

[34] M. J. Egenhofer and J. R. Herring. Categorizing binary topological relations between regions, lines, and points in geographic databases.

[35] B. Eriksson, P. Barford, B. Maggs, and R. Nowak. Posit: A Lightweight Approach for IP Geolocation. *ACM SIGMETRICS*, 2012.

[36] N. Feamster, M.Mao, and J. Rexford. BorderGuard: Detecting Cold Potatoes from Peers. In *ACM IMC*, 2004.

[37] F. Fischer, J. Fuchs, P.-A. Vervier, F. Mansmann, and O. Thonnard. VisTracer: A Visual Analytics Tool to Investigate Routing Anomalies in Traceroutes. In *ACM VizSec*, 2012.

[38] Z. Hu, J. Heidemann, and Y. Pradkin. Towards geolocation of millions of ip addresses. In *ACM IMC*, 2012.

[39] B. Huffaker, M. Fomenkov, and K. C. Claffy. Geocompare: A Comparison of Public and Commercial Geolocation Databases. In *CAIDA Tech Report*, 2011.

[40] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. *IEEE/ACM TON*, 2001.

[41] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang. PHAS: A Prefix Hijack Alert System. In *Usenix Security*, 2006.

[42] M. Lad, D. Massey, and L. Zhang. Visualizing Internet Routing Changes. *IEEE TVCG*, 2006.

[43] M. Lad, L. Zhang, and D. Massey. Link-rank: A Graphical Tool for Capturing BGP Routing Dynamics. In *IEEE NOMS*, 2004.

[44] J. Oberheide, M. Karir, and D. Blazakis. VAST: Visualizing Autonomous System Topology. In *IEEE VizSec*, 2006.

[45] S. Papadopoulos, K. Moustakas, and D. Tzovaras. BGPViewer: Using Graph representations to explore BGP routing changes. In *IEEE DSP*, 2013.

[46] S. Papadopoulos, G. Theodoridis, and D. Tzovaras. Bgpfuse: using visual feature fusion for the detection and attribution of bgp anomalies. In *ACM VizSec*, 2013.

[47] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye. IP Geolocation Databases: Unreliable? *ACM SIGCOMM CCR*, 2011.

[48] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang. BGP Routing Stability of Popular Destinations. In *ACM IMW*, 2002.

[49] M. Roughan, T. Griffin, M. Mao, A. Greenberg, and B. Freeman. Combining Routing and Traffic Data for Detection of IP Forwarding Anomalies. In *ACM SIGCOMM NTW*, 2004.

[50] J. Shearer, K.-L. Ma, and T. Kohlenberg. BGPeep: An IP-space Centered View for Internet Routing Data. In *IEEE VizSec*. 2008.

[51] C. Strobl. Dimensionally extended nine-intersection model (de-9im). In *Encyclopedia of GIS*, pages 240–245. Springer, 2008.

[52] S. Teoh, S.Ranjan, A. Nucci, and C. Chuah. BGP Eye: A New Visualization Tool for Real-time Detection and Analysis of BGP Anomalies. In *IEEE VizSec*, 2006.

[53] S.-T. Teoh, K.-L. Ma, S. F. Wu, D. Massey, X.-L. Zhao, D. Pei, L. Wang, L. Zhang, and R. Bush. Visual-based Anomaly Detection for BGP Origin AS change (OASC) Events. In *Self-Managing Distributed Systems*. 2003.

[54] S. T. Teoh, K.-L. Ma, S. F. Wu, and X. Zhao. A visual technique for internet anomaly detection. In *IASTED International Conference on Computer Graphics and Imaging*, 2002.

[55] S. T. Teoh, K. Zhang, S.-M. Tseng, K.-L. Ma, and S. F. Wu. Combining Visual and Automated Data Mining for Near-real-time Anomaly Detection and Analysis in BGP. In *ACM CCS*, 2004.

[56] T. Wong, V. Jacobson, and C. Alaettinoglu. Internet Routing Anomaly Detection and Visualization. In *IEEE DSN*, 2006.

[57] J. Wu, M. Mao, J. Rexford, and J. Wang. Finding a Needle in a Haystack: Pinpointing Significant BGP Routing Changes in an IP Network. In *USENIX NSDI*, 2005.

[58] H. Yan, R. Oliveira, K. Burnett, D. Matthews, L. Zhang, and D. Massey. BGPmon: A Real-time, Scalable, Extensible Monitoring System. In *IEEE CATCH*, 2009.