Geofeeds: Revolutionizing IP Geolocation or Illusionary Promises?

IOANA LIVADARIU*, Simula Metropolitan, Norway
KEVIN VERMEULEN*, LAAS-CNRS, France
MAXIME MOUCHET, IPinfo, United States
VASILIS GIOTSAS, Cloudflare, United States

Despite more than twenty years of efforts, the research community is still looking for a publicly available Internet-scale IP geolocation dataset with an explainable methodology. Recently, a new hope has appeared, with the emergence of geofeeds. Geofeeds are a self published IP geolocation dataset where operators give the geolocation of their IP addresses, with the underlying idea being that other network providers can tune their services to better serve the IP address depending on its geolocation. In this paper, we analyze whether the hope of finally obtaining the golden geolocation dataset is a real possibility or a mirage. Two years after the standardization of geofeeds, we look at how they are adopted by operators, and what is their accuracy, and how we can use them for operational and research scenarios. First, geofeeds are in the early adoption process with 1.50% and 0.70% of the allocated IPv4 and IPv6 prefixes covering the geofeed prefixes. Second, even if we cannot use geofeeds as ground truth as we found 0.9%, 4.0%, and 8.5% of the client, router, and server IP addresses with an erroneous geofeed, most of them look correct and provide at least a geolocation hint for building an internet scale IP geolocation dataset. Finally, we provide some recommendations on how to use geofeeds and how we could improve the format and the process of sharing geofeeds to improve their quality.

CCS Concepts: • **Networks** → **Network measurement**.

Additional Key Words and Phrases: Geolocation, Geofeeds, Internet measurement

ACM Reference Format:
Ioana Livadariu, Kevin Vermeulen, Maxime Mouchet, and Vasilis Giotsas. 2024. Geofeeds: Revolutionizing IP Geolocation or Illusionary Promises?. *Proc. ACM Netw.* 2, CoNEXT3, Article 15 (September 2024), 21 pages. https://doi.org/10.1145/3676869

1 Introduction

Obtaining an Internet-scale accurate IP geolocation dataset has been a longstanding goal of the research and industry communities. For instance, Content Delivery Networks (CDNs) can leverage the IP geolocation of their clients to redirect them to their closest Point of Presence (PoP) and thus improve their performance. Other examples for the industry include location-specific advertisement, fraud prevention, or access to media only where licensing permits [10, 48]. For research, IP geolocation has been used to measure broadband performance [20], map of Internet infrastructure [16, 50], understand inter-domain routing policies [14], and analyse security practices [64].

*Equal contribution.

Authors' Contact Information: Ioana Livadariu, Simula Metropolitan, Norway, ioana@simula.no; Kevin Vermeulen, LAAS-CNRS, France, kevin.vermeulen@laas.fr; Maxime Mouchet, IPinfo, United States, max@ipinfo.io; Vasilis Giotsas, Cloudflare, United States, vasilis@cloudflare.com.

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only. Request permissions from owner/author(s).

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 2834-5509/2024/9-ART15
https://doi.org/10.1145/3676869

Despite this wide variety of use cases, the existing Internet-scale IP geolocation datasets are commercial and closed-source, while various validation efforts by the research community have found systematic geolocation errors [22, 30, 43, 59].

To address these issues, geofeeds (Geolocation Feeds) have been proposed as a means for operators to provide the geolocation of their own IP addresses, so so that this accurate information an be used to improve location-specific services [42]. These data have the potential to greatly contribute in addressing both operation and research needs; operators are the authoritative source of information on their own IP addresses, and therefore information published by them is in theory more accurate, complete, and up-to-date compared to inferential techniques. However, as we have seen in other cases of self-published network datasets, such as PeeringDB [12], or the Internet Routing Registries (IRR) [49], it may include incorrect or stale information [26, 27, 44, 51]. We seek to assess geofeeds in terms of accuracy, usability and adoption by operators.

To this end, we make the following contributions:

- We measure the adoption, coverage and network properties of the Autonomous Systems (ASes) and their corresponding IP space included in geofeeds.
- An evaluation of the accuracy of the geofeeds for client, router, and server IP addresses.
- We discuss the applicability of geofeeds in a research and operational context and potential improvements needed to address geofeeds shortcomings.

We find that geofeeds are on their way of being widely adopted. As of October 2023, the geofeeds published in WHOIS cover 1.5% of the allocated IPv4 prefixes, and 0.7% of the IPv6 prefixes. Overall, geofeed IP addresses span 8,316 cities in 249 different countries. However, we also observe large disparities between different continents, with Africa and Latin America having the smallest fraction of registered geofeed IPs. Geofeeds also include invalid locations, indicating that geofeed data hygiene should be considerably improved.

In terms of geofeed accuracy, we find that although most geofeed locations are valid, 0.9%, 4.1% and 8.5% of the client, router and server IP addresses have an incorrect geofeed. This rate of incorrectness prevents us from considering geofeeds as ground truth without further verification. More generally, today's geofeeds cannot be used as a standalone geolocation method, but can serve as a viable geolocation source that complements other geolocation methods and contributes in reducing measurement overhead to geolocate an IP address.

Finally, based on the experience of a geolocation database provider, we list recommendations to improve the process of sharing and formatting the geofeeds. We encourage operators to share their geofeeds through WHOIS records, following RFC 9092 [17], to facilitate discovery and improve security. We also call for an evolution of the geofeed format to allow operators to provide more context, for example to distinguish between user and infrastructure location in a CG-NAT setup.

2 Background and Related Work

2.1 IP Geolocation Feed Background

A geofeed (Geolocation Feed) is an IP geolocation data format which network operators can use to publish their IP address space geographic location information. Organization choose to publish geofeed as these in theory offer an accurate and up-to-date IP location information, which in turn can be used by both content and Internet service provider to personalize location-based service [17]. This geolocation format has been initially proposed in 2013 as an IETF informational document [41], and published in 2021 as RFC8805 [42]. Network operators publish their geofeed files as CSV files [58], where each line is comprised of five fields: *IP address block*, *country code*, *geographic region*, *city* and *postal code*. The IP block is mandatory, while the rest of the entries are

optional. Organization must use the ISO 3166 standard for country and region names [39], and the UTF-8 encoding [66] should be used for city names.

The Regional Internet Registries (RIRs) have incorporated IP geolocation in their WHOIS data [17]. Geofeed files can be currently found via the *geofeed* marker in `inetnum` objects. Leveraging this information, the *geofeed-finder* [47] tool can help user collect geofeed snapshots. More precisely, geofeed-finder extracts the geofeed URL from WHOIS records and fetches the IP prefixes in these files along with their geographic locations. The tool also validates the geolocation information format in the retrieved geofeeds. Organizations may opt not to list their geofeed information in the registry data, but share it via private communication with geolocation providers (see Section 3.5).

2.2 Related work

Many studies have focused on different aspects of IP geolocation. While our work was under submission, some recent work has been published looking at the adoption of geofeeds [29], looking at the adoption of the geofeeds and whether they were used by IP geolocation database providers. Our work adds the first study on the accuracy of the geofeeds, as well as adding more characterization about the cities reported in the geofeeds, and provide recommendations on their usage.

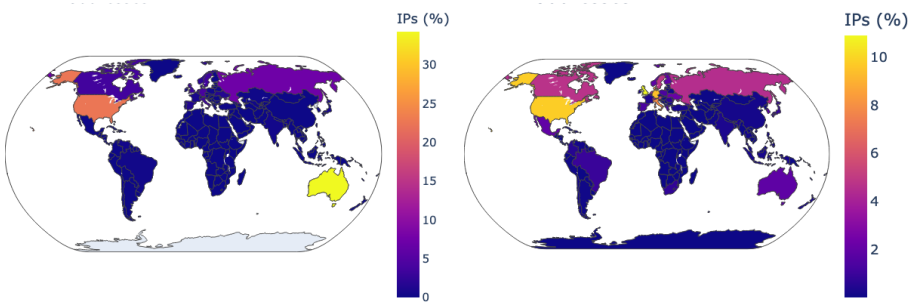
On IP geolocation techniques, over the last twenty years, the research community has focused greatly on designing inferential IP geolocation techniques. Such mechanisms leverage latency-based measurements [28, 35, 40], reverse DNS measurements [37] or the combination of both [21, 46, 56]. These approaches, however, come with a series of drawbacks that ultimately limit the resulting geolocation dataset. For instance, latency based measurement have a high measurement overhead preventing them to run at scale [22], despite efforts to make them scalable [36]. Reverse DNS based measurements usually have a poor coverage, as not every IP address has a reverse DNS name, and even when there is one, it might not contain a geolocation hint. Hoiho [46], the state-of-the-art method, geolocates 7.4% of the Internet-scale dataset of the Internet router level topology [18]. In addition, these techniques mostly cover infrastructure IP addresses, such as routers [37, 46]. We note, however, that recent work also looked at reverse DNS for end user IP addresses[21].

Moreover, although some techniques release their code [21, 46, 56] very few of them come with a publicly available dataset. Only the 2012 study by Hu *et. al* [36] produced such dataset. The RIPE team proposed Single Radius [25], a geolocation approach that uses latency based techniques in combination with network information such as peering information to reduce the possible locations of a router and the measurement overhead. This tool is deployed using the RIPE Atlas platform [53]. However, the resulting RIPE IPMap geolocation datasets still suffers in coverage, i.e., the March 2023 snapshot contained only 419K router IP address which is significantly lower that results obtained by state-of-the-art topology mapping techniques and systems [34, 62].

In parallel with the research community efforts, several commercial geolocation databases exists like MaxMind [48], IP2Location [38], and IPinfo [10] which map IP prefixes to geographic information like country and city. While these datasets have a high coverage, prior work has reported that they lack in accuracy [30]. These databases are closed-source without providing their IP geolocation inferring approach, making it thus difficult to know which record is trustworthy.

3 What is the state of adoption of the geofeeds?

Using geofeed-finder, we collect geofeeds every two weeks over a period of 3.5 months, from the half of June to the beginning of October 2023. We retrieve entries for 73,827 IPv4 and 46,549 IPv6 prefixes. We note that 97% of these entries are compliant with the RFC specification and 90% include at-least the country and the city. However, none of the entries in the geofeeds include the postal code. Moreover, we find 135 IPv4 and 143 IPv6 prefixes without any location information. These



(a) IPv4 addresses (b) IPv6 addresses

Fig. 1. Percentage of IPv4 (a) and IPv6 (b) addresses per country listed in the geofeed files.

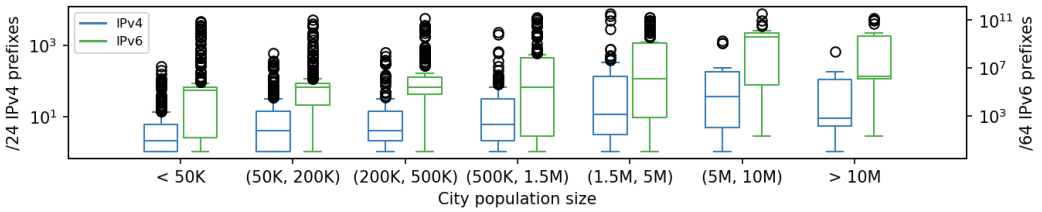


Fig. 2. /24 IPv4 and /64 IPv6 prefix distribution for different city sizes.

numbers are encouraging and shows that geofeeds are adopted by many operators, with the bulk of them providing city-level information.

In this section we characterize geofeed entries: (1) Where are they located? (2) Which part of the IP space are they covering? (3) How much of their parent organization are they covering? and (4) How stable are the locations?

3.1 Geographic spread

Countries. Geofeed IP prefixes are mapped to 249 different countries. The number of geofeed prefixes varies greatly per country. The country with the most prefixes is the US, with 24,056 IPv4 and 17,800 IPv6 prefixes. Canada, Australia and Germany also have a significant number of geofeed prefixes. However, 80% of the countries have at most 10 IPv4 and 100 IPv6 geofeed prefixes.

Figure 1 shows the world map distribution of IPv4 and IPv6 geofeed IP addresses per country. For IPv4, Australia and the US stand out, while in IPv6, the US, some EU countries, Canada and Russia stand out. Australia's large number of geofeed IPv4 adresses is a direct consequence of the high number of large IPv4 prefixes mapped to this country, i.e., 441 prefixes with the prefix length between /11 and /20. For the US, we find geolocated 123 prefixes between the same boundaries.

Cities. Having seen that IP address blocks are mapped through geofeeds across the world, we investigate whether prefixes are mapped to large or small cities. Geofeed files only contain city names, and thus we use the GeoNames 500 dataset [8], which reports cities with more than 500 inhabitants, to extract the cities' corresponding population information For the 8,316 geofeed location with both country and city details, we map them to the entry in the GeoNames dataset that has the same spelling and country, also using alternate names available in the GeoNames dataset, as the geofeed specification does not enforce a specific language for the city name. If there are

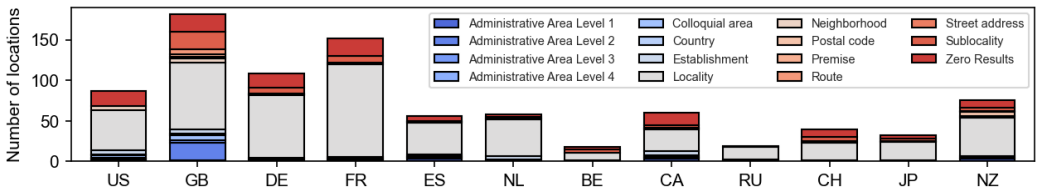


Fig. 3. Geofeed location type per country as retrieved for top countries.

multiple matches, we retain the one with the highest number of inhabitants. We rely on the OECD classification of cities based on their population [61] to split the cities into different categories: Small-size urban areas (50k, 200k), medium-size urban areas (200k, 500k), metropolitan area (500k, 1.5M), and large metropolitan areas (> 1.5M). To add finer granularity we add areas with < 50k inhabitants, areas between 1.5 M and 5M, areas between 5M and 10M, and areas above 10M.

We find that 71% of the cities observed in geofeed entries match the GeoNames data, mapping to cities of different sizes. Using our OECD augmented classification, we find that 84% of the selected locations in the geofeed files are medium-size urban areas or smaller. The 16% remaining include 184 cities with a population higher than 1.5M inhabitants, and 21 reported locations are large metropolitan, including very big cities such as Mumbai and New Delhi in India, and Beijing and Shenzhen in China. In addition, Figure 2 shows the number of /24 IPv4 and /64 IPv6 address prefixes in the geofeeds for each city class. Unsurprisingly, the median number for IP addresses grows as the city population increases, for both IPv4 and IPv6, except for the last level of 10M. Furthermore, we analyze whether there is a strong correlation between the city population size and the number of geofeed IP addresses, computing the Pearson correlation coefficient. We obtain 0.16 and 0.22 for for IPv4 and IPv6 addresses, respectively, indicating that overall there is no strong correlation between the number of geofeed IP addresses located in a city and the city size. Note that for completeness we show in Appendix A the scatter plot of the number of IP address blocks versus the city.

Unmatched city locations. We investigate the geofeed city locations with no match in the GeoNames dataset, calling them *unmatched*, and calling *matched* the others. On average, the percentage of matched city locations per country is twice the percentage of unmatched locations, but the distribution is not uniform. Half of the countries have less than 33% of their locations unmatched. However, 42 (17) countries have all locations matched (unmatched). Most of these are small countries and list their IP addresses in only a few locations. For countries with all the unmatched locations, the location is only the country in 35% of the cases. We focus on the unmatched city locations for countries with more than 150 geofeed IP prefixes. We query these locations in the Google Maps API to retrieve the geocoder results [32] which we further use to classify the locations into 14 different types (see Fig. 3). The *locality* geocoder type corresponds to an incorporated city or town political entity [32]. The following ones are *street names*, *postal codes* or *neighborhoods*, and even very precise locations, such as *establishments* like schools, city halls, tourist attractions or airport, e.g., we find /32 IPv4 address blocks mapped to the Maastricht Airport and Auckland Airport. Such small geographical granularity mapped to very specific IP address blocks can potentially affect the anonymity the geofeed IPs. Conversely, there are also locations mapped to *administrative area levels*, i.e, different orders of civil entity below the country level, such as 'Florida' (US) and 'Normandy' (FR). Some operators map their IPs to entire countries like Spain and Japan. Finally, "Zero results" indicate no geocoder results which mainly correspond to locations with typos, such as "Hambug" and "ZÃ¼rich" instead of "Hamburg" and "Zürich". We find cases of inaccurate city and country geographic pairing like ("NL", "London").

IPv4	/11	/14	/15	/16	/17	/18	/19	/20	/21	/22	/23	**/24**	/25	/26	/27	/28	**/29**	**/30**	**/31**	**/32**
Count	1	1	2	40	77	157	231	390	468	1108	1777	**14,638**	589	693	1,144	2,179	**10,626**	**6,967**	**6,336**	**23,742**
Percentage	0.001	0.001	0.002	0.056	0.108	0.220	0.324	0.548	0.657	1.557	2.497	**20.568**	0.827	0.973	1.607	3.062	**14.931**	**9.789**	**8.903**	**33.361**

Table 1. Number and percentage of geofeed IPv4 prefixes per prefix length. Half of the prefixes are wither IPv4 /24s or IPv4 /32s.

IPv6	/28	/29	/30	/31	/32	/33	/34	/35	/36	/37	/38	/39	/40	/41	/42	/43	**/44**	/45
Count	3	124	4	7	252	3	31	3	118	40	116	214	309	3	27	21	**3,204**	103
Percentage	0.006	0.270	0.008	0.015	0.550	0.006	0.067	0.006	0.257	0.087	0.253	0.467	0.675	0.006	0.059	0.045	**6.998**	0.225
IPv6	/46	/47	**/48**	/49	/50	/51	/52	/54	/55	**/56**	/60	/62	**/64**	/125	/126	/127	/128	
Count	255	710	**22,066**	8	15	11	347	1	13	**2,084**	6	2	**15,270**	1	22	53	337	
Percentage	0.557	1.550	**48.197**	0.017	0.0327	0.024	0.757	0.002	0.028	**4.552**	0.013	0.004	**33.353**	0.002	0.048	0.115	0.736	

Table 2. Number and percentage of geofeed IPv6 prefixes per prefix length. Approximately 80% of the prefixes are IPv6 /48s or IPv6 /64s.

Takeaways. Geofeeds prefixes are located around the world as most of the countries have at least few IPs mapped to them, and we find a correlation between the country size and the number of prefixes mapped to the geofeed countries. At a lower granularity, operators map their IPs to small, mid-size and large cities. Our city validation, however, reveals some location inaccuracies that we trace back to typos, city neighborhoods and towns and as well as country regions.

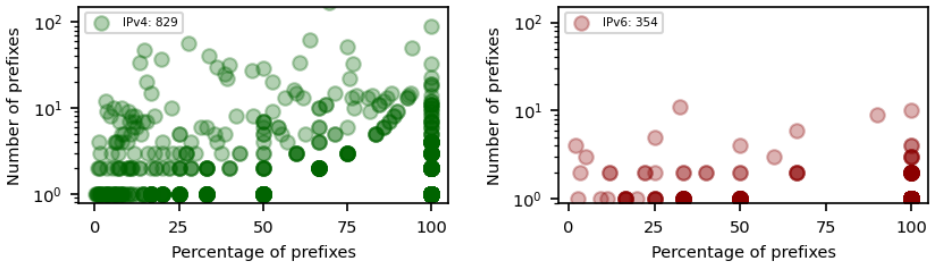
3.2 IP address space coverage

IP Prefix length. Network operators publish prefixes of different length in the geofeed files. To analyze whether operators geolocate more or less specific prefixes, we compute the number (percentage) of geofeed prefixes per prefix length. Tables 1 and 2 list these values for IPv4 and IPv6 prefixes. Most of the IPv4 prefixes are /24s or more specific prefixes, in particular from /29s to /32s, while IPv6 prefixes are either /48 or /64. Thus, network operators use a more specific length than the commonly accepted most specific length in BGP for both IPv4 and IPv6 geofeed prefixes, as the prefix length with the most occurrences is /32 for IPv4, and /48 and /64 for IPv6. However, this does not mean that these geofeeds cover only one client, as for instance ISPs and mobile carriers can hide behind a /32 due to the use of CG-NAT to share IPs between clients. Additionally, the spatial resolution of the CG-NATed prefixes is constrained by the geographical distribution of the users. Some ISPs might assign a /32 per state or region, while some might use a /32 for the whole country, in which case the geofeed entry is only useful to infer the country. We know that this case exists because some network operators reached out to IPinfo explicitly stating the presence of CG-NAT prefixes in geofeeds, however, we do not know how prevalent this is amongst all ASNs.

In other cases, operators can use /32 prefixes in geofeed files as this prefix length expresses the granularity at which operators write their network management policies. For example, after private discussions with a geofeed provider, we know that operators may use network automation tools such as Salt for both configuring their network policies and export geofeed data. The prevalence of /48 and /64 prefixes can be explained by typical IPv6 allocation patterns: it is recommended that operators attribute a /48 per end-user site [63], while /64 is the default subnet size for IPv6 [23].

Coverage in the registry data. Our next step is to analyze the coverage of the geofeed prefixes in the IP address space. We thus map the geofeeds to information extracted from Regional Internet Registry (RIR) delegation data [13]. Such data provides a list of the allocated and assigned Internet resources within each region. Moreover, the extended version of these files includes an *opaque-id* that identifies an organization within a single file [1]. We compute the overlap between allocated

[1]RIRs do not guarantee the opaque-id consistency over time.



(a) IPv4 prefixes (b) IPv6 prefixes

Fig. 4. Number versus Percentage of geofeed prefixes per organization for IPv4 (a) and IPv6 (b) addresses. The legend list the number of organizations that list their IP prefixes in the geofeeds.

and geofeed prefixes during our measurement period, finding that 1.50% and 0.70% of the allocated IPv4 and IPv6 prefixes, respectively, cover the geofeeds prefixes. Moreover, using a snapshot[2] of this data, we find that geofeed IPs represent only 0.4% of the overall allocated IP addresses.

Organization geofeed policies. We look at the organizations that provide geofeeds, and when they do, we quantify the fraction of their allocated IP prefixes published in their geofeeds. We use the RIR delegation data to find organizations listing their geofeeds as follows. First, we map to opaque-id to its allocated IP addresses and Autonomous System Numbers (ASNs). At the same time, we match the geofeed prefixes with the registered IP prefixes. Then, we select organizations for which their registered IP address space overlaps the geofeed IP addresses. Averaging across our collected snapshots, we find that approx. 828 organizations publish geofeeds, whereas it is 951 for our last geofeed snapshot. One quarter of these provide geofeeds for both IPv4 and IPv6 prefixes, while 62.7% and 12.8% list only IPv4 and IPv6 geofeed prefixes, respectively.

For each organization we compute its percentage of geofeed IPs prefixes. We show this percentage along with the number of geofeed prefixes in figure 4 for the 829 IPv4 and 354 IPv6 organizations. We list in the legend the number of organizations. Our analysis shows a high variability in the fraction of covered prefixes in the geofeeds. Some organizations provide geofeeds for all of their allocated IP prefixes, while others provide geofeeds for only a part of them. For IPv6, 74.3% of the organizations provide geofeed for all their allocated prefixes. However, most of these have only one allocated IPv6 prefix. For IPv4, approx. half of the organizations provide a geofeed for all of their allocated IPv4 prefixes, with one quarter of them having more than two allocated IPv4 prefixes.

Takeaways. Published geofeeds account for a small fraction of the allocated IP addresses which is most likely due to the early stage of their adoption process. Thus, a follow up longitudinal study would help in understanding whether more network operators adopt geofeeds or not. Additionally, we find that there is a high variability in the fraction of IP prefixes that operators publish in their geofeeds. We hypothesis that one reason is that the IP address space of a provider can have different usages. For instance, a provider could have prefixes for users and prefixes for routing infrastructure.

3.3 Size and type of the ASes publishing geofeeds.

Having seen that geofeeds cover a significant fraction of the IP address space, we further analyze the organizations that own these IP addresses. Using WHOIS data, RIR delegation data[13], and CAIDA's AS-to-Organization dataset [3], we infer ASes mapped to organization that list their

[2]We choose 2nd of October 2023 as this date corresponds to the last date of measurement period.

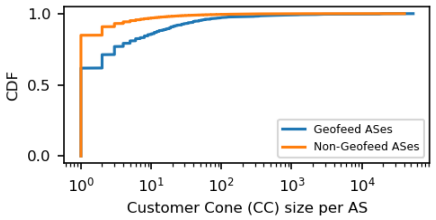


Fig. 5. Distribution of the customer cone size for Geofeed (Non-Geofeed) ASes owned by network operators that (do not) publish their geofeeds.

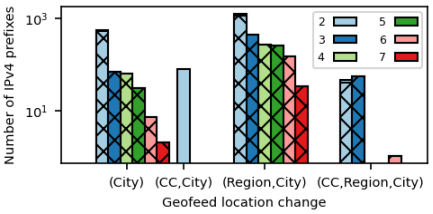


Fig. 6. Geofeed IPv4 prefixes mappped to multiple locations during the measurement period. The color indicates the number of location changes.

IP prefixes in the geofeed files. Recall that we group IP addresses and ASes that have the same opaque-id in the RIR delegation files. Thus, we extract 1,785 ASes for 773 of the 951 opaque-id obtained from the last geofeed snapshot data. One third of these ASes also publish their geofeeds.

To assess the type and size of the inferred ASes we use information from the PeeringDB database [12] and the CAIDA's customer cone dataset [45]. In order to identify the AS type, we search and extract the AS type for the selected ASes from PeeringDB [12], finding AS types for 32% of the ASes. These are distributed mainly among three different types, "Cable/DSL/ISP" (9%), "Network Service Providers (NSP)" (7.6%) and "Content" (5%). As we have only a 32% coverage with the PeeringDB AS type classification dataset, we also analyse the ASes publishing their geofeeds based on the size of their customer cone using AS Rank [4], where the minimum customer cone is one. Figure 5 shows the distribution of the customer cone size per AS for ASes publishing at least one geofeed (Geofeed ASes line) and ASes that do not publish any geofeed (Non-Geofeed ASes line). The majority of the ASes publishing geofeeds are stub ASes with a customer cone size of one. However, there are nonetheless 10% ASes with a customer cone size of 19, corresponding to small to medium transit providers. Comparing with the Non-Geofeed ASes line, we see that the proportion of ASes with a customer cone size of one is higher for the Geofeed ASes than Non-Geofeed ASes.
Takeaways. A relatively small number of organizations list their IP address blocks in geofeed files. These organizations are mostly ISP networks and content providers with a low customer cone size.

3.4 Geolocation dynamics

Spatial dynamics. Given our short period of measurement of three months, our intuition is that few IP prefixes should have multiple locations in different snapshots. A location is defined by the tuple *(country code, region, city)* with the city location matched from the GeoNames data. We find 3,575 IPv4 and 119 IPv6 prefixes with at least two locations, and anecdotally we even find IPv4 prefixes moved seven times. Figure 6 shows the number of times an IPv4 prefix changed its location when it changed at least once, and the type of information that changed in the geofeed. Most prefixes that changed their location once either changed their region (61.5%) or their city (20.9%).

We quantify the location change for IPv4 prefixes that switched location at least once, by computing the maximum distance between each pair of locations. Half of the prefixes moved within 200 km. However, we also find 25% of the prefixes with locations that are 1,100 apart km. Finally, we find one /32 that moved approximately 17,000 km from Australia, Sydney to Great Britain, London. We give some potential explanations for this behavior in Section 4.
Temporal dynamics. To analyze the temporal aspect of the geofeed dynamics we use the methodology from prior work [33], computing the Jaccard distance between the set of prefixes in two different geofeed snapshots. The median Jaccard distance between two geofeed snapshots for IPv4

is 0.13 and 0.07 for IPv6. The full distribution is shown in figure 10 in Appendix B. These results are higher than what was found in prior work for geolocation database, but are due to the progressive adoption of geofeeds rather than prefixes appearing and disappearing in the geofeeds. To quantify the latter phenomenon, we compute the stable and unstable prefixes. We label a prefix as stable if this prefix or a less specific one containing the prefix consistently appears in the geofeed snapshots after its first appearance, and unstable otherwise. Overall, 2,551 (3.45%) IPv4 and 9,174 (19.7%) IPv6 prefixes are unstable during our measurement period. This temporal instability suggests that one should not take only one snapshot of geofeeds when using them.

Takeaways. During our measurement period, most of the geofeed prefixes have one location. However, a small set of geofeed prefixes are mapped to multiple locations. If most geofeed prefixes that experienced a change move by less than 200 km, some of them experience a bigger change, and can even change their country. Moreover, some prefixes appear to be unstable over time. Thus, usage of the geofeed data requires multiple snapshots and filtering only the stable geofeeds.

3.5 Geofeeds obtained outside WHOIS data

Our analysis relied on publicly available geofeeds found in WHOIS records. However, some operators publish geofeeds that are not referenced in WHOIS records. This might be due to administrative reasons, for example the entity publishing the geofeed has no easy mean to update objects in the RIR database, or for the lack of knowledge that geofeeds can be advertised in WHOIS (RFC 9092).

IPinfo use geofeeds in its geolocation pipeline. While the majority of these feeds are obtained from WHOIS records, it also accepts manual submissions as a mean of submitting bulk corrections. By going to *IPinfo* website, operators can submit a file in the geofeed format.

Throughout this process, *IPinfo* obtained 1102 feeds, with 922 containing 1,117,972 distinct IPv4 ranges in 4233 ASNs, and 561 containing 645,072 distinct IPv6 ranges in 1592 ASNs. 375 of those feeds were later added to WHOIS records. If we exclude them we count 1,022,657 distinct IPv4 ranges in 2220 ASNs and 619,376 distinct IPv6 ranges in 654 ASNs.

Amongst the initially manually submitted feeds that were later found in WHOIS records, two of them covered a large number of ASNs: LACNIC[3] (1263 ASNs) which is automatically generated from information submitted by their members and IPXO[4] (505 ASNs), an IP leasing company.

Manually submitted geofeeds represent a challenge as there is no automated way of authenticating them. Anyone could submit a geofeed for ranges that it does not own or operate. To mitigate the risk of using bad or stale data, *IPinfo* relies on RTT validation and expiring geofeed entries whose ASN or range-ownership changed since the submission time. In addition *IPinfo* encourages operators to add manually submitted to their WHOIS record, if feasible.

4 How accurate are the geofeeds?

Since geofeeds are self-reported by network operators, they might contain inaccuracies and thus not usable as ground truth data before their accuracy is an asset. We divided our study of accuracy by IP address type, i.e., IPs of clients, servers, and routers IPs, since most research and commercial applications focus on a specific type of IP address [34]. For example, CDNs are interested in geolocation of their client IP addresses to redirect them to their closest site [19, 57, 67], while the research community might also be interested in geolocating servers [65] or routers [24].

In a nutshell, by perfoming a validation of the location based on obtaining small RTTs from vantage points with known geolocation, our main results are that although most of the geofeeds look correct, they should not be considered as ground truth, especially for router and server IP

[3]https://www.lacnic.net/4874/2/lacnic/lacnic-geofeeds-service
[4]https://geofeed.ipxo.com/geofeed.txt

addresses. We also look at the reasons behind some geofeed being wrong, finding cases where the geofeeds can be trusted despite the invalidating measurement, but also cases where some provider might have incentives to voluntarily put erroneous geofeeds.

4.1 Dataset

4.1.1 Client IP addresses. Cloudflare operates a CDN with more than 300 Points of Presence (PoPs) across more than 100 countries. For a sample of HTTP(S) connections, the CDN logs relevant statistics including the TCP Round-Trip Time (RTT) of the connection, which we use to estimate the distance between the clients' IP addresses and the CDN's PoPs. We randomly subsampled the CDN connection log over a limited time window [5]. Overall, the CDN logs we obtained include connections from more than 277 million client IP addresses, of which 2.5 million IP addresses are present in the geofeeds spread across 135 countries and 2,306 cities.

We use this dataset rather than pinging the public IP addresses of the clients seen in the requests, as the clients can be located behind NATs or CG-NATs that are often unresponsive to pings and can span large geographic areas. Distinguishing clients behind a NAT or a CGNAT is a non trivial problem [54], so we still group the RTTs per source IP. This makes the validation of geofeeds stricter: within a set of RTTs between different PoPs and the same IP address, if one RTT violates the geofeed location but another validates it, we consider the geofeed invalidated for the IP (Sec. 4.2).

4.1.2 Router IP addresses. We collect 2 months of RIPE Atlas[53], M-Lab [1] and Iris [34] traceroutes, between September and October 2023, to find router IP addresses covered by the geofeed prefixes found in the dataset of Section 3, for a total of 8,684 IP addresses in 69 countries and 6 continents.

4.1.3 Server IP addresses. We perform an Internet-wide TCP scan executed in October 2023 from Rapid7 [2], and we keep the IP addresses with port 80 and port 443 open. Similarly to routers, we look at the IP addresses that are covered by geofeeds, bringing a total of 826,561 IP addresses in 150 countries countries and 6 continents. We randomly sample one IP address per geofeed prefix to reduce the measurement overhead on the RIPE Atlas platform for a total of 14,160 IP addresses.

4.1.4 Sanitizing the geolocation of RIPE Atlas probes. We follow the methodology of prior work to remove the RIPE Atlas probes that might be wrongly geolocated [22], which consists in probing all the RIPE Atlas anchors from all the RIPE Atlas probes (including anchors), and compute the number speed of internet (SOI) violations given the indicated locations of the anchors and the probes. Then, we start by looking at the subset of the meshed measurements between the anchors and greedily remove the anchors that have the most violations, until no anchor has any SOI violations. We repeat the greedy removal for the rest of the probes, and remove 9 anchors and 96 probes.

4.2 Methodology

For an IP address within a prefix covered by a geofeed, our goal is to validate or invalidate its geofeed. Validating the geofeed means that we are able to find a vantage point close to the geofeed location and obtain a small RTT between the vantage point and the target, while invalidating a geofeed means that we can find a vantage point far from the geofeed location and also obtain a small RTT to the target. We first describe how we formally validate and invalidate geofeeds and then describe our methodology to find a vantage point with small RTT to a target.

4.2.1 Validating and invalidating geofeeds. Formally, we say that a geofeed is validated for a target IP address t if we can find a vantage point v so that:

$$(\mathrm{RTT}(v,t) < \alpha) \wedge (\mathrm{distance}(v, \mathrm{geofeed}) < \beta) \tag{1}$$

[5] We cannot reveal the exact time range due to commercial sensitivity concerns.

where α and β are two parameters representing how conservative the validation is. In the context of city level accuracy, we would like to use values of α and β so that we can conclude that the target is in the city indicated by the geofeeds. We present results for different values of α, between 0 and 5 ms, to not select an arbitrary threshold, but typically, an RTT threshold between 0 and 2 ms between a vantage point and a target has been used in prior work to consider the vantage point and the target are in the same city [31]. We discuss selection of the value of β below.

Similarly, formally, we consider that a geofeed is invalidated for a target IP address t if we can find a vantage point v such that:

$$\text{distance}(v, \text{geofeed}) > \text{distance\_from\_RTT}(\text{RTT}(v, t)) + \beta \quad (2)$$

This constraint translates the speed of the Internet in fiber ($\frac{2}{3}c$ [35]) violation between the location of the vantage point and the location of the geofeed. We add a β km offset to be conservative: As the granularity reported by operators is the name of the city without coordinates, we map a city to the coordinates found in the geonames dataset [9]. These coordinates generally map to a point in the center of the city, so adding an offset allows us to include the metro area (as this is the best accuracy that we can have for β, see below), so that if a vantage point is located in opposite cities of the same metro, we would increase the likelihood to not invalidate the geofeed. We thus derive categories to which an IP address can belong: validated (Eq. 1); invalidated (Eq. 2), or uncertain.

Selecting β: Ideally, we would use the diameter of each city. Some websites [5, 7] provide cities boundaries that would allow us to compute the diameter. Manually comparing for some cities these diameters with the ones obtained on Google Maps reveals that for most cities, the diameter in the datasets does not encompass the boundaries of the cities given by Google Maps, which looked correct for the cities that we knew, making the retrieved boundaries inaccurate and unusable. Instead, we approximate β, using OpenStreetMap [11] city data retrieved from Geoapify [6]. This dataset contains, for each city, a bbox field or "bounding box", representing the geolocation coordinates of a diagonal of a rectangle encompassing the city, so the corresponding distance is a good candidate for β. The left plot of Figure 8 shows that all the geofeeds cities except two (over 5,205) have a diagonal of 50 km, and we confirmed for 100 randomly sampled geofeed cities that their bounding boxes encircled the city. However, we found multiple cases where the bounding box was actually encompassing more than the city. Thus, we set β =50 km, and our geofeeds validation/invalidation should be considered at metro level rather than city, where the metro area represents a 50 km diagonal that encompasses the city, and we use metro level for the rest of the paper. Finally, note that one could use a different value of β to better suit the precision needed for its use case.

4.2.2 Obtaining small RTTs between vantage points and targets. The key to perform a good classification is to obtain low RTTs between vantage points and the target. For end-user IPs, we cannot directly ping the IP address from which we receive connections (Section 4.1), as some clients are behind a NAT or a CG-NAT middlebox, and therefore the location of the middlebox is not necessarily close to the actual end-user IP address [52]. We therefore apply two different methodologies to validate and invalidate the geofeeds depending on whether an IP address is an end user or not.

Router and server IP addresses: As we have a priori no reason to trust the geolocation given in a geofeed, we replicate the two-tier geolocation technique from previous work [22] to geolocate an IP address: We first greedily select 500 RIPE Atlas probes where each probe is selected to be the furthest possible from the already selected probes. Then, for each target, we run pings from these 500 RIPE Atlas probes to the target, and then compute the constraint-based geolocation (CBG) area given by the RTTs [35]. We finally run pings from all the RIPE Atlas probes that are located in the CBG area, and extract the RTTs. This technique has proved to achieve similar performance to find the probe with the shortest ping to a target as if we were trying all the RIPE Atlas vantage points,

	Invalidated	Validated with different values of α (ms)				Uncertain with different values of α (ms)			
		0.5	1	2	5	0.5	1	2	5
Clients	24,399 (0.9%)	124,073 (4.4%)	163,325 (5.8%)	190,331 (6.7%)	284,609 (10%)	2,689,295 (94.7%)	2,650,043 (93.4%)	2,623,037 (92.4%)	2,528,759 (89.1%)
Routers	274 (4.0%)	565 (8.3%)	1441 (21.1%)	3833 (56.1%)	4293 (62.8%)	5998 (87.7%)	5122 (74.9%)	2730 (39.9%)	2270 (33.2%)
Servers	789 (8.5%)	1726 (18.6%)	2968 (32.0%)	3977 (42.9%)	4478 (48.3%)	6756 (72.9%)	5514 (59.5%)	4505 (48.6%)	4004 (43.2%)

Table 3. Fraction of the IP addresses with their geofeed validated (Eq. 1), invalidated (Eq. 2) or uncertain. For a value of α, Invalidated + Validated + Uncertain sums up to 100%. Although geofeeds cannot be considered ground truth, most geofeeds can be validated for routers and servers when setting α=2 ms.

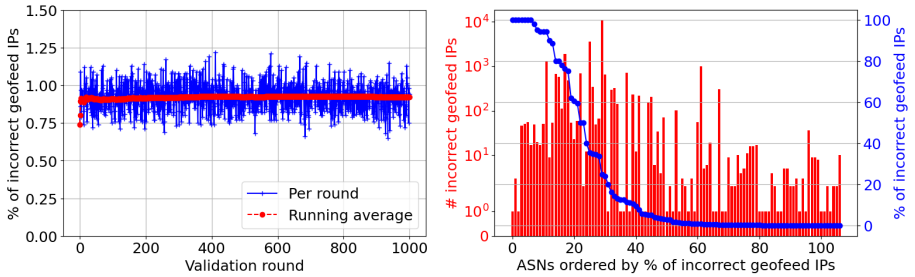


Fig. 7. Percentage of client geofeed IPs with an incorrect location according to the TCP RTTs observed in the CDN HTTP requests log. (Left) Results for each of the $1K$ validation rounds using a set of $10K$ randomly selected IP addresses. (Right) Results per ASN with at least one invalid geofeed IP address. The red bars show the number of invalid geofeed IPs per ASN (left y-axis - log scale), while the blue line corresponds to the percentage of invalid geofeed IP addresses per ASN (right y-axis).

while giving a good tradeoff to save measurement overhead [22]. Notice that obtaining a small RTT is only possible if there exists a vantage point close to the target, so our ability to validate and invalidate the geofeeds is limited by the coverage of the RIPE Atlas platform.

End user IP addresses: The RTTs are passively extracted from connections of HTTP clients that reside in the geofeeds IP address space to the PoPs of the CDN (see Sec. 4.1). To ensure that our validation is not biased by a specific subset of sampled HTTP client connections, we validated $10K$ randomly selected geofeed IP addresses against the TCP RTTs recorded in the CDN's HTTP requests log and we repeated this process $1K$ times.

Anycast prefixes: We do not expect geofeed prefixes with a metro level accuracy to be anycast prefix. Indeed, from the 64,029 IPv4 geofeed prefixes, we find that 72 (0.1%) are identified as anycast prefixes by the dataset provided by bgptools, which uses a methodology similar to MAnycast [60]. Nevertheless, if an IP address has its geofeed validated and is within an anycast prefix, we do not consider the IP address as validated and conservatively put it as uncertain.

4.3 Results

4.3.1 Classification of the geofeeds. Table 3 shows the number of IP addresses that fall into each category (validated, invalidated, uncertain), by type of IP address. The main takeaway is that we cannot consider geofeeds as ground truth, at least for routers and servers: 4.0% and 8.5% of the IP addresses have their geofeed invalidated. Second, these numbers are typically a lower bound as a significant fraction of the IP addresses are uncertain: Even with a high validation threshold $\alpha = 5$ms, 89.1%, 33.1%, and 43.2% or the client, router and server IP addresses fall in the uncertain category. The high difference between the numbers for the client IP addresses and the rest is explained by the coverage of the vantage points used to collect the RTTs. Whereas our CDN has about 300 PoPs,

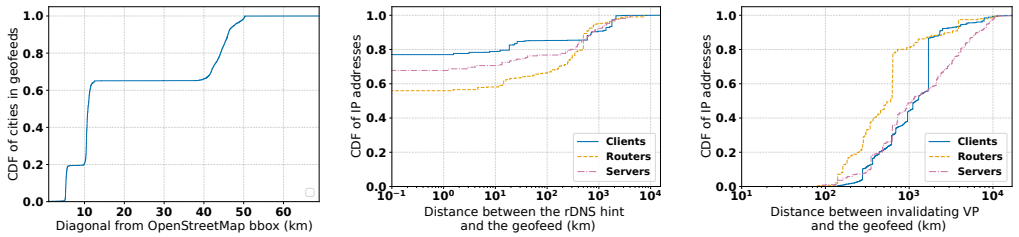


Fig. 8. (Left) Diagonal of cities in geofeeds of bounding box according to OpenStreetMap data. (Middle) Distance between the rDNS hint and the geofeed of IP addresses classified as uncertain. (Right) Maximum distance between the geofeed and a vantage point with a low RTT (<2ms) to a target covered by the geofeed.

RIPE Atlas has more than 10K vantage points, and so is more likely to be able to find a small RTT to a target, either to validate or to invalidate the geofeed. Additionally, broadband clients tend to have significantly higher last-mile latency compared to servers and routers, making it less likely to observe sufficiently low RTTs for client devices even when the client is located in the same city as a CDN PoP. Figure 7 plots the valiation results for client geofeed IP addresses.

Figure 7(a) shows the percentage of client geofeed IP addresses with an incorrect location for each of the $1K$ validation rounds using a randomly sampled subset of $10K$ IP addresses. While there are small variations between separate validation rounds (from 0.7% to 1.25%), the running average remains stable around 0.9%. However, we observe significant differences among ASNs (Figure 7(b)). Overall, there are 554 ASNs with geofeed IPs that appear in the validation dataset, of which 447 have no invalid IP location (not ploted in Figure 7(b)), and 107 have at least one IP address with invalid geofeed location. Although the majority of ASes with invalidated locations have less than 1% of their geofeed IP addresses invalidated, 22 of those ASNs have 40% or more invalid geofeed IP addresses. This results highlights that while geofeeds are accurate in the general case, there are non-trivial exceptions that need to be taken into consideration before considering geofeeds as ground-truth. However, we should caution that no definitive conclusion can be drawn for client IP addresses due to the high percentage of RTTs that fall within the "uncertain" category.

To investigate IP addresses falling in the uncertain category, we look if some of these IP addresses have a reverse DNS name containing a geolocation hint. Note that this does not constitute a ground truth validation, as we cannot validate the rDNS hints with small RTT measurements, otherwise the IPs would not fall into the uncertain category. Instead, this is more like a consistency check, as both geofeeds and rDNS are reported by the same entity, so if the two locations do not correspond, at least one of them is wrong. We extract the geolocation hints from the rDNS names with HLoc [56].

On the 2,623,037 client 2,730 router and 4,505 server IP addresses classified as uncertain with a threshold of α = 2ms, 10,361 (0.4%) clients, 321 (12%) routers, and 211 (5%) servers have a geohint in their rDNS name. The middle plot of Figure 8 shows the CDF of the distance between the rDNS hint and the geofeed, showing that most IP addresses have a consistent geofeed and rDNS with 86% of the clients, 64% of the routers, and 73% of the servers, that are less 50 km apart. However, seen in another way, 14% of the clients, 36% of the routers, and 27% of the servers have an rDNS geohint not in the same metro as their geofeed. We even have 9% of the clients, 5% of routers and 8% of servers with geohints at more than 1000 km! These results suggest that even if most IP addresses have consistent geolocation with rDNS and geofeeds, there is a significant part with uncertainty that emphasize our point that additional measurements are needed to validate them.

Characteristics of invalidated geofeeds: From the 24,399 (0.9%) client, 274 (4.0%) router, and 789 (8.5%)

server IP addresses with invalidated geofeeds, we extract the 9325, 196, and the 465 which have a vantage point invalidating the geofeed with a minimum RTT < 2 ms to the target, so that the target and the vantage points have a maximum distance of 200 km between them. If there are multiple possible vantage points, we select the one with the minimum RTT. Figure 8(c) shows the CDF of the distance between the vantage point invalidating a geofeed IP address and the geolocation of the geofeed itself. There is a significant fraction of geofeeds, 56% for clients, 19% for routers, and 51% of servers, that have a distance greater than 1000 km. Looking into more details about these mismatches, we find that there are 20 cases for routers and 137 cases for servers (10% and 29% of the invalidated geofeeds) where the country of the geofeed is not correct, showing that one should be cautious, even at country level, before using the geofeeds, especially for servers. To check for a potential reason for such mismatch, we look at whether the invalidated servers belong to anycast prefixes. Using the anycast dataset provided by bgptools, we find that on the 792 invalidated server IP addresses, 17 (2.1%) belong to anycast prefixes, showing that anycast is not a significant reason for why some IP addresses are invalidated. We give other potential explanations in the next section based on the feedback provided to our geolocation company by customers and network operators.

4.3.2 Other potential explanations for invalidated geofeeds. *IPinfo* provides a geolocation database mapping IP addresses to geographical coordinates, using a variety of active and passive measurements, as well as administrative data such as geofeeds. Because such data can be outdated or incorrect, every geofeed entries are validated using RTT measurements.

When a geofeed entry is invalidated by RTT measurements, it is excluded from the geolocation database until further inspection, and eventually an exchange with the network operators. This process has given us insights on the reasons a geofeed entry might not match RTT measurements.

It is important to note that an IP address can have two types of geolocation: the one from the host or router that bears the IP address and responds to pings, and the one from the users behind it (NAT). In this context, an invalidated geofeed entry might still be useful. For example, use cases such as advertisement or content personalization are concerned with the location of the users rather than the location of the infrastructure, whereas it might be the opposite for research use cases. We describe below the 4 main causes of invalidated geofeeds that we have identified.

Privacy-preserving VPN providers. NAT can also be used by privacy-preserving VPN providers which aim to improve user privacy by encrypting traffic and sharing a public IP, but do not necessarily aim to hide the location of the user. These services typically want to preserve the physical location of their customers, so that they are still served with the right local content (e.g., ads, weather, language). An example of such service is the iCloud Private Relay [55] whose geofeed[6] contains fine-grained /32 entries with multiple cities per country. These entries map to the user location [15], which may be different from the exit node location. Another example is centralized browsing solutions[7] where the traffic from a company is routed through a single exit node from another company, which aims to filter out malicious traffic.

Location-changing VPN providers. Some VPN providers aim to hide the location of their users and make them appear in other locations, for example to bypass content restrictions. Although in most cases the RTT measurements match the exit node location given by the provider, we have observed instances where they contradict the claimed location. One such example is the 95.181.238.0/24 which is registered and routed by M247. Its WHOIS description is *M247 LTD, Bahamas Infrastructure* and M247 geofeed[8] geolocates it in Nassau, Bahamas. This range is used by Private Internet Access (PIA), a VPN provider, for its Bahamas server. This can be verified by resolving the

[6]Apple Privacy Relay geofeed: https://mask-api.icloud.com/egress-ip-ranges.csv
[7]Palo Alto Secure Web Gateway (SWG): https://www.paloaltonetworks.com/sase/secure-web-gateway
[8]https://geoip.m247.ro/geofeeds.csv

`bahamas.privacy.network` domain contained in the PIA OpenVPN configuration file. However, all 125 pingable IPs within the range, including the IPs returned for `bahamas.privacy.network`, are reachable within 1.5ms of New York (112 are less than 1ms), which is incompatible with the 1700km distance between New York and Nassau. Previous research has highlighted similar behavior [65], suggesting that some VPN providers might try to artificially increase their location count by not giving the true IP address location.

Cloud geofeeds. Some geofeed contain entries slightly off their true location. For example, Google Cloud's geofeed[9] returns Lappeenranta as the geolocation for their cloud IPs in Finland, but to the best of our knowledge their only datacenter in the region is in Hamina, 87 km away. RTTs from Helsinki are also consistent with a network location in Hamina. It is unclear why Google does not give the true location of the datacenter, but it could be because Lappeenranta is a larger city than Hamina. We have observed a similar case in Taiwan, where GCP's geofeed returns Taipei, whereas the datacenter is actually 187km south in Changhua County.

Stale data. Finally, outdated and conflicting entries are an issue for geofeeds obtained through direct contact and not WHOIS records. In that case, there are few options besides manual inspection and contact with operators. We provide a set of recommendations to avoid these cases in Section 5.2.

5 Discussion and recommendations

We take a step back and put our findings in perspective to discuss whether geofeeds are ready to serve the operational and research community, and provide some recommendations that could potentially improve their usability.

5.1 Are geofeeds ready for prime time?

From an operational perspective, which is the first purpose of the geofeeds, the coverage given by the geofeeds does not yet meet the requirements of a large scale CDN. Section 4 showed that only 2.5 million over 277 million IP addresses (0.9%) were covered by the geofeeds. However, at the scale of a large CDN, 0.9% of its IP addresses can represent a high number of clients, even more if IP prefixes in the geofeeds span large geographic areas. Given that most of the geofeeds are correct for client IP addresses, only 0.9% were invalidated in our dataset, we can imagine that a large scale CDN can incorporate this information to help providing a good redirection. In particular, geofeeds can be helpful to CDNs to tune their redirection. For instance, if a CDN bases its redirection on the ECS option (the DNS option allowing to specify the prefix of the client in the DNS query), knowing that the ECS prefix is actually a CG-NAT prefix and that users are in fact located in another location than the one of the CG-NAT gateway can help redirecting them to the right PoP.

From a research perspective, geofeeds are a step towards obtaining an internet scale geolocation dataset. Although they cannot be considered as ground truth without additional verification (see Section 4), as 4.1% and 8.5% of routers and servers have their geofeeds invalidated, a subset of 56.1% and 42.9% of the routers and servers have been geolocated at city level. This shows that future geolocation techniques can use geofeeds as a hint for the IP geolocation, similar to rDNS usage and reduce the number of latency measurements needed to geolocate an IP address [22]. Finally, to evaluate the potential additional coverage that geofeeds would give, we compute the set of IPv4 addresses that have rDNS hints, the set of IP addresses that have geofeeds, and the union of both. We find 202M IPs with rDNS hints and 19M covered by geofeed entries, including 15M IPs without rDNS hints. This gives a total of 217M covered IPs. While a small increase, it is interesting to note that the overlap between the two datasets is small, making them complementary.

[9]Google Cloud geofeed: https://www.gstatic.com/ipranges/cloud_geofeed

5.2 Lessons learned

Operators should only publish their geofeed in WHOIS data. A significant number of operators do not publish geofeeds in their WHOIS records (Section 3.5), but rather share it via their website, or directly with IP geolocation database providers. However, these channels come with security risk, i.e., the IP geolocation database provider needs to trust unverified sources.

Geofeeds shared through WHOIS data benefit from the RIR's security procedures for login information into this data. For example an `inetnum` object in the RIPE database can only be edited by its designated maintainer. While the maintainer may not necessarily belong to the same entity that publishes the geofeed, this procedure restricts the ability to publish a rogue geofeed to entities explicitly. Another benefit of associating geofeeds to range objects is the ability to restrict geofeed entries to their parent range and to filter them down to the most specific range. Finally, fetching feeds from WHOIS data is easily automatable and reduce the risks of stale data being used (e.g., an operator publishing a new feed but not notifying users of the old one).

We strongly encourage the operators to associate geofeeds in WHOIS data, as specified in RFC 9092 [17], to facilitate automation and reduce risks of feed hijacking. This is especially important if entities such as CDNs were to rely on geofeeds to improve routing: in absence of authentication an attacker could craft a rogue feed to degrade network performance.

Geofeed users should perform sanity checks. In this study we found that geofeeds can contain both errors in the listed geographic information as well as inaccurate IP geolocation. Thus, geofeed users should not blindly trust the listed geographical location, but rather use public or in-house measurement platforms. For example, users can run ping measurements from vantage point geographically close to the locations listed in the geofeed files. Depending on the obtained delay value, users can opt to use/not use the tested geofeed.

Geofeeds should have more expressiveness. According to the RFC8805 [42], Network operators must use standardized format for the country and region information, but there is no such requirement for the city information. This can lead to difficulties to identify the geofeed location (Section 3). We recommend that operators use a standardized geographic spelling for the city field, for instance using the GeoNames [9] name and identifier for disambiguation.

Geofeeds should take into account for privacy. Malicious actors can use geofeed information to discover the geographic location of client IP addresses, or infrastructure IP addresses. Indeed, our work revealed cases where the geographic information is city neighborhood, small communities, or even buildings! We recommend that the geofeed standard takes this into account and operators only list cities that have a high number of inhabitants.

6 Conclusion

In this paper we took a first look at the geofeeds, measuring their adoption, their accuracy, and provided recommendations for future usage and how to improve their quality. Although we found that geofeeds cannot be considered as a ground truth, especially for Internet infrastructure IP addresses, our overall message about this geolocation dataset remains positive. Geofeeds provide at worse a good basis for IP geolocation, and one should start to consider them for operational or research purposes. Their current coverage does not yet meet the requirements to build an Internet-scale IP geolocation dataset based on the geofeeds, and our conclusions on their accuracy are therefore limited by the current coverage, so updating our study in the following years will certainly be of interest to understand if geofeeds become more popular and whether they can play a serious role for research.

References

[1] 2022. M-Lab. https://www.measurementlab.net
[2] 2022. Rapid7. https://opendata.rapid7.com
[3] 2023. AS to organizations mappings. https://catalog.caida.org/dataset/as_organizations. https://doi.org/dataset/as_organizations Dates used: 2023..
[4] 2023. CAIDA - AS Rank. http://as-rank.caida.org.
[5] 2023. GADM data. https://gadm.org/data.html.
[6] 2023. Geoapify OpenStreetMap cities, town, villages dataset. https://www.geoapify.com/download-all-the-cities-towns-villages.
[7] 2023. geojson-world-cities dataset from vdstech. https://github.com/drei01/geojson-world-cities.
[8] 2023. Geonames - All Cities with a population > 500. https://public.opendatasoft.com/explore/dataset/geonames-all-cities-with-a-population-500/table/?disjunctive.country.
[9] 2023. GeoNames geographical database. https://www.geonames.org/.
[10] 2023. ipinfo.io. https://ipinfo.io/.
[11] 2023. OpenStreetMap. https://www.openstreetmap.org/.
[12] 2023. PeeringDB. http://www.peeringdb.com.
[13] 2023. RIR Delegation (Extended) Files. https://ftp.ripe.net/pub/stats/.
[14] Ruwaifa Anwar, Haseeb Niaz, David Choffnes, Ítalo Cunha, Phillipa Gill, and Ethan Katz-Bassett. 2015. Investigating interdomain routing policies in the wild. In *Proceedings of the 2015 Internet Measurement Conference*. 71–77.
[15] Apple. [n. d.]. Prepare your network or web server for iCloud Private Relay. https://developer.apple.com/support/prepare-your-network-for-icloud-private-relay/. [Accessed 06-06-2024].
[16] Zachary S Bischof, Romain Fontugne, and Fabián E Bustamante. 2018. Untangling the world-wide mesh of undersea cables. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*. 78–84.
[17] Randy Bush, Massimo Candela, Warren "Ace" Kumari, and Russ Housley. 2021. Finding and Using Geofeed Data. RFC 9092. https://doi.org/10.17487/RFC9092
[18] CAIDA. 2023. Macroscopic Internet Topology Data Kit (ITDK). https://www.caida.org/catalog/datasets/internet-topology-data-kit/.
[19] Matt Calder, Ashley Flavel, Ethan Katz-Bassett, Ratul Mahajan, and Jitendra Padhye. 2015. Analyzing the Performance of an Anycast CDN. In *Proceedings of the 2015 Internet Measurement Conference*. 531–537.
[20] Igor Canadi, Paul Barford, and Joel Sommers. 2012. Revisiting broadband performance. In *Proceedings of the 2012 Internet Measurement Conference*. 273–286.
[21] Ovidiu Dan, Vaibhav Parikh, and Brian D Davison. 2021. IP geolocation through reverse DNS. *ACM Transactions on Internet Technology (TOIT)* 22, 1 (2021), 1–29.
[22] Omar Darwich, Hugo Rimlinger, Milo Dreyfus, Matthieu Gouel, and Kevin Vermeulen. 2023. Replication: Towards a Publicly Available Internet scale IP Geolocation Dataset. In *2023 ACM Internet Measurement Conference (IMC 2023)*. ACM.
[23] Dr. Steve E. Deering and Bob Hinden. 2006. IP Version 6 Addressing Architecture. RFC 4291. https://doi.org/10.17487/RFC4291
[24] Amogh Dhamdhere, David D Clark, Alexander Gamero-Garrido, Matthew Luckie, Ricky KP Mok, Gautam Akiwate, Kabir Gogia, Vaibhav Bajpai, Alex C Snoeren, and Kc Claffy. 2018. Inferring persistent interdomain congestion. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. 1–15.
[25] Ben Du, Massimo Candela, Bradley Huffaker, Alex C. Snoeren, and kc claffy. 2020. RIPE IPmap Active Geolocation: Mechanism and Performance Evaluation. *SIGCOMM Comput. Commun. Rev.* 50, 2 (may 2020), 3–10. https://doi.org/10.1145/3402413.3402415
[26] Ben Du, Katherine Izhikevich, Sumanth Rao, Gautam Akiwate, Cecilia Testart, and Alex C Snoeren. 2023. IRRegularities in the Internet Routing Registry. In *Proceedings of the 2023 ACM on Internet Measurement Conference*. 104–110.
[27] Ben Du, Cecilia Testart, Romain Fontugne, Gautam Akiwate, Alex C Snoeren, and Kc Claffy. 2022. Mind your MANRS: measuring the MANRS ecosystem. In *Proceedings of the 22nd ACM Internet Measurement Conference*. 716–729.
[28] Brian Eriksson, Paul Barford, Joel Sommers, and Robert Nowak. 2010. A learning-based approach for IP geolocation. In *Passive and Active Measurement: 11th International Conference, PAM 2010, Zurich, Switzerland, April 7-9, 2010. Proceedings 11*. Springer, 171–180.
[29] Rahel A Fainchtein and Micah Sherr. 2024. You Can Find Me Here: A Study of the Early Adoption of Geofeeds. In *International Conference on Passive and Active Network Measurement*. Springer, 228–245.
[30] Manaf Gharaibeh, Anant Shah, Bradley Huffaker, Han Zhang, Roya Ensafi, and Christos Papadopoulos. 2017. A look at router geolocation in public and commercial databases. In *Proceedings of the 2017 Internet Measurement Conference*. 463–469.

[31] Vasileios Giotsas, Thomas Koch, Elverton Fazzion, Ítalo Cunha, Matt Calder, Harsha V Madhyastha, and Ethan Katz-Bassett. 2020. Reduce, reuse, recycle: Repurposing existing measurements to identify stale traceroutes. In *Proceedings of the ACM Internet Measurement Conference*. 247–265.
[32] Google Maps Platform. [n. d.]. Geocoding request and response: Address types and address component types. https://developers.google.com/maps/documentation/geocoding/requests-geocoding#Types.
[33] Matthieu Gouel, Kevin Vermeulen, Olivier Fourmaux, Timur Friedman, and Robert Beverly. 2021. IP geolocation database stability and implications for network research. In *Network Traffic Measurement and Analysis Conference*.
[34] Matthieu Gouel, Kevin Vermeulen, Maxime Mouchet, Justin P Rohrer, Olivier Fourmaux, and Timur Friedman. 2022. Zeph & Iris map the internet: A resilient reinforcement learning approach to distributed IP route tracing. *ACM SIGCOMM Computer Communication Review* 52, 1 (2022), 2–9.
[35] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. 2004. Constraint-Based Geolocation of Internet Hosts. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC '04)*. Association for Computing Machinery, New York, NY, USA, 288–293. https://doi.org/10.1145/1028788.1028828
[36] Zi Hu, John Heidemann, and Yuri Pradkin. 2012. Towards geolocation of millions of IP addresses. In *Proceedings of the 2012 Internet Measurement Conference*. 123–130.
[37] Bradley Huffaker, Marina Fomenkov, and KC Claffy. 2014. DRoP: DNS-based router positioning. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 5–13.
[38] IP2Location. 2023. Identify Geographical Location and Proxy by IP Address. https://www.ip2location.com/.
[39] ISO (International Organization for Standardization). [n. d.]. ISO 3166 Country Codes. https://www.iso.org/iso-3166-country-codes.html.
[40] Ethan Katz-Bassett, John P John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson, and Yatin Chawathe. 2006. Towards IP geolocation using delay and topology measurements. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. 71–84.
[41] Erik Kline, Krzysztof Duleba, and Zoltan Szamonek. 2013. *Self-published IP Geolocation Data*. Internet-Draft draft-google-self-published-geofeeds-02. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-google-self-published-geofeeds/02/ Work in Progress.
[42] Erik Kline, Krzysztof Duleba, Zoltan Szamonek, Stefan Moser, and Warren "Ace" Kumari. 2020. A Format for Self-Published IP Geolocation Feeds. RFC 8805. https://doi.org/10.17487/RFC8805
[43] Ioana Livadariu, Thomas Dreibholz, Anas Saeed Al-Selwi, Haakon Bryhni, Olav Lysne, Steinar Bjørnstad, and Ahmed Elmokashfi. 2020. On the accuracy of country-level IP geolocation. In *Proceedings of the applied networking research workshop*. 67–73.
[44] Aemen Lodhi, Natalie Larson, Amogh Dhamdhere, Constantine Dovrolis, and Kc Claffy. 2014. Using peeringDB to understand the peering ecosystem. *ACM SIGCOMM Computer Communication Review* 44, 2 (2014), 20–27.
[45] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, and KC Claffy. 2013. AS relationships, customer cones, and validation. In *Proceedings of the 2013 conference on Internet measurement conference*. 243–256.
[46] Matthew Luckie, Bradley Huffaker, Alexander Marder, Zachary Bischof, Marianne Fletcher, and K Claffy. 2021. Learning to extract geographic information from internet router hostnames. In *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*. 440–453.
[47] Massimo Candela. 2023. geofeed-finder. https://github.com/massimocandela/geofeed-finder.
[48] Maxmind. 2023. GeoIP Databases. https://www.maxmind.com/en/geoip-databases.
[49] Merit Network, Inc. [n. d.]. The Internet Routing Registry . https://www.irr.net/docs/list.html.
[50] Reza Motamedi, Bahador Yeganeh, Balakrishnan Chandrasekaran, Reza Rejaie, Bruce M Maggs, and Walter Willinger. 2019. On mapping the interconnections in today's internet. *IEEE/ACM Transactions on Networking* 27, 5 (2019), 2056–2070.
[51] Arnold Nipper. 2018. PeeringDB Update.
[52] Philipp Richter, Florian Wohlfart, Narseo Vallina-Rodriguez, Mark Allman, Randy Bush, Anja Feldmann, Christian Kreibich, Nicholas Weaver, and Vern Paxson. 2016. A multi-perspective analysis of carrier-grade NAT deployment. In *Proceedings of the 2016 Internet Measurement Conference*. 215–229.
[53] RIPE NCC. [n. d.]. RIPE Atlas. https://atlas.ripe.net/.
[54] Loqman Salamatian, Todd Arnold, Ítalo Cunha, Jiangchen Zhu, Yunfan Zhang, Ethan Katz-Bassett, and Matt Calder. 2023. Who Squats IPv4 Addresses? *SIGCOMM Comput. Commun. Rev.* 53, 1 (apr 2023), 48–72.
[55] Patrick Sattler, Juliane Aulbach, Johannes Zirngibl, and Georg Carle. 2022. Towards a tectonic traffic shift? investigating Apple's new relay network. In *Proceedings of the 22nd ACM Internet Measurement Conference*. 449–457.
[56] Quirin Scheitle, Oliver Gasser, Patrick Sattler, and Georg Carle. 2017. HLOC: Hints-based geolocation leveraging multiple measurement frameworks. In *2017 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 1–9.
[57] Kyle Schomp, Onkar Bhardwaj, Eymen Kurdoglu, Mashooq Muhaimen, and Ramesh K Sitaraman. 2020. Akamai dns: Providing authoritative answers to the world's queries. In *Proceedings of the Annual conference of the ACM*

Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication. 465–478.

[58] Yakov Shafranovich. 2005. Common Format and MIME Type for Comma-Separated Values (CSV) Files. RFC 4180. https://doi.org/10.17487/RFC4180

[59] Yuval Shavitt and Noa Zilberman. 2011. A geolocation databases study. *IEEE Journal on Selected Areas in Communications* 29, 10 (2011), 2044–2056.

[60] Raffaele Sommese, Leandro Bertholdo, Gautam Akiwate, Mattijs Jonker, Roland van Rijswijk-Deij, Alberto Dainotti, KC Claffy, and Anna Sperotto. 2020. Manycast2: Using anycast to measure anycast. In *Proceedings of the ACM Internet Measurement Conference*. 456–463.

[61] The Organisation for Economic Co-operation and Developmen (OECD). 2014. Urban population by city size. https://www.oecd-ilibrary.org/content/data/b4332f92-en.

[62] Kevin Vermeulen, Justin P Rohrer, Robert Beverly, Olivier Fourmaux, and Timur Friedman. 2020. {Diamond-Miner}: Comprehensive Discovery of the Internet's Topology Diamonds. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. 479–493.

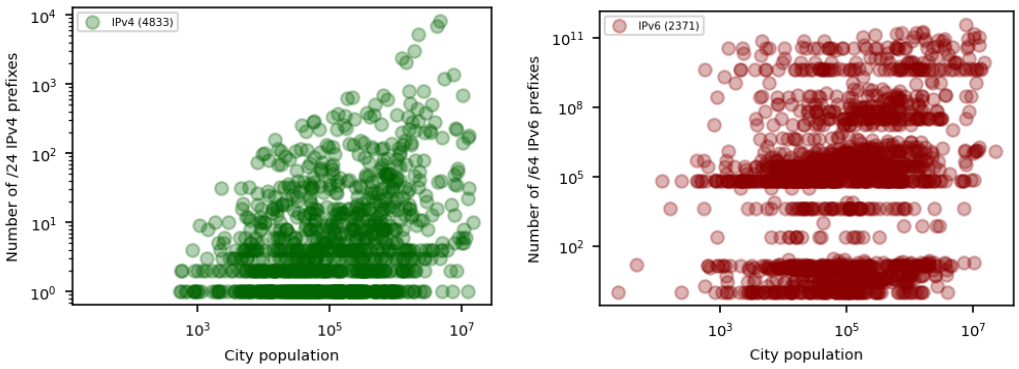
[63] Jan Žorž Sander, Sander Steffann, Primož Dražumerič, Mark Townsley, Andrew Alston, Gert Doering, Jordi Palet, Jen Linkova, Luis Balbino, Kevin Meynell, and Lee Howard. 2017. Best Current Operational Practice for Operators: IPv6 prefix assignment for end-users - persistent vs non-persistent, and what size to choose — ripe.net. https://www.ripe.net/publications/docs/ripe-690. [Accessed 05-12-2023].

[64] Brandon Wang, Xiaoye Li, Leandro P de Aguiar, Daniel S Menasche, and Zubair Shafiq. 2017. Characterizing and modeling patching practices of industrial control systems. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 1, 1 (2017), 1–23.

[65] Zachary Weinberg, Shinyoung Cho, Nicolas Christin, Vyas Sekar, and Phillipa Gill. 2018. How to catch when proxies lie: Verifying the physical locations of network proxies with active geolocation. In *Proceedings of the Internet Measurement Conference 2018*. 203–217.

[66] François Yergeau. 2003. UTF-8, a transformation format of ISO 10646. RFC 3629. https://doi.org/10.17487/RFC3629

[67] Jiangchen Zhu, Kevin Vermeulen, Italo Cunha, Ethan Katz-Bassett, and Matt Calder. 2022. The best of both worlds: high availability CDN routing without compromising control. In *Proceedings of the 22nd ACM Internet Measurement Conference*. 655–663.



(a) IPv4 addresses (b) IPv6 addresses

Fig. 9. City population versus number of /24 IPv4 (a) and /64 IPv6 (b) prefixes mapped to each city in the geofeeds.

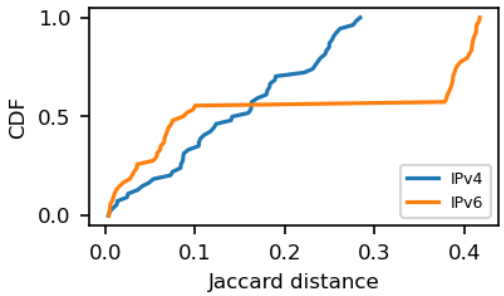


Fig. 10. Comparison of geofeed IPv4 and IPv6 prefixes collected in different snapshots.

A Geofeed city population

We rely on the Geoname dataset [8] to extract the city population of the matched geofeed locations. For each such city we compute the number of /24 IPv4 and /64 IPv6 address blocks and show in figure 9 these values per city. Our results show that operators geolocated their IP prefixes to cities of different sizes, i.e., from small-size urban areas to super metropolitans.

B Temporal dynamics

Figure 10 shows the distribution of the Jaccard distance between the set of geofeed prefixes for pairs of geofeed snapshots.

C Type of geocoder responses

We list and detail in table 4 the type of geocoder responses we retrieved when querying the Google MAP API. We use these responses in our analysis to classify the geofeed locations that did not match GeoNames data, i.e., inaccurate city locations. Our results shows that most of these locations are classified as locality, i.e., incorporated city or town. However, we find that operators map IP

Response Type	Details
Country	National-level political entity.
Administrative area level 1	First-order (political) entity within a country; US states, regions in other countries.
Administrative area level 2	Second-order (political) entity within a country like counties, cities and districts.
Administrative area level 3	Third-order (political) entity within a country. Second-order administrative areas that are townships, towns, municipalities, villages.
Administrative area level 4	Administrative areas smaller than townships or towns.
Colloquial area	Commonly (alternative) name for entities.
Establishment	Non-political entity like stores, schools and airports.
Locality	Incorporated city or town.
Neighborhood	District or community within an entity (eg. city).
Postal code	Address postal mail code used within a country.
Premise	Location name like one building or group of building referred as an unit.
Route	Named routes that can overlap with roads (eg. US 101)
Street address	Exact location of a street with an entity.
Sublocality	Top-level civil entity below locality.

Table 4. Geocoder response types for the inaccurate city locations (i.e., geofeed city locations that are unmatched against the GeoName data).

prefixes both to small-scale entities like airports, city halls or postal codes as well as large-scale entities like entire countries or regions of a country.

Received December 2023; revised June 2024; accepted July 2024