Uncovering BGP Action Communities and Community Squatters in the Wild

BRIVALDO ALVES DA SILVA JR., UFMS, Brazil
ADRIANO BASTOS DE CARVALHO, UFMS, Brazil
ÍTALO CUNHA, UFMG, Brazil
TIMUR FRIEDMAN, Sorbonne Université, France
ETHAN KATZ-BASSETT, Columbia University, USA
RONALDO ALVES FERREIRA, UFMS, Brazil

The Border Gateway Protocol (BGP) offers several "knobs" to control routing decisions, but they are coarse-grained and only affect routes received from neighboring Autonomous Systems (AS). To enhance policy expressiveness, BGP was extended with the *communities* attribute, allowing an AS to attach metadata to routes and influence the routing decisions of a remote AS. The metadata can carry *information* to (*e.g.,* where a route was received) or request an *action* from a remote AS (*e.g.,* not to export a route to one of its neighbors). Unfortunately, the semantics of BGP communities are not standardized, lack universal rules, and are poorly documented. In this work, we design and evaluate algorithms to automatically uncover BGP *action communities* and ASes that violate standard practices by consistently using the *information communities* of other ASes, revealing undocumented relationships between them (*e.g.,* siblings). Our experimental evaluation with billions of route announcements from public BGP route collectors from 2018 to 2023 uncovers previously unknown AS relationships and shows that our algorithm for identifying action communities achieves average precision and recall of 92.5% and 86.5%, respectively.

CCS Concepts: • **Networks** → *Network dynamics*; *Network management*;

Additional Key Words and Phrases: Internet Routing, BGP Communities

ACM Reference Format:
Brivaldo Alves da Silva Jr., Adriano Bastos de Carvalho, Ítalo Cunha, Timur Friedman, Ethan Katz-Bassett, and Ronaldo Alves Ferreira. 2024. Uncovering BGP Action Communities and Community Squatters in the Wild. *Proc. ACM Meas. Anal. Comput. Syst.* 8, 3, Article 49 (December 2024), 23 pages. https://doi.org/10.1145/3700431

1 Introduction

In the Internet, autonomous systems (ASes) define routing policies that govern how traffic is routed and enforce them using the Border Gateway Protocol (BGP) [43]. The BGP protocol is flexible and offers several parameters to control routing decisions—*e.g.,* setting route preferences (LocalPref), signaling preferred interconnections between a pair of neighboring ASes (MEDs), or minimizing intradomain traffic costs. These mechanisms, however, are coarse-grained and work only to control decisions for routes received from neighboring ASes. Nonetheless, network operators often need

Authors' Contact Information: Brivaldo Alves da Silva Jr., brivaldo.junior@ufms.br, UFMS, Campo Grande, Brazil; Adriano Bastos de Carvalho, adriano.bastos@ufms.br, UFMS, Campo Grande, Brazil; Ítalo Cunha, cunha@dcc.ufmg.br, UFMG, Belo Horizonte, Brazil; Timur Friedman, timur.friedman@sorbonne-universite.fr, Sorbonne Université, Paris, France; Ethan Katz-Bassett, ethan@ee.columbia.edu, Columbia University, New York, USA; Ronaldo Alves Ferreira, raf@facom.ufms.br, UFMS, Campo Grande, Brazil.

ACM 2476-1249/2024/12-ART49
https://doi.org/10.1145/3700431

finer-grained control to, for example, avoid traffic crossing a distant congested link or to spread traffic across multiple providers.

To enhance policy expressiveness, BGP was extended with the *communities* attribute [30], which allows an AS to attach metadata to routes. A BGP community is a transitive and optional attribute, which implies it should be carried across multiple ASes as routes propagate and ignored by ASes that do not know how to handle the community. Each AS can define its own *action communities* that other ASes may tag on routes to request special treatment from the AS. An action community can request the AS to prepend its AS number to the BGP AS-path to make a route artificially longer and less attractive or to not advertise a particular prefix to one of the AS's peers to steer traffic destined to that prefix away from a low-performance AS.

Unfortunately, the semantics of BGP communities are not standardized, do not follow universal and well-accepted rules, and are not well-documented [28, 47]. Network operators must rely on BGP community documentation manually built and provided by each individual AS, and the documentation may be incomplete, incorrect, outdated, or completely missing. Also, determining community semantics from route announcements presents many challenges, including partial visibility of routes on the Internet, communities that are only available to customers, and ASes that strip communities from route announcements after use.

In this work, we present mechanisms to uncover BGP action communities and an undocumented type of confounding use of BGP communities in the wild, information that might help operators understand BGP community uses and their implications. Specifically, we design and evaluate algorithms to automatically identify BGP action communities and ASes that violate common practice and consistently *squat*[1] the BGP communities of another AS, which we label as a *squatting relationship*. This behavior can impact previous researches that infer AS relationships or the semantics of BGP communities [20, 27, 28, 32, 47, 49].

Our approach relies solely on public route announcements observed by BGP route collectors (*e.g.,* RouteViews [35] and RIPE RIS [44]). It fundamentally differs from previous efforts that rely on public documentation about BGP communities—published by the networks on Internet Routing Registries (IRRs) or web pages—as a basis for classifying undocumented communities [28] or to extract community semantics using natural language processing [18, 21]. These approaches do not generalize well to ASes that do not follow common practices to define their communities and are limited in the number of communities they can infer because they depend on free text descriptions provided by network operators, which may be incomplete or outdated.

Our key insight lies in the fundamental difference between the usage of information and action communities. *Information communities* are used by ASes to pass information to other ASes, such as where the AS learned a route or its business relationship (customer, provider, or peer) with the previous AS on the route. Consequently, an information community should appear on routes that traverse the community's AS, as the AS is in charge of tagging routes with the relevant information. Conversely, an action community is less likely to be tagged on routes where its AS is present, as the community carries a request from and is tagged by a network other than the AS that defines the community. Also, RFC7454 prescribes that the controlling AS should remove an action community from a route after performing the requested action [15]. Therefore, if the AS that defines the community is on the route, it should have removed its action communities. As such, an action community should only appear if the route does not traverse its AS. Our algorithms rely on this

[1]We borrow the term *squat* and its derived forms from "IP address squatting" [45], where a network uses another's IP address space internally for its own purposes. In this work, however, an AS may *squat* the communities of another AS legitimately, *e.g.,* the communities of a sibling AS.

fundamental difference to build reliable classifiers of action communities and to uncover *potential squatters*.

While our insight is simple to state, designing algorithms that perform well in the wild presents significant challenges, such as ASes that squat the information communities of other ASes, ASes that do not remove their action communities after performing the requested actions, route announcements with a large number of communities, and limited visibility of the existing BGP route collectors. We address these challenges by identifying squatting relationships based on how routes with information communities propagate on the Internet. Then, we build an initial set with the action communities that are mostly absent from routes traversing the ASes that define them. Using this initial set, we construct an efficient data structure to identify action communities in route announcements where the ASes that define them can be present in the AS-paths.

We evaluate our algorithm for identifying action communities using a ground-truth dataset of action and information communities from 17 Tier-1, 14 Tier-2, and 43 other ASes, totaling 74 ASes. We use data gathered from IRRs, public web pages, and NL NOG [22] to determine the semantics of the communities in the ground-truth dataset.

Our experimental evaluation with billions of route announcements from 2018 to 2023 shows that the algorithm to identify action communities achieves precision and recall of 92.5% and 86.5%, respectively, averaged over all communities in BGP dumps covered by our ground truth in the longitudinal study. We also analyzed over 739 million announcements from December 2023 and inferred 19,564 action communities from 2,099 autonomous systems. Our algorithm for uncovering potential squatters found 54 pair-wise squatting relationships involving 105 ASes that systematically used another AS's communities in December 2023. These identified squatting relationships may uncover undocumented relationships between the ASes.

This work is a step towards a better understanding of the Internet's complex routing system. Our algorithms provide automatically updated metadata (*i.e.,* a database of action communities and potential squatters) that can benefit novel tools and models. For example, action community information can help operators troubleshoot routing anomalies, *e.g.,* when routes that follow an unexpected or undesired path carry specific action communities, and identify opportunities for traffic engineering, *e.g.,* when an operator observes preferable routes induced by action communities not publicly documented. Our results can also be used to help identify and flag announcements carrying BGP communities to perform route manipulation attacks [4, 5, 37]. We show that operators use action communities much more extensively than publicly available documentation would indicate. We identified action communities for 2,099 ASes, but found BGP community documentation for only 74 ASes.

Additionally, our algorithm to uncover squatting relationships can complement techniques for validating AS-relationship inferences, tracking route changes, and inferring sibling ASes. Of the 54 pairs of squatting relationships we uncovered, five are sibling relationships that the state-of-the-art technique described in [8] did not detect.

2 Background

Despite the flexibility built into BGP's best path selection algorithm [42], additional flexibility is provided by BGP communities to support, *e.g.,* more complex or fine-grained traffic engineering policies [30]. A BGP community is a 32-bit tag that can be attached to an announcement. A BGP announcement can carry an arbitrary number of BGP communities. The standards suggest, and most operators (but not all) follow the convention that the first 16 bits represent the AS number (ASN, a number that identifies the AS) of the AS that defines the community's semantics, in this paper

referred from now on as the *controlling AS*, and that the last 16 bits is an arbitrary operator-defined value [30].[2]

BGP communities are an optional, transitive attribute. Although transitivity means communities should propagate broadly, all router vendors provide configuration options to drop communities from announcements, and some vendors drop communities by default (*e.g.,* Cisco [24]). This limits community propagation and visibility in an uncontrolled manner [27], which imposes a challenge to inferring BGP community semantics and to our work.

Network operators have flexibility in defining semantics for BGP communities and are limited only by the (increasing) community-handling capabilities of BGP routers. BGP community semantics fit into two classes [14, 26, 27, 34]:

Action communities signal an action that an AS executes on behalf of another, and are usually used to trigger actions at a provider on behalf of a customer. Action communities generally influence the BGP path selection process or how routing announcements propagate to realize some traffic engineering policy [5, 27, 34]. Examples include adjusting LocalPref to make the route less preferable at the transit provider, prepending the BGP AS-path to make it longer and thus less preferable for other ASes (often used for backup routes), and constraining route propagation to a subset (or none) of the transit provider's neighbors. For example, 3356:70 is an action community that asks Level3/Lumen (AS3356) to decrease the LocalPref of a route to 70 (from the default 100), making the route less preferable. BGP communities can even impact traffic forwarding on the data plane, as it is commonly used to blackhole malicious traffic [21, 49].

Informational communities add metadata to a route announcement. Use cases include assisting operators with traffic engineering [31], troubleshooting issues, refining policies, and capacity planning [19, 32, 52]. Example metadata in informational communities include specifying whether a route was originated either internally or learned externally; whether external routes were learned from a customer, provider, or peer; or the location (city, country, or region) where the route was learned or originated. For example, 3356:2009 is an informational community added by Level3/Lumen to routes learned at San Francisco. Informational communities may be used by the controlling AS itself as well as downstream ASes.

Recent years have seen increased adoption of BGP communities by network operators [14, 49]. The percentage of routes received by BGP collectors with at least one community increased from 59% in 2018 to 71% in 2023, even taking into account the 3,07× increase (from 161,878,003 to 496,846,470) in the number of BGP routes in public table dumps across all RouteViews and RIPE RIS collectors [35, 44].

3 Challenges in Identifying Action Communities on the Internet

This section initially describes the expected way action communities should propagate in BGP routes. We then discuss common situations that violate the norm and present challenges for identifying them in the wild.

3.1 BGP Community Propagation

Given BGP community semantics, information communities should be tagged only on routes traversing their controlling ASes, as the controlling AS is the one that tags routes with the relevant information [47]. For example, an information community X:Y specifying that AS X received a route from a customer and a community X:Z specifying that AS X received a route in Europe can only be meaningfully added to a route by AS X.

[2]In this paper, we consider only 32-bit communities [30]. Although we do not analyze *extended* [50] or *large* [23] communities in this work as their use is still incipient, our techniques can be applied to them.

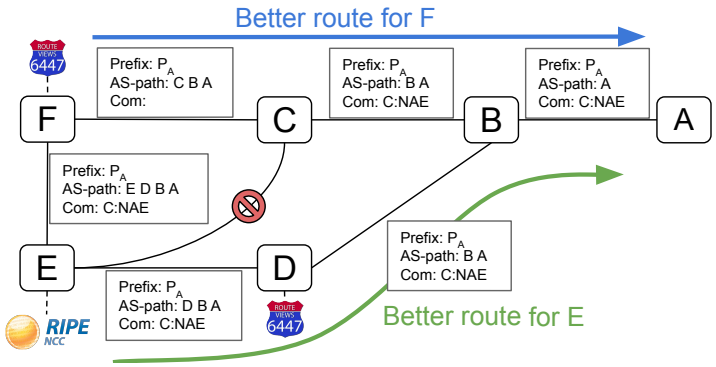


Fig. 1. Example illustrating how an action community is more likely to appear in routes that do not include its controlling AS. The community C:NAE instructs AS C not to advertise routes to AS E. We can observe the community C:NAE on routes without AS C exported by ASes D, E, and F.

On the other hand, action communities are less likely to be tagged on routes after traversing their controlling ASes due to multiple factors we discuss next. Figure 1 illustrates each factor; it shows propagation of a prefix P originated by AS A with action community C:NAE, which asks the controlling AS C to **n**ot **a**dvertise the route to AS **E**. Such a community could be used, for example, to steer traffic from AS E through AS D for load balancing or performance reasons.

(1) An action community X:Y is added to a route by other ASes to request that AS X takes action Y. A route tagged with X:Y may be received by other ASes and exported to BGP collectors without traversing AS X. In Figure 1, AS A added the action community to its announcement. The route propagates, carrying the community, and is exported to a collector by AS D without traversing the controlling AS C.
(2) Many action communities make routes less preferable by making them longer (prepending), reducing their preference (set LocalPref), or directly restricting propagation (no-advertise). As a result, routes with action communities that traverse the controlling AS are less likely to propagate compared to routes that avoid the target AS. In Figure 1, AS E does *not* receive a route from AS C, leading AS E to choose the route received from AS D, which does not traverse AS C.
(3) An action community has no use for ASes other than the controlling AS after the requested action has been taken, so ASes often remove their action communities from routes before propagating them [27, 49]. In Figure 1, AS C removes the community from the route it announced to AS F, which chooses a route through AS C that does not carry the action community.

3.2 Challenges

Although we expect action communities not to be tagged on routes traversing their controlling ASes, this is not always true. Several factors may lead to action communities being tagged on routes traversing their controlling ASes, making their identification challenging. Figure 2 illustrates some scenarios on routes for a prefix P announced by AS A.

(1) The controlling AS may take action on an action community and not untag it from the route due to unintended BGP configuration or by design (when the operator willfully propagates action communities). In Figure 2, AS C does not untag action communities from routes after taking the requested action. AS B tags community C:P2 asking AS C to prepend itself twice to the AS-path, and the community is observed with the controlling AS C on the route exported

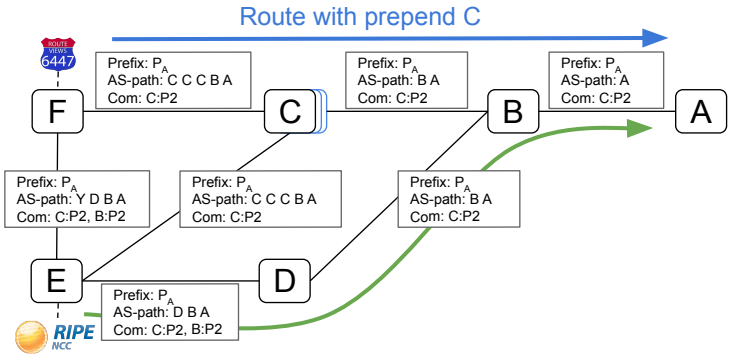


Fig. 2. Example illustrating scenarios where action communities may appear in routes traversing their controlling ASes. AS C does not remove its action communities from routes after taking the requested action, and AS D adds an action community for AS B in routes that have already traversed AS B.

by AS F to the collector. If these ASes propagate their action and information communities equally, then our inference algorithm may be penalized in accuracy and recall.

(2) The issue above is aggravated when the controlling AS does not act upon receiving an action community because of router misconfiguration or depending on the relationship with the neighboring AS from where it received the route, *e.g.,* an AS's routers may ignore action communities received from providers. In this case, the action community remains tagged on the route but does *not* reduce the route's preference; as route propagation is unconstrained, the route propagates broadly and causes the action community to be widely observed on routes traversing its controlling AS.

(3) An AS may uselessly tag a route with an action community *after* the route has traversed the controlling AS, which has no impact on the route itself but may happen depending on how the router is configured. In Figure 2, AS D adds community B:P2 uselessly asking AS B, which is already in the path and will not receive the community, to prepend AS B twice to the AS-path. The community B:P2 is observed with the controlling AS B on the route exported by AS E to the collector.

(4) An operator may define *non-standard* BGP communities, where the first 2 bytes are set to a value different than the controlling AS's number. For example, AS9002 (RETN) uses community X:65533 as an action community that asks "prepend AS9002 three times when exporting the route to AS X." In this case, our algorithm would correctly infer the action communities but associate them with incorrect controlling ASes.

4 Automatic Identification of Action Communities

In this section, we describe practical uses of BGP communities that violate the three factors described in Section 3.1 and complicate the inference of action communities (§4.1). We then describe how we identify communities that rarely appear with their controlling ASes as action communities (§4.2) and how we use them to uncover other action communities that do not necessarily satisfy our premise of appearing in route announcements without their controlling ASes (§4.3).

4.1 Identifying BGP Community Squatting

We observe that ASes may use BGP information communities defined by or belonging to other ASes. As an AS X is not supposed to tag routes with AS Y's information communities, we refer

to this type of use as *squatting*. A common case is ASes using communities defined by one of their *siblings*, *i.e.,* another ASN under the control of the same organization [8, 17]. This behavior seems particularly common after network mergers and could result from the homogenization of routing policies defined using BGP communities across the merged ASes. For example, we observe routes traversing AS3549 (Global Crossing, acquired by Level3/Lumen [36]) tagged with several communities from AS3356 (Level3/Lumen); routes traversing AS286 (KPN, acquired by GTT [6]) tagged with communities from AS3257 (GTT); routes traversing AS5607 (British Sky Broadcasting, BSB) tagged with communities from AS4589 (Easynet, owned by BSB between 2006–2010 [51]).

As a result, a BGP AS-path traversing a set of ASes $\mathcal{S}$ may include communities belonging to other squatted ASes. This leads to information communities appearing in routes that do not traverse the controlling AS, which violates our intuition that only action communities will appear in routes without their controlling AS.

4.1.1 Inference Algorithm. We propose an algorithm to infer ASes that squat another AS's communities. Our goal is to identify an AS X that systematically tags routes with BGP *information* communities whose first 16-bits is another AS Y. The challenge lies in differentiating between *(i)* an AS X squatting AS Y's information communities from *(ii)* an AS X simply using AS Y's action communities. We do this by assuming that action communities are used selectively for specific, generally short-term, traffic engineering policies. In contrast, information communities are consistently applied after being defined, as routes are automatically tagged when an announcement traverses a router. Thus, we identify an AS X consistently appearing with AS Y's communities as a potential squatter.

We identify squatting AS-pairs using the routes from each RIPE RIS and RouteViews collector separately and then aggregate the inferences. Alternate approaches may be possible given different inference mechanisms; our approach strikes a compromise between obtaining enough routes for inferences, combining routes from all ASes peering with each collector, while trying to capture route properties specific to the view of the Internet's topology captured by that collector [38, 40].

For instance, one collector might be unable to identify that AS X squats the communities of another AS Y because an intermediate AS Z strips the squatted communities tagged by AS X. Another collector may observe routes with AS X's squatted communities if its routes do not traverse AS Z.

Our algorithm uses only publicly available information from RouteViews and RIPE RIS collectors. Consider the following notation:

- $C(y)$ is the set of routes tagged with at least one community from AS Y;
- $\mathcal{R}(x)$ is the set of routes that traverse AS X; and
- $\mathcal{R}(\neg y)$ is the set of routes that do not traverse AS Y.

We check if an AS X is related to another AS Y by computing the following three metrics for each pair of ASes:

Coverage. Among the routes that do not traverse AS Y but are tagged with a community from AS Y, we compute the fraction that traverse AS X. More precisely, we define *coverage* $C(x, y) = |\mathcal{R}(x) \cap \mathcal{R}(\neg y) \cap C(y)| \div |\mathcal{R}(\neg y) \cap C(y)|$. Coverage is high when AS X appears in most of the routes tagged with AS Y communities even though they do not traverse AS Y. This implies that AS X "explains" most of the unexpected observations of AS Y's communities and could be squatting. Coverage is low when there are many routes unexpectedly tagged with AS Y's communities that cannot be attributed to AS X.

Local Prevalence. Among the routes that traverse AS X but do not traverse AS Y, we compute the fraction tagged with a community from AS Y. More precisely, we define *local prevalence*

$P_{\text{local}}(x, y) = |\mathcal{R}(x) \cap \mathcal{R}(\neg y) \cap C(y)| \div |\mathcal{R}(x) \cap \mathcal{R}(\neg y)|$. Local prevalence is high when most routes traversing AS X are tagged with a community from AS Y even when the routes do not traverse AS Y. This implies AS X may be squatting and using AS Y's communities as its own information communities. Prevalence is low when many routes traversing AS X do not have a community from AS Y, which indicates AS X is not systematically squatting AS Y's communities: AS X may be simply using AS Y action communities or another AS on some routes traversing AS X is tagging them with AS Y's communities.

Global Prevalence. Among the routes that traverse AS X, we compute the fraction that do not traverse AS Y but are tagged with a community from AS Y. More precisely, we define *global prevalence* $P_{\text{global}}(x, y) = |\mathcal{R}(x) \cap \mathcal{R}(\neg y) \cap C(y)| \div |\mathcal{R}(x)| \leq P_{\text{local}}(x, y)$. Global prevalence is low when the supporting evidence that an AS is squatting is small compared to the number of routes observed through that AS. For example, AS X may appear on many routes through AS Y, which may not remove action communities from routes it propagates to AS X. Alternatively, AS X may be close to a BGP collector and appear on most collected routes, which may contain AS Y's action communities tagged by other ASes.

To infer if an AS Y is squatted by other ASes, we check if another AS X has coverage $C(x, y) > 0.9$, local prevalence $P_{\text{local}}(x, y) > 0.7$, and global prevalence $P_{\text{global}}(x, y) > 0.3$ (§6.1.1). To avoid inferences with weak support and possibly caused by noise in the BGP dumps, we also require that AS X appears squatting at least two communities from AS Y and that these communities are observed in at least six routes each. We justify these choices in Section 6.1. If multiple ASes are identified as possibly squatting AS Y's communities, we select the one with the largest coverage, largest local prevalence, largest global prevalence, or appearing furthest away from the route collector, in order. The high required coverage of 0.9 allows for at most one squatting relationship with a target AS Y from each BGP collector, but multiple squatting relationships with the same AS Y can be identified across multiple collectors.

When handling squatting relationships, we consider that the inferred relationships are bidirectional and transitive, such that if ASes A and B squat communities from AS C, we consider that ASes A, B, and C are part of one squatting relationship.

4.1.2 Special Cases. Manual inspection of the identified squatting AS-pairs indicates that some pairs are likely caused by typing errors. For example, we observed a community 15**9**85:9999 on paths traversing AS15**8**95, which leads to inferring AS15**9**85 as squatting AS15**8**95. We ignore a squatting relationship between two ASes when their ASNs have five digits and the ASNs have an edit distance of 1. We consider edit operations of substituting one digit for another or reordering two consecutive digits. We ignore all communities involved in these squatting relationships when inferring action communities to avoid errors. We consider only five-digit communities because typos are more likely to occur in longer character sequences [46] and are more challenging for an operator to detect visually. This choice is conservative, as typos in shorter communities may decrease the precision of our algorithm. However, this length is not a fundamental limitation of the approach and can be adjusted if necessary.

We also found some squatting AS-pairs likely caused by an integer overflow when 32-bit ASNs are used with classic 32-bit communities that store ASNs in just 16 bits. For example, we identified many communities from AS303 on routes traversing AS327983, where 303 = `0xffff & 327983`. We ignore all squatting relationships where the squatter ASN's last 16 bits are identical to the squatted ASN, and ignore all such communities when inferring action communities.

We also ignore all squatting AS-pairs involving an IXP ASN, as identified in CAIDA's AS-relationship database [7]. Many IXPs define action communities to control announcement propagation through route servers (*e.g.,* [1]), but IXP route servers do not add their ASN to propagated routes, which may lead to some ASes being identified as squatting the IXP's communities.

4.2 Inferring BGP Action Communities

Our inference algorithm centers around checking how often a community is tagged on a route that does not traverse the controlling AS or any of its squatters, from now on collectively referred to as *controlling ASes*. For the reasons stated in Section 3, enforcing a requirement that a community never appears with its controlling ASes is too restrictive. We design and evaluate different approaches to account for lack of visibility and noise in observed community usage. Algorithm 1 presents pseudocode covering all approaches.

Handling squatting ASes. We use the sets of squatting ASes identified in §4.1 to avoid inferring communities squatted upon as action communities. We compute the squatters for the same collectors used to infer action communities resulting in different AS relations. These relations will be used during the inference of the action communities. Before we execute our algorithm, we rewrite ASNs with squatting relationships when they appear in a route's AS-path or communities. In particular, we rewrite each ASNs with the smallest ASN among its set of squatting ASes (Line 2). This ensures that if a route traverses a squatting AS X and is tagged with a community from a squatted AS Y, then both ASNs will be rewritten with the smallest ASN in their set of squatting ASes. This effectively prevents identifying squatted communities as action communities.

Filtering Low-Visibility Communities. We do not make inferences for communities that have limited visibility in public BGP dumps. We require that a community c is observed by at least two collector peers, and that each collector peer observes the community in at least four routes (counted in N^c_{vps}, Line 7, and verified in $C_{\text{candidates}}$, Line 24). These thresholds are chosen empirically (§6.1); however, we show that inferences are not sensitive to their values as long as they are large enough to remove the long tail of rarely-seen communities from the inference process. This filter removed 11,836 communities from our inferences, representing less than 11% of the communities on BGP dumps. Our algorithm would be able to classify these communities if their use and visibility became more widespread.

Inferring Action Communities. Our algorithm operates on each community independently (Line 5). For each community, our inference relies on computing the fraction of routes tagged with a community from AS Y that do not traverse AS Y. This is done by counting the number of routes with each community c (Line 6) and the number of these routes that do not traverse any of c's controlling ASes (Lines 8–10). Using these variables, we infer as action communities those that are mostly *absent* from routes traversing their controlling ASes (C_{absent}, Line 25). This approach allows some occurrences of the controlling ASes and accommodates errors and unexpected cases, like when an action community is not acted upon, *e.g.,* because it was not set by a customer of the controlling ASes, and remains tagged on the route after traversing the controlling AS.

Handling prepend communities. Action communities that ask an AS Y to prepend itself to the AS-path *will* appear on routes traversing AS Y (prepended multiple times) if AS Y does not remove action communities from announcements. To allow the detection of prepend communities in these scenarios, we count the number of times a community appears on routes with AS-paths that have the community's controlling ASes prepended (Lines 13 and 26). This approach has the negative side-effect of possibly inferring some information communities that often appear on routes prepended with the respective controlling ASes as action communities.

Handling action communities added after the controlling AS. An action community has no use after the controlling AS has taken the requested action. However, an AS may (uselessly) tag a route with an action community *after* it has traversed the controlling AS, which has no impact on the route itself but may happen depending on when the tagging is performed. These behaviors directly impact our inferences as they make action communities more likely to appear on routes traversing controlling ASes and thus harder to differentiate from information communities. To filter this case, we use only uphill AS-paths, *i.e.,* AS-paths composed entirely of customer-to-provider relationships starting from the origin AS (C_{before}, Lines 16–21 and 27). Our intuition is that action communities are often used by customers; and thus a community c tagged on an *uphill* AS-path traversing c's controlling ASes is less likely to have been tagged after the controlling AS and more likely to be an information community.

Handling ASes that do not remove action communities from route announcements. An action community has no use after the controlling AS has taken the requested action. However, the controlling AS is not required to untag the action community from the route. To sidestep the uncertainty added by ASes that do not remove action communities, we apply a relaxation filter allowing the community to appear with its controlling AS in a small fraction F of the announcements in each selected vantage point (Lines 25–27).

Algorithm 1 Inference of Action Communities

1: **Input:** $\mathcal{R} \leftarrow$ set of all routes, each with AS-path P and set of communities C.
2: **Requirement:** AS-paths and communities rewritten with each ASN mapped to the lowest ASN in its set of squatting ASes, if any.

3: **for each** route with rewritten AS-path P and set of communities C **in** $\mathcal{R}$ **do**
4: $\quad C_{\text{global}} \leftarrow C_{\text{global}} \cup C$ *{Track all communities visible in BGP dumps.}*
5: $\quad$ **for each** community c **in** C **do**
6: $\quad\quad N^c_{\text{routes}} \leftarrow N^c_{\text{routes}} + 1$ *{Count routes tagged with community c.}*
7: $\quad\quad N^c_{\text{vps}}[P_0] \leftarrow N^c_{\text{vps}}[P_0] + 1$ *{Count routes exported by BGP collector peer P_0 tagged with community c.}*
8: $\quad\quad$ **if** c's controlling ASes $\notin P$ **then**
9: $\quad\quad\quad N^c_{\text{absent}} \leftarrow N^c_{\text{absent}} + 1$ *{Count routes tagged with community c that do not traverse c's controlling ASes.}*
10: $\quad\quad$ **else if** P is uphill **then**
11: $\quad\quad\quad N^c_{\text{info}} \rightarrow N^c_{\text{info}} + 1$ *{Count routes tagged with community c that traverse c's controlling ASes on uphill path.}*
12: $\quad\quad$ **end if**
13: $\quad\quad$ **if** any of c's controlling ASes is prepended in P **then**
14: $\quad\quad\quad N^c_{\text{prepended}} \leftarrow N^c_{\text{prepended}} + 1$ *{Count routes tagged with community c with its controlling ASes prepended.}*
15: $\quad\quad$ **end if**
16: $\quad\quad$ **if** P is uphill **then**
17: $\quad\quad\quad N^c_{\text{uphill}} \leftarrow N^c_{\text{uphill}} + 1$ *{Count uphill routes tagged with community c.}*
18: $\quad\quad\quad$ **if** c's controlling ASes not in the customer cone of ASes in P **then**
19: $\quad\quad\quad\quad N^c_{\text{before}} \leftarrow N^c_{\text{before}} + 1$ *{Count uphill routes terminating before c's controlling ASes.}*
20: $\quad\quad\quad$ **end if**
21: $\quad\quad$ **end if**
22: $\quad$ **end for**
23: **end for**

24: $C_{\text{candidates}} \leftarrow \{c \mid c \in C_{\text{global}} \wedge |N^c_{\text{vps}}| \geq 3 \wedge \min(\text{values}(N^c_{\text{vps}})) \geq 4\}$
25: $C_{\text{absent}} \leftarrow \{c \mid c \in C_{\text{candidates}} \wedge (N^c_{\text{absent}}/N^c_{\text{routes}}) \geq 1 - F\}$
26: $C_{\text{prepend}} \leftarrow \{c \mid c \in C_{\text{candidates}} \wedge (N^c_{\text{absent}} + N^c_{\text{prepend}})/N^c_{\text{routes}} \geq 1 - F\}$
27: $C_{\text{before}} \leftarrow \{c \mid c \in C_{\text{candidates}} \wedge (N^c_{\text{before}}/N^c_{\text{uphill}}) \geq 1 - F\}$
28: $C_{\text{prefix\_tree}} \leftarrow \text{PrefixTree}(C_{\text{absent}}, C_{\text{candidates}})$ *{All communities that match the prefix-tree leaves tagged as action.}*

29: **Output:** $C_{\text{action\_communities}} \leftarrow C_{\text{prepend}} \cup C_{\text{prefix\_tree}}$

4.3 Uncovering Missing Action Communities

Our inference algorithm requires a minimum number of announcements carrying a community to classify it as an action community with high confidence. However, route collectors do not provide complete coverage of the Internet routes, and some ASes filter all communities before forwarding route announcements, impacting the communities' visibility and our algorithm's recall. To circumvent this limitation, we use the communities we infer with high confidence in Algorithm 1 to build a prefix tree from the decimal digits of the community labels and classify other communities with low visibility or that fall under the special cases we list in Section 4.2.

The rationale behind using a prefix tree is that ASes generally define their communities by numbering communities of the same type sequentially and leaving some space between types to accommodate future expansions of the existing types. By following this pattern, communities of the same type share a common prefix, whose length can vary depending on the number of communities of the same type defined sequentially and the space between the types. We observe that most ASes on the Internet follow this pattern. Some use large blocks of fixed size for each type, while others use smaller blocks of variable sizes. Figure 3 shows a prefix tree for the communities documented by AS3257. A leaf, annotated with A for action and I for information, indicates the type of communities that share the prefix starting at the root up to the leaf. For example, labels 3257:02XXX and 3257:1XXXX represent action communities, while 3257:08XXX and 3257:3XXXX represent information communities.

Specifically, we build a prefix tree for each AS that Algorithm 1 infers at least one action community. We treat the label of a community as a string with five digits (*i.e.,* the maximum number of decimal digits a 16-bit label can represent), filling in the string with zeros on the left when the label has fewer than five digits. Then, we divide the communities into sets containing communities with the longest common prefixes. We build one branch of the prefix tree for each set using only the digits in the longest common prefix of the communities in the set. As Algorithm 1 infers only action communities, all the leaves of the prefix trees are labeled with A. We apply the AS's prefix tree to all its communities that appear in $\mathcal{R}$, *i.e.,* the set of all routes from the BGP collectors we process, and classify the communities that share a prefix with a leaf as action communities.

We validated this idea using the communities of 15 ASes in our ground-truth dataset that have at least 20 communities. Specifically, we conducted experiments by building a prefix tree with a random subset of communities from an AS's ground truth and testing with the remaining communities from the same AS. We varied the subset sizes from 20% to 90% of the total communities and ran 100 experiments for each subset size. We measured the average precision and recall, with the average precision exceeding 99.5% for all subset sizes and the average recall ranging from 90.9% to 96.82%. These results indicate that the prefix trees effectively capture the structure of the community definitions of the selected ASes.

5 Datasets

We evaluate our algorithms using the first BGP routing table (RIB) dumps of Dec. 1st, 2023, from all 55 BGP route collectors operated by RIPE RIS [44] and RouteViews [35]. We use bgpscanner [41] to process the RIB dumps and remove routes with AS-level loops (0.005% of routes) or AS-sets [29] (0.03%). For each route, we extract the prefix, the AS path, and the possibly-empty set of attached BGP communities.

We use CAIDA's AS-relationship database [20, 32] from Dec. 1st, 2023, to identify the *uphill*, *peak*, and *downhill* regions of the AS-path. We ignore 0.27% of routes that violate valley-free routing and attempt to infer relationships for AS-pairs in a route missing from CAIDA's database. If the

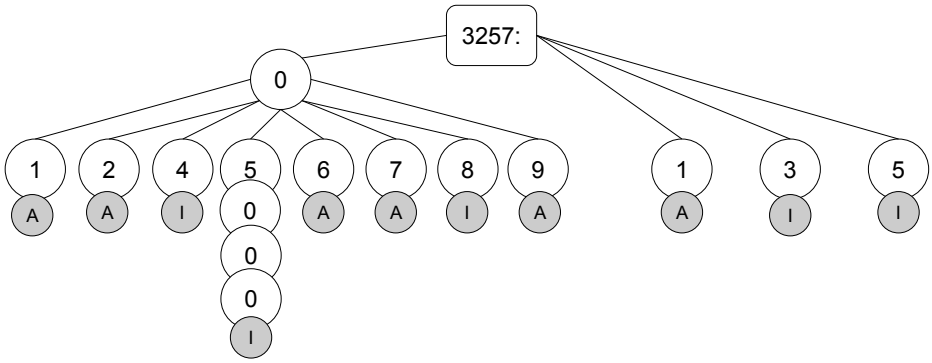


Fig. 3. A prefix tree for the documented BGP communities from AS 3257. The branch 05000 is unusually long because it contains only one community, with no other communities sharing the 05* prefix.

existing relationships are compatible with valley-free routing and at most one relationship is missing at the peak, we infer missing relationships as customer-to-provider in the uphill region, provider-to-customer in the downhill region, and peer-to-peer if there is a missing relationship at the peak.[3] We perform this inference of missing relationships for each route separately; inferences from one path do not carry over to other routes.

We parse public information from Internet Routing Registry (IRR), NL NOG [22], and OneStep [48] databases to extract ground-truth information to classify BGP communities according to their semantics. We use this ground-truth dataset to evaluate the precision and recall of our inference algorithm. Our database includes information about the type of AS (*i.e.,* Tier-1, Tier-2, and others) of each community to evaluate how the performance metrics vary as a function of where the AS is on the Internet hierarchy. It contains 16,421 action communities from 74 ASes: 14322, 532, and 1567 from Tier-1, Tier-2, and other ASes, respectively. Although our ground-truth dataset contains a little over 1% of 6,158 ASes appearing on BGP communities in public BGP dumps, the ASes we consider are large and make more significant use of BGP communities than the average AS on the Internet. Overall, the ASes in our ground-truth dataset account for 16.8% of visible BGP communities in Dec. 2023. Also, our ground-truth dataset covers a variety of action communities, including selective advertisements, blackholing, prepending, and changing the LocalPref; with several ASes defining action communities that apply to specific peers or geographical locations.

To build the ground-truth dataset of ASes that squat the communities of other ASes, we also use public information about organizations, their ASNs, and their prefixes from the IRR databases. We use these databases to map ASNs to their controlling organizations and determine if two ASes are related by manually looking for similarities in organization names, geographical addresses, descriptions, and domain names for peering, operations, and abuse e-mail addresses [2, 3, 47]. To add relationships to the ground-truth dataset, we initially generated a set with the relationships that our algorithm for identifying squatting inferred with very restrictive parameters—*i.e.,* coverage = 0.9, local prevalence = 0.9, and global prevalence = 0.9—and manually classified the inferred relationships. We then gradually reduced coverage and local and global prevalence from 0.9 to 0.1 to increase the number of classified relationships until we could not validate the new ones.

[3]This approach is equivalent to reapplying steps 5 and 11 of the original algorithm [32], but visiting ASes in the route from the peak toward the origin and from the peak toward the collector instead of following the transit and node degree gradients.

We classified 59 relationships as confirmed and 23 as unconfirmed. To confirm a relationship, we used the similarities described previously. We consider a relationship unconfirmed if we find the documentation about the two ASes and it does not have any similar information that leads us to believe they are related. Note that this approach is conservative, as the ASes may be related even though the documentation does not reflect their relationship either by lacking the information or by being outdated.

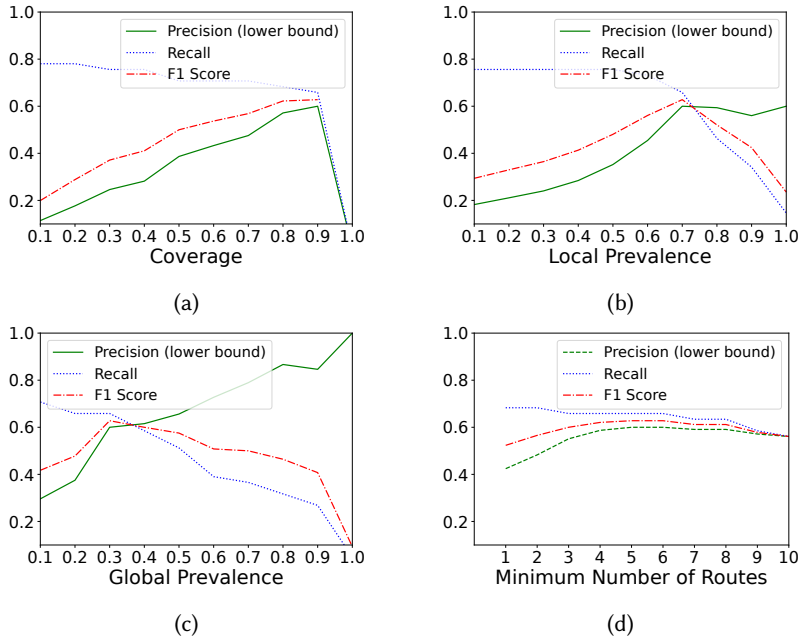


Fig. 4. Each of the graphs (a)-(c) shows the behavior of one of the parameters of our algorithm when we keep the other two at their default (best) values. Increasing threshold values improves precision at the cost of recall, as expected, and the default values represent the inflection points of the F1-score curves. Graph (d) shows the impact of the minimum number of routes communities must appear to determine a squatting relationship.

6 Evaluation

This section describes how we configure the parameters of our inference algorithms, evaluates the precision and accuracy of our inferences, and compares them with related prior work. We show that our algorithms are not strongly dependent on specific parameter configurations, *i.e.,* a broad range of configurations yields positive results. We make our datasets and evaluation code public to ease the replication of our results and independent executions of the inference algorithms [12].

6.1 Setting Parameters

6.1.1 Configuration of the Squatting Inference Algorithm. As described in Section 4.1, our algorithm for identifying squatters relies on three parameters: *coverage*, *local prevalence*, and *global prevalence*. These parameters are fractions in the interval [0, 1] computed over sets of routes. To determine the best parameters and investigate if they generalize to other datasets, we use route announcements from December 2022 to explore different combinations of the parameters. Specifically, we vary coverage, local prevalence, and global prevalence in the interval [0.1, 1] in steps of 0.1, resulting in

1000 (*i.e.,* 10^3) combinations. We validate the inferred squatting relationships by computing the precision and recall for each parameter combination using the ground-truth dataset (Section 5).

The combination of coverage = 0.9, local prevalence = 0.7, and global prevalence = 0.3 yields the highest F1 score, so we choose it as the default configuration.[4] The best configuration achieves a lower bound on precision of 0.71 and a recall of 0.65. We note that 0.71 is a lower bound on precision because some of the inferred squatting relationships may be missing from our ground truth dataset (*i.e.,* we have not manually checked a pair of ASes); we take a conservative approach and report these inferences as incorrect, but some could be correct.

Figures 4(a)-(c) show the precision, recall, and F1 score when we vary one parameter and keep the other two parameters fixed at their default (best) values. As expected, increasing threshold values improves precision at the cost of recall, and the selected values represent inflection points of the F1 score. We also observe that every parameter impacts the inferred relationships. Our algorithm infers no squatters when coverage = 1; thus, both precision and recall are zero. Figure 4(d) shows the results when we fix the minimum number of routes communities must appear to determine a squatting relationship. The selected value six represents the inflection point of the F1-score curve.

Our algorithm inferred 54 pairs of squatting relationships, with 7 ASes appearing in multiple pairs, which we join for a final count of 48 (transitive) relationships. Of these relationships, the validated inferences include 26 sibling ASes, 2 neighboring ASes, 19 missing from our ground truth, and 7 unconfirmed.

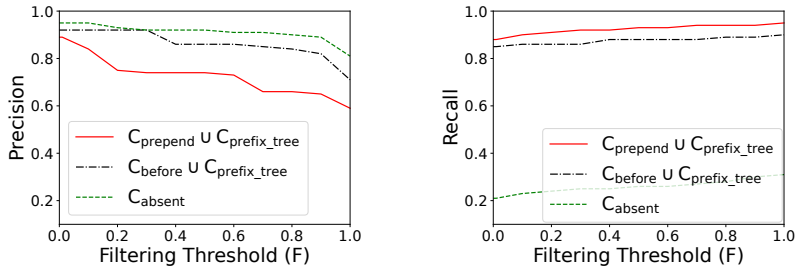
We believe our automated inference of ASes squatting BGP communities might have applications for other studies relying on BGP communities (*e.g.,* validation of AS-relationship inference [20, 25, 32] and route change tracking [13, 19]). It might also benefit other efforts that seek to identify relationships between ASes. For example, the intersection of our community-based inference of squatters and our ground truth dataset contains five sibling relationships not identified by Chen *et al.*'s recent technique [8]. Finally, it is unclear why apparently unrelated ASes squat another's communities in some cases. We note that this practice, even if well-intended, may confuse troubleshooting efforts and policy filters not only for the ASes involved but also their neighbors [49].

6.1.2 Configuration of the Action Communities Inference Algorithm. Section 3 discussed the main challenges in inferring action communities. We evaluated the parameters of Algorithm 1 to mitigate those issues.

We evaluate precision and recall for different community filtering thresholds (F in Algorithm 1). Figure 5a shows precision as we vary the filtering threshold on the x-axis, while Figure 5b shows the recall for the same configurations. We compare the more conservative C_{before} vs. the more inclusive C_{prepend}. As expected, considering only uphill paths leads to higher precision overall, as we avoid the case of ASes that uselessly tag their provider p's action communities on an AS-path that has already traversed p; the drawback is lower recall as less information is available for inferences. The figures also show the results when using $C_{\text{prepend}} \bigcup C_{\text{prefix\_tree}}$; overall, we find that the prefix tree nearly doubles the recall, at the cost of some loss of precision.

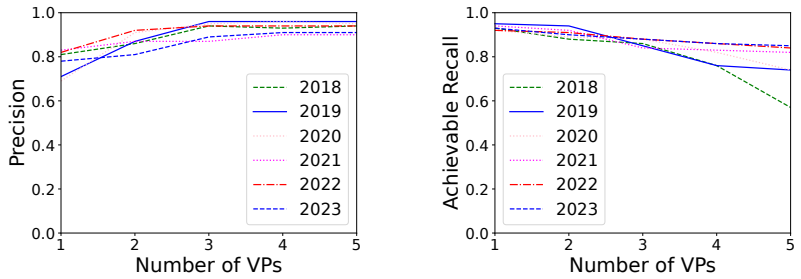
Figure 5 also shows that setting F to zero is too conservative. With this configuration, our algorithm infers few information communities as action communities, achieving very low recall. Very low thresholds perform best, as they allow for some noise (*i.e.,* action communities appearing

[4]We evaluated the Phi coefficient (also known as the Matthews Correlation Coefficient, MCC) [11, 33], and found that it is strictly higher than the F1-score, quantitatively similar to recall, and has no inflection point to aid in choosing default values for each parameter (not shown). While the Phi coefficient considers imbalance between classes, it is less suitable for our evaluation because the number of true negatives—AS pairs that have no squatting relationship—is exceedingly large [9, 10]. Our use of the F1-score focuses on the worse-performing minority class (the positive inferences) and is thus a more relevant, *conservative* result.



(a) Precision (b) Recall

Fig. 5. Inference performance as a function of the noise filter threshold F. Higher F values allow a BGP community to appear on more routes with its controlling AS and still be inferred as an action community.



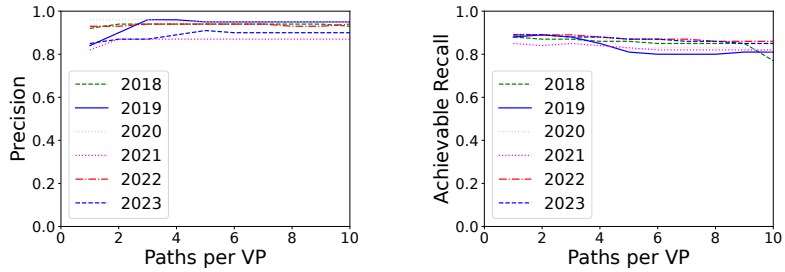
(a) Precision (b) Recall

Fig. 6. Impact of varying the minimum number of Vantage Points (VPs) observing a community in Algorithm 1 (N^c_{vps}). We analyze the precision and recall from 2018 to 2023, showing stable performance for all datasets.

with their ASes) and significantly improve recall without sacrificing precision. After this initial filtering (increasing F from zero to, *e.g.,* 0.01), the performance of our algorithm is stable across all threshold values. Considering this finding, in the rest of this paper we set $F = 0.01$.

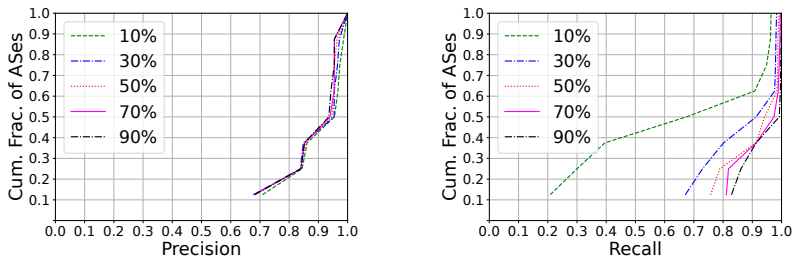
We also require a minimum visibility of a BGP community at vantage points (VPs) to make inferences. If we increase the number of VPs where a community must be observed, the precision increases but recall decreases as we make fewer inferences. Figure 6 shows the precision and recall achievable when we compare the inference of action communities using the first RIB of December from 2018 to 2023, varying the number of vantage points (VPs). We note that the algorithm's performance as a function of configuration parameters is consistent, meaning that the algorithm's configuration does not need to be reevaluated often. Considering the inflection points in the graphs, we choose 3 VPs as the minimum for action community inference as a good trade-off between precision and recall. Different applications can increase the number of VPs if they benefit from higher precision, or decrease to favor recall.

Finally, a VP observing very few routes with a community could lead to incorrect inferences. Therefore, we also evaluate how many routes with a particular community a VP must have before we consider that (VP, community) in our inference. Figure 7 shows the impact of the minimum number of routes required when we fix $F = 0.01$ and the minimum number of VPs at 3, for every month of December between 2018 and 2023. Again, we observe that performance is stable throughout the period. We also find that the minimum number of required routes has limited impact, but



(a) (b)

Fig. 7. Performance as a function of the minimum number of required routes per (VP, community) pair before making inferences. Results are stable across the evaluation period. We conservatively chose a minimum of 4 routes per VP.



(a) Precision (b) Recall

Fig. 8. Cumulative distributions of precision and recall for inferences made by the prefix trees built from a random subset of inferred action communities. Different lines vary the fraction of inferences used as input to build the prefix tree and show that the prefix tree does not require many inferences to achieve high precision and recall.

that setting it too low may hurt precision. We take a conservative approach and set the minimum number of routes to 4 in the rest of the paper, which the graphs indicate should work in general.

6.1.3 Building the Prefix-Tree. Section 4.3 proposed using a prefix tree for classifying communities. We evaluate how practical this approach is by evaluating how many communities are needed to build a prefix tree that achieves high precision and recall.

Figure 8a shows the distributions of precision and recall for 8 ASes with at least 20 communities in our ground-truth. We built prefix trees using action communities inferred with C_{absent}, which avoids the loss of accuracy incurred by C_{prepend}. Each point in the distribution represents the average of 100 executions with random subsets of the communities in C_{absent}. The different lines vary the fraction of inferred communities used to build the tree. We report precision and recall obtained when classifying the communities in our ground-truth dataset using the prefix tree.

We can see that prefix trees for most ASes achieve very high precision even when we build trees with as few as 10% of an AS's inferred action communities. Consequently, we need to infer only a small number of action communities for the prefix tree to be effective. Figure 8b shows that the recall is also high, increasing from an average of 0.66 when using 10% of the inferred communities to 0.95 when using 90%. For three of the 8 ASes, the recall is smaller than 0.4 for samples with 10% of the inferred communities, but it increases significantly for samples with 30% or more.

Table 1. Evaluation for ASes Tier-1 and Tier-2 on the BGP dumps from December 2023. The table shows the number of inferred communities (Num), precision (Prec), recall (Rec), and the number of inferred communities not in our ground-truth dataset (Unk) for three configurations of our algorithm. The last three columns (GT ∩ BGP) classify all communities appearing in the BGP dumps using our ground-truth dataset. The line Total at the bottom shows weighted averages of precision and recall.

	Inf. without Prepend C_{absent}				**Inf. with Prepend** C_{prepend}				**Inf. Prepend with Tree** $C_{\text{prepend}} \cup C_{\text{tree}}$				**GT ∩ BGP**		
ASN	Num	Prec	Rec	Unk	Num	Prec	Rec	Unk	Num	Prec	Rec	Unk	Act	Info	Unk
1299	131	0.98	0.43	36	131	0.98	0.43	36	340	0.84	1.0	80	218	98	138
174	75	1.0	0.97	47	75	1.0	0.97	47	82	1.0	1.0	47	29	4	118
1764	2	0	0	2	13	1.0	0.18	6	13	1.0	0.18	6	38	38	16
2914	61	1.0	0.93	22	61	1.0	0.93	22	67	0.95	1.0	23	42	81	30
3257	36	0.88	0.38	19	36	0.88	0.38	19	61	0.87	0.85	23	39	844	26
3292	14	1.0	0.53	5	18	1.0	0.76	5	21	1.0	0.88	6	17	10	40
3356	27	0.75	0.6	23	37	0.5	0.6	31	239	0.38	0.6	231	5	144	331
33891	4	1.0	0.02	3	5	1.0	0.03	3	5	1.0	0.03	3	63	24	236
3491	62	0.94	0.25	12	69	0.95	0.28	13	252	0.94	0.99	50	16	139	60
3549	34	1.0	0.57	22	34	1.0	0.57	22	49	1.0	0.86	31	21	22	109
4589	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0
5400	2	0	0	0	2	0	0	0	2	0	0	0	1	0	68
5511	18	0.86	0.35	4	18	0.86	0.35	4	39	0.94	0.88	7	35	51	253
6461	43	1.0	0.64	4	43	1.0	0.64	4	63	0.95	0.92	4	61	289	54
6663	2	1.0	1.0	1	6	1.0	1.0	5	6	1.0	1.0	5	1	0	23
6762	65	1.0	0.13	54	69	0.93	0.16	54	196	0.68	1.0	70	86	46	79
701	19	1.0	1.0	12	19	1.0	1.0	12	20	1.0	1.0	13	7	0	16
7922	5	1.0	0.83	0	5	1.0	0.83	0	5	1.0	0.83	0	6	0	35
Total	600	0.97	0.37	266	641	0.96	0.40	283	1460	0.87	0.86	599	862	1794	1632

Table 2. Longitudinal Evaluation of Inferred Action Communities by Semantics.

	2018			2019			2020			2021			2022			2023		
Semantics	$\lvert C\rvert$	BGP	Frac	$\lvert C\rvert$	BGP	Frac	$\lvert C\rvert$	BGP	Frac	$\lvert C\rvert$	BGP	Frac	$\lvert C\rvert$	BGP	Frac	$\lvert C\rvert$	BGP	Frac
Local Preference	36	48	0.75	40	55	0.72	43	57	0.75	52	59	0.88	59	66	0.89	78	82	0.95
No Advertise/Export	145	165	0.88	168	169	0.99	179	183	0.98	187	226	0.83	221	222	0.99	207	234	0.88
Prepend (1x, 2x, 3x)	368	414	0.89	389	443	0.88	402	433	0.93	413	438	0.94	436	474	0.92	461	505	0.91
Blackhole	8	12	0.67	8	10	0.8	9	13	0.69	11	14	0.79	7	10	0.7	7	10	0.7

6.2 Inference Accuracy

Table 1 shows the number of communities, the *precision*, and *recall* for every Tier-1 and Tier-2 AS in our ground-truth dataset. The unknown columns (Unk) show the number of inferred action communities that are not in our ground-truth dataset. We color values larger than 0.8 green and values between 0.5 and 0.8 orange. We show three configurations of our algorithm: the baseline inferences (C_{absent}), the inferences considering prepended paths (C_{prepend}), and the inferences considering prepended paths and the prefix tree ($C_{\text{prepend}} \cup C_{\text{tree}}$). As expected, relaxing the algorithm improves recall at the cost of precision. However, considering the small reduction in precision and large improvements in recall, we recommend the use of the inferences considering the prepended paths and the prefix tree. Applications where precision is paramount, however, can still opt for the more conservative configuration for the highest precision.

The last column (GT ∩ BGP) classifies the communities observed in the BGP dumps into action communities, information communities, or unknown depending on their type in our ground-truth dataset. This column shows that our algorithm achieves high precision and recall for the majority of ASes whose communities have a significant presence in the BGP dumps. Our algorithm makes few inferences for ASes 4589 and 5400, which make limited use of BGP communities.

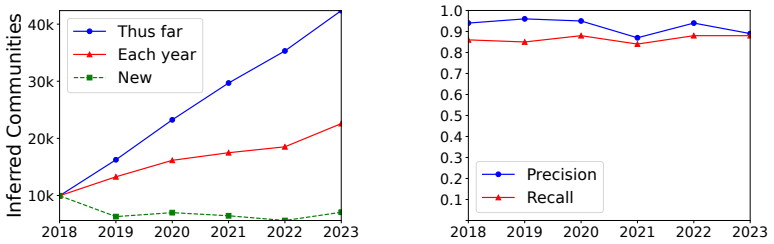


Fig. 9. Results of our algorithms over six years (2018-2023): (a) the number of distinct communities found each year and (b) the precision and recall achieved. The data is based on the first BGP RIB collected in December from all RIPE and RouteViews collectors.

Table 3. Example of AS1764's grouping of BGP communities by neighbor.

Community	Meaning	Category
1764:40020	Received via AS174	Information
1764:40021	Prepend (1x) to AS174	Action
1764:40022	Prepend (2x) to AS174	Action
1764:40023	Prepend (3x) to AS174	Action

We carried out a longitudinal evaluation considering the first RIB of the month of December between 2018 and 2023. Figure 9 shows the results. On average, the precision is 92.5% (standard deviation of 3.62%) and the recall is 86.5% (standard deviation of 1.76%). Table 2 shows the type of correctly-inferred action communities across each year and demonstrates balanced semantic coverage across all action community classes. We classified the semantics of all but 1,617 action communities in our ground-truth dataset into the four classes in Table 2. For these communities we have no information to classify them, *e.g.,* 51 communities from AS5511 are labeled simply "tune" in the WHOIS documentation. Although we consider these action communities and correctly infer them as such, we do not include them in the table.

6.3 Clustering vs. Prefix Tree

Krenc *et al.*recently presented a mechanism for classifying BGP communities as action versus information [28]. They classify as action communities any community that often appears on AS-paths that do not traverse the controlling AS.[5] The technique then clusters communities with integer values less than 140 apart and applies a majority vote across all communities in a cluster to determine their type. It reclassifies the communities in the minority group to match the type of the majority. The paper evaluates the mechanism using ground truth from the NLNog database [22] and communities classified based on their descriptions using regular expressions.

We compared the inferences from our algorithm with the results available in their paper for the period they considered (May 1–7, 2023). We consider their original and our extended ground-truth datasets. On their ground-truth dataset, the prior work achieves an F1 score of 0.95 for the action communities, while our technique achieves 0.94. On our extended ground-truth dataset, the prior work achieves an F1 score of 0.92, while our technique also achieves 0.92.

However, there is a significant difference in performance when we consider only communities not in the original ground-truth dataset. Our algorithm achieves an F1 score of 0.80 vs. 0.50 for the previous technique, and Phi coefficient of 0.78 vs. 0.51. Table 4 shows a detailed evaluation of the Tier-1 and Tier-2 ASes present in our extended ground-truth but missing from the ground-truth

[5]The specific threshold they use is 99.37% (a ratio of 160:1), which maximizes the F1 score for their ground-truth.

Table 4. Comparison of cluster and prefix tree inferences for May 2023 by ASN that was not in the GT of the cluster inference algorithm. We compute the inference of the first RIB from all available collectors for the same interval used in [28] (who used all RIBs and updates). To be able to fully compare the communities captured across all collectors, we relaxed our algorithm to restrict to just one announcement per VP, while maintaining all other algorithm parameters. This way, both algorithms have the same visibility of all RIB communities. We cannot compute the Phi coefficient [11, 33] when there are no inferences (positives) or when an AS has no documented information communities (true negatives).

	Prefix Tree					**GT vs BGP 7 days (First RIB)**			**Cluster**					**GT vs BGP 7 days (all RIB/Up)**		
ASN	Infer	Prec	Rec	F1 Sc.	Phi	Act	Info	Unk	Infer	Prec	Rec	F1 Sc.	Phi	Act	Info	Unk
701	16	1.0	0.71	0.83	—	7	0	17	24	1.0	1.0	1.0	—	9	0	43
703	3	1.0	1.0	1.0	—	2	0	1	3	1.0	1.0	1.0	—	2	0	1
1764	22	1.0	0.41	0.58	0.50	41	36	14	0	0	0	0	—	41	40	14
3257	50	0.88	0.79	0.83	0.83	38	860	25	30	0.6	0.23	0.33	0.36	39	911	26
3549	55	1.0	0.91	0.95	0.91	22	22	129	63	1.0	0.3	0.47	0.42	23	22	144
4589	0	0	0	0	—	4	4	0	8	0.5	1.0	0.67	—	4	4	0
5400	1	0	0	0	—	2	0	62	3	1.0	1.0	1.0	—	2	0	62
5511	38	0.94	0.86	0.9	0.84	35	51	32	39	1.0	0.72	0.84	0.78	36	52	32
6663	12	1.0	1.0	1.0	—	1	0	15	7	1.0	1.0	1.0	—	1	0	17
7922	3	1.0	0.43	0.6	—	7	0	39	34	1.0	1.0	1.0	—	7	0	39
33891	55	1.0	0.7	0.82	0.63	63	24	245	35	1.0	0.19	0.32	0.25	64	24	247
Total	255	**0.96**	**0.68**	**0.80**	**0.78**	222	997	579	246	**0.89**	**0.35**	**0.50**	**0.51**	228	1053	625

dataset used by the prior work. These differences may be explained by the more recent publication of the communities of the ASes in the subset, which affects their visibility as fewer networks use them. Also, some of these ASes assign community numbers in ways that violate the assumptions of the prior approach. The prior approach performs poorly for AS1764 (NextLayer) because AS1764 intermixes action and information communities when assigning community numbers, violating the assumption that ASes allocate communities in contiguous blocks. Table 3 shows how AS1764 groups information and action communities by neighbor, which leads to systematic errors when the majority vote is applied to communities in each cluster. This behavior is not exclusive to AS1764; *e.g.,* AS33823 also groups communities by neighbor. This practice reduces the recall of our prefix tree, but there is no impact on precision as we do not overwrite inferences. AS3549 and AS33891 define both action and information communities in intervals smaller than 140, leading to low performance for the prior approach. Our approach performs better for these ASes, as the prefix tree can dynamically adjust group sizes.

Finally, the prior approach's worse performance for AS3257 (GTT) results from it not handling ASes squatting GTT's communities. We find that AS286 (previously KPN, acquired by GTT) and AS29140 (HostServer, unclear relationship to GTT) both squat GTT's communities, leading the previous work to incorrectly infer some of GTT's information communities as action communities because they appear on routes without AS3257 (but with AS286 or AS29140). The similar F1 scores for the extended ground-truth dataset indicate that both techniques have similar overall performance, but our technique is more resilient to ASes with unknown operational practices or BGP community squatting.

7 Related Work

AS Relationships. Characterizing the relationships between ASes is challenging, whether due to constant changes on the Internet or the lack of reliable public information that does not expose certain relationships, such as backup connections or regional connections that are not visible at route collectors [39]. Also, ASes may have hybrid relationships [20, 25] that vary depending on where they peer.

Determining the relationship between two ASes, however, has many practical applications [32], such as detecting if a customer AS is exporting routes from one of its providers to a peer or another provider, which leads to route leaks and disrupts the traffic on parts of the Internet [49, 53]. Recent research aims to infer sibling relationships using multiple data sources provided by network operators [3, 8]. Our previous work [47], in turn, presents a heuristic for detecting the existence of sibling ASes on a set of route announcements from BGP route collectors. The approach, however, detects the relationship's existence without identifying the ASes involved. Here we also use data from route collectors, but go beyond detection and identifies the ASes squatting the communities of other ASes, which can indicate sibling ASes or some other agreed-upon relationship between the ASes. By using public data from the route collectors, our approach can uncover undocumented relationships that elude the current approaches relying on public documentation [3].

BGP Communities. The use of BGP communities has increased significantly in recent years [49], whether to improve network quality, prevent DDoS attacks [21], or detect infrastructure failures [18] or instability in route announcements [16]. However, the lack of standardization and documentation for the semantics of BGP communities makes it difficult for researchers and network operators to reason about Internet routing dynamics or use them to implement more complex routing policies.

Several recent research efforts propose different techniques for building dictionaries of BGP communities [18, 21, 28, 47] or inferring which ASes tag routes with communities [27]. Giotsas *et al.* [18, 21] use natural language processing to extract the semantics of BGP communities from IIRs, web pages, and public documents from the ASes. Krenc *et al.* [28] propose a clustering algorithm for classifying information and action communities that depends on a ground-truth database to define the parameters that separate the two types of clusters. The paper shows high precision for the algorithm. However, the evaluation uses the same communities that were used to define the parameters of the algorithm, *i.e.,* it doesn't split the communities into training and test datasets to determine if the parameters generalize to the test dataset. As we show in Section 6, their approach may not generalize to other ASes, resulting in lower precision and recall for the action communities than the ones reported in [28]. The approaches of Giotsas *et al.* [18, 21] and Krenc *et al.* [28] depend on the availability of documentation from the ASes, which is sometimes incomplete, outdated, or nonexistent. In our work, we use the existing documentation only to build the ground-truth database and evaluate the results of our inference algorithms. Our previous work [47] uses public information from route collectors to infer location communities, which are information communities that tell where a route is learned. This approach is complementary to our focus here on inferring action communities and identifying squatters.

8 Conclusion

In this paper, we design and evaluate an algorithm for automatically identifying BGP action communities that relies only on route announcements observed by BGP route collectors. We also present an algorithm for uncovering ASes that consistently use (*i.e.,* squat) other ASes' communities, revealing undocumented relationships and shedding light on the complex interactions between networks on the Internet. These relationships help, for instance, filter out information communities that would otherwise be identified as action communities. Our evaluation results show that our algorithm for identifying action communities achieves average precision and recall of 92.5% and 86.5%, respectively, in a longitudinal study with BGP data from 2018 to 2023.

Our work uses only public data from BGP collectors and raises no ethical concerns. We employ a non-invasive approach that does not disrupt Internet announcements, and all data processing is performed offline.

Acknowledgments

We thank our shepherd, Esteban Carisimo, and the anonymous ACM SIGMETRICS reviewers for their valuable feedback. This work was partially funded by CNPq procs. 420934/2023-5, 308101/2022-7, and 307061/2021-3; FAPESP procs. 2023/00812-7 and 2023/00811-0, FAPEMIG procs. APQ-02793-23 and APQ-02856-18; CAPES Finance Code 01; and NSF Award 2344761.

References

[1] AMX-IX. 2023. AMS-IX Route Servers.

[2] 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*. ACM, Tokyo, Japan, 71–77. https://doi.org/10.1145/2815675.2815712

[3] Augusto Arturi, Esteban Carisimo, and Fabián E Bustamante. 2023. as2org+: Enriching AS-to-Organization Mappings with PeeringDB. In *International Conference on Passive and Active Network Measurement*. Springer, Springer, Springer Nature Switzerland, 400–428.

[4] Henry Birge-Lee, Maria Apostolaki, and Jennifer Rexford. 2024. Global BGP Attacks that Evade Route Monitoring. arXiv:2408.09622 [cs.CR] https://arxiv.org/abs/2408.09622

[5] Henry Birge-Lee, Liang Wang, Jennifer Rexford, and Prateek Mittal. 2019. SICO: Surgical Interception Attacks by Manipulating BGP Communities. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*. ACM Press, London, United Kingdom, 431–448. https://doi.org/10.1145/3319535.3363197

[6] Business Wire. 2019. GTT Completes Acquisition of KPN International. https://www.businesswire.com/news/home/20191202005214/en/GTT-Completes-Acquisition-of-KPN-International.

[7] CAIDA. 2022. The CAIDA AS Relationships Dataset, 2022-12-01. https://www.caida.org/catalog/datasets/as-relationships/.

[8] Zhiyi Chen, Zachary S Bischof, Cecilia Testart, and Alberto Dainotti. 2023. Improving the Inference of Sibling Autonomous Systems. In *International Conference on Passive and Active Network Measurement*. Springer, Springer, Springer Nature Switzerland, 345–372.

[9] Davide Chicco and Giuseppe Jurman. 2020. The Advantages of the Matthews Correlation Coefficient (MCC) over F1 Score and Accuracy in Binary Classification Evaluation. *BMC genomics* 21 (2020), 1–13.

[10] Davide Chicco, Niklas Tötsch, and Giuseppe Jurman. 2021. The Matthews Correlation Coefficient (MCC) is More Reliable than Balanced Accuracy, Bookmaker Informedness, and Markedness in Two-Class Confusion Matrix Evaluation. *BioData mining* 14 (2021), 1–22.

[11] Harald Cramér. 1999. *Mathematical Methods of Statistics*. Vol. 26. Princeton University Press, 41 William St, Princeton, NJ 08540.

[12] Brivaldo A. da Silva Jr, Adriano B. de Carvalho, Ítalo Cunha, Timur Friedman, Ethan Katz-Bassett, and Ronaldo A. Ferreira. 2024. BGP Action Communities – Supplemental Material. https://github.com/TopoMapping/bgp-action-communities

[13] Benoit Donnet. 2009. Incentives for BGP Guided IP-Level Topology Discovery. In *International Workshop on Traffic Monitoring and Analysis*. Springer, Springer, Springer Berlin Heidelberg, 101–108.

[14] Benoit Donnet and Olivier Bonaventure. 2008. On BGP Communities. *ACM SIGCOMM Computer Communication Review* 38, 2 (2008), 55–59.

[15] Jerome Durand, Ivan Pepelnjak, and Gert Döring. 2015. *BGP7454: BGP Operations and Security*. Technical Report. RFC 7454, February.

[16] Anja Feldmann, Olaf Maennel, Z. Morley Mao, Arthur Berger, and Bruce Maggs. 2004. Locating Internet Routing Instabilities. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (Portland, Oregon, USA) *(SIGCOMM '04)*. ACM, New York, NY, USA, 205–218. https://doi.org/10.1145/1015467.1015491

[17] Lixin Gao. 2001. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions on Networking* 9, 6 (2001), 733–745. https://doi.org/10.1109/90.974527

[18] Vasileios Giotsas, Christoph Dietzel, Georgios Smaragdakis, Anja Feldmann, Arthur Berger, and Emile Aben. 2017. Detecting Peering Infrastructure Outages in the Wild. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, New York, NY, USA, 446–459. https://doi.org/10.1145/3098822.3098855 event-place: Los Angeles, CA, USA.

[19] 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* (Virtual Event, USA) *(IMC '20)*. ACM Press, New York, NY, USA, 247–265.

https://doi.org/10.1145/3419394.3423654
[20] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, and kc claffy. 2014. Inferring Complex AS Relationships. In *Proceedings of the ACM Internet Measurement Conference* (Vancouver, BC, Canada) *(IMC '14)*. ACM Press, New York, NY, USA, 23–30. https://doi.org/10.1145/2663716.2663743
[21] Vasileios Giotsas, Philipp Richter, Georgios Smaragdakis, Anja Feldmann, Christoph Dietzel, and Arthur Berger. 2017. Inferring BGP blackholing activity in the internet. In *Proceedings of the ACM Internet Measurement Conference*. ACM Press, London, United Kingdom, 1–14. https://doi.org/10.1145/3131365.3131379
[22] Netherlands Network Operator Group. 2024. NLNOG Looking Glass - Known communities. https://github.com/NLNOG/lg.ring.nlnog.net/tree/main/communities
[23] J Heitz, J Snijders, K Patel, I Bagdonas, and N Hilliard. 2017. RFC8092: BGP Large Communities Attribute. https://www.rfc-editor.org/rfc/rfc8092.txt
[24] CISCO Inc. 2021. CISCO Route-Map. https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/49111-route-map-bestp.html [Online; accessed 11-Oct-2021].
[25] Yuchen Jin, Colin Scott, Amogh Dhamdhere, Vasileios Giotsas, Arvind Krishnamurthy, and Scott Shenker. 2019. Stable and Practical AS Relationship Inference with ProbLink. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. USENIX Association, Boston, MA, 581–598. https://www.usenix.org/conference/nsdi19/presentation/jin
[26] Thomas Krenc, Robert Beverly, and Georgios Smaragdakis. 2020. Keep Your Communities Clean: Exploring the Routing Message Impact of BGP Communities. In *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies* (Barcelona, Spain). ACM Press, New York, NY, USA, 443–450. https://doi.org/10.1145/3386367.3432731
[27] Thomas Krenc, Robert Beverly, and Georgios Smaragdakis. 2021. AS-Level BGP Community Usage Classification. In *Proceedings of the 21st ACM Internet Measurement Conference* (Virtual Event) *(IMC '21)*. Association for Computing Machinery, New York, NY, USA, 577–592. https://doi.org/10.1145/3487552.3487865
[28] Thomas Krenc, Matthew Luckie, Alexander Marder, and kc claffy. 2023. Coarse-grained Inference of BGP Community Intent. In *Proceedings of the 2023 ACM on Internet Measurement Conference*. ACM, Montréal, Canada, 66–72.
[29] Warren Kumari and Kotikalapudi Sriram. 2011. RFC 6472-Recommendation for Not Using AS_SET and AS_CONFED_SET in BGP.
[30] Tony Li, Ravi Chandra, and Paul S. Traina. 1996. BGP Communities Attribute. RFC 1997. https://doi.org/10.17487/RFC1997
[31] Zhihao Li, Dave Levin, Neil Spring, and Bobby Bhattacharjee. 2018. Internet Anycast: Performance, Problems, & Potential. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '18)*. ACM Press, Budapest, Hungary, 59–73. https://doi.org/10.1145/3230543.3230547
[32] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, and kc claffy. 2013. AS Relationships, Customer Cones, and Validation. In *Proceedings of the ACM Internet Measurement Conference* (Barcelona, Spain) *(IMC '13)*. ACM Press, Barcelona, Spain, 243–256.
[33] Brian W Matthews. 1975. Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405, 2 (1975), 442–451.
[34] Fabricio Mazzola, Pedro Marcos, and Marinho Barcellos. 2022. Light, Camera, Actions: Characterizing the Usage of IXPs' Action BGP Communities. In *Proceedings of the 18th International Conference on Emerging Networking EXperiments and Technologies* (Roma, Italy) *(CoNEXT '22)*. Association for Computing Machinery, New York, NY, USA, 196–203.
[35] David Meyer. 1997. University of Oregon Route Views Archive Project.
[36] Rich Miller. 2011. Level 3 Buys Global Crossing for $3 Billion. https://www.datacenterknowledge.com/archives/2011/04/11/level-3-buys-global-crossing-for-3-billion.
[37] Alexandros Milolidakis, Tobias Bühler, Kunyu Wang, Marco Chiesa, Laurent Vanbever, and Stefano Vissicchio. 2023. On the Effectiveness of BGP Hijackers That Evade Public Route Collectors. *IEEE Access* 11 (2023), 31092–31124.
[38] Wolfgang Mühlbauer, Anja Feldmann, Olaf Maennel, Matthew Roughan, and Steve Uhlig. 2006. Building an AS-Topology Model that Captures Route Diversity. *ACM Computer Communication Review* 36, 4 (2006), 195–206.
[39] Ricardo Oliveira, Dan Pei, Walter Willinger, Beichuan Zhang, and Lixia Zhang. 2009. The (In) completeness of the Observed Internet AS-level Structure. *IEEE/ACM Transactions on Networking* 18, 1 (2009), 109–122.
[40] Lars Prehn and Anja Feldmann. 2021. How Biased is Our Validation (data) for AS Relationships?. In *Proceedings of the 21st ACM Internet Measurement Conference*. ACM, New York, US, 612–620.
[41] Isolario Project. 2019. BGPScanner. https://gitlab.com/Isolario/bgpscanner [Online; accessed 25-April-2024].
[42] Yakov Rekhter. 2006. RFC 4271: A Border Gateway Protocol 4 (BGP-4). https://www.rfc-editor.org/rfc/rfc4271
[43] Y Rekhter and T Li. 1994. RFC1654: A Border Gateway Protocol 4 (BGP-4).
[44] NCC RIPE. 2021. RIPE RIS Project.

[45] Loqman Salamatian, Todd Arnold, Ítalo Cunha, Jiangchen Zhu, Yunfan Zhang, Ethan Katz-Bassett, and Matt Calder. 2023. Who Squats IPv4 Addresses? *ACM SIGCOMM Computer Communication Review* 53, 1 (2023), 48–72.

[46] Martin Sendra, Rudy Sutrisno, Josep Harianata, Derwin Suhartono, and Almodad Biduk Asmani. 2016. Enhanced Latent Semantic Analysis by Considering Mistyped Words in Automated Essay Scoring. In *International Conference on Informatics and Computing (ICIC'16)*. IEEE, Lombok, Indonesia, 304–308. https://doi.org/10.1109/IAC.2016.7905734

[47] Brivaldo A Silva Jr, Paulo Mol, Osvaldo Fonseca, Italo Cunha, Ronaldo A Ferreira, and Ethan Katz-Bassett. 2022. Automatic Inference of BGP Location Communities. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 1 (2022), 1–23.

[48] One Step. 2022. One Step. https://onestep.net/communities/

[49] Florian Streibelt, Franziska Lichtblau, Robert Beverly, Anja Feldmann, Cristel Pelsser, Georgios Smaragdakis, and Randy Bush. 2018. BGP Communities: Even More Worms in the Routing Can. In *Proceedings of the Internet Measurement Conference 2018* (Boston, MA, USA) *(IMC '18)*. ACM, New York, NY, USA, 279–292. https://doi.org/10.1145/3278532.3278557

[50] Dan Tappan, Srihari R. Sangli, and Yakov Rekhter. 2006. BGP Extended Communities Attribute. https://doi.org/10.17487/RFC4360

[51] Richard Wray and Dan Milmo. 2005. Watchdog clears BSkyB acquisition of Easynet. The Guardian. https://www.theguardian.com/technology/2005/dec/31/news.citynews.

[52] Jianhong Xia and Lixin Gao. 2004. On the Evaluation of AS Relationship Inferences [Internet Reachability/Traffic Flow Applications]. In *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.*, Vol. 3. IEEE, Dallas, TX, USA, 1373–1377 Vol.3. https://doi.org/10.1109/GLOCOM.2004.1378209

[53] Man Zeng, Dandan Li, Pei Zhang, Kun Xie, and Xiaohong Huang. 2023. Federated route leak detection in inter-domain routing with privacy guarantee. *ACM Transactions on Internet Technology* 23, 1 (2023), 1–22.

Received August 2024; revised September 2024; accepted October 2024