IRRedicator: Pruning IRR with RPKI-Valid BGP Insights

Minhyeok Kang[†], Weitong Li[§], Roland van Rijswijk-Deij[‡], Ted "Taekyoung" Kwon[†], Taejoong Chung[§],

[†]Seoul National University, [‡]University of Twente, [§]Virginia Tech

***Abstract*—Border Gateway Protocol (BGP) provides a way of exchanging routing information to help routers construct their routing tables. However, due to the lack of security considerations, BGP has been suffering from vulnerabilities such as BGP hijacking attacks. To mitigate these issues, two data sources have been used, Internet Routing Registry (IRR) and Resource Public Key Infrastructure (RPKI), to provide reliable mappings between IP prefixes and their authorized Autonomous Systems (ASes). Each of the data sources, however, has its own limitations. IRR has been well-known for its stale Route objects with outdated AS information since network operators do not have enough incentives to keep them up to date, and RPKI has been slowly deployed due to its operational complexities. In this paper, we measure the prevalent inconsistencies between Route objects in IRR and ROA objects in RPKI. We next characterize inconsistent and consistent Route objects, respectively, by focusing on their BGP announcement patterns. Based on this insight, we develop a technique that identifies stale Route objects by leveraging a machine learning algorithm and evaluate its performance. From real trace-based experiments, we show that our technique can offer advantages against the status quo by reducing the percentage of potentially stale Route objects from 72% to 40% (of the whole IRR Route objects). In this way, we achieve 93% of the accuracy of validating BGP announcements while covering 87% of BGP announcements.**

I. Introduction

The Border Gateway Protocol (BGP) plays a crucial role in facilitating the exchange of routing information and the construction of routing tables across the Internet. Unfortunately, BGP was originally introduced almost three decades ago when its stable operations was the most critical consideration. As security aspects were not much considered in the BGP design, routers lack the ability to (1) authenticate the origin of the announced IP prefixes and (2) make informed decisions about whether incoming IP prefixes should be propagated or not. Such security issues have made the Internet plagued with prevalent security incidents: router misconfigurations such as route leaks, which caused multiple Internet outages [19], [61], [40], and attacks such as prefix hijacking [65], [8], [43], [58], to name a few.

To mitigate such limitations, the Internet Routing Registry (IRR) was introduced in 1995; IRR is a distributed database managed by Regional Internet Registries (RIRs) or Internet service providers (ISPs) so that network operators can publish their routing information by creating routing objects, or download routing objects of other Autonomous Systems (ASes) to validate and filter BGP announcements. Due to its simple and straightforward mechanisms, many network operators have been using it to protect their IP prefixes. Some network operators (e.g., Google [63]) also require others to register their IP prefixes with IRRs when building a peering relationship [38], [48], [47], [52]. However, it has been often criticized due to its prevalent stale objects; for example, based on some anecdotal evidence, Kuerbis et al. raised concerns that some objects do not seem to be updated, and stale objects tend to remain unchanged, and thus they are unreliable [34].

To overcome the limitation, Resource Public Key Infrastructure (RPKI) was introduced in 2008; the main objective was to provide cryptographically verifiable attestation via a Route Origin Authorization (ROA) object, which can bind an IP prefix to the AS who is authorized to announce it. Thus, routers can validate BGP announcements by checking if the origin Autonomous System Number (ASN) announcing an IP prefix matches with the ASN in ROA. Despite its strong attestation, however, it has not been widely deployed yet due to the negative impact of misissued ROA, the certificate dependencies in the hierarchy of RPKI [27], [29], and its incapability of route-leak protection. Also, RPKI depends on the Public Key Infrastructure (PKI) managed by RIRs, the quality of RPKI deployment and its management is significantly different across the RIRs; for example, we found that on March 1st, 2023, 59.2% of IP prefixes in RIPE NCC, the European RIR, were covered by ROA objects, while only 23.9% of IP prefixes in AFRINIC, the RIR for Africa, were done (§III), consistent with findings reported in 2019 [17]. Because of this limitation, Mutually Agreed Norms for Routing Security (MANRS) recommends the use of both IRR and RPKI [60].

Given the potential of IRR to complement the limitations of RPKI, several approaches have been suggested to enhance the quality of IRR objects. On one hand, certain Internet registries, such as the Japan Network Information Center (JPNIC) managing JPIRR, have attempted to enforce rules within their IRR databases to address stale objects; for instance, JPIRR removes IRR objects that have not been updated for a specified timeframe [20], typically a year. However, this approach relies on active participation from network operators, who are required to re-register their objects annually. Unfortunately, the lack of sufficient incentives to adhere to this policy hinders its effectiveness. On the other hand, RIPE NCC, the Internet registry for Europe, utilizes RPKI to discard IRR objects that fail validation against RPKI. Furthermore, Internet

Network and Distributed System Security (NDSS) Symposium 2024
26 February - 1 March 2024, San Diego, CA, USA
ISBN 1-891562-93-2
https://dx.doi.org/10.14722/ndss.2024.24524
www.ndss-symposium.org

Routing Registry Daemon (IRRd), a well-known IRR database server software [33], introduced support for validating IRR objects against RPKI starting from version 4 [56]. However, this approach has limited applicability since it only applies to *a subset of IRR objects that are covered by RPKI*, which constitutes less than half of the total IRR objects.

In this paper, we introduce a novel technique that improves the accuracy of IRR by eliminating inaccurate IRR objects while maintaining a comparable coverage. By conducting a comprehensive longitudinal study spanning 12 years, we investigate the deployment of both IRR and RPKI in conjunction with BGP announcements. Our study aims to shed light on the distinctions between RPKI-valid and invalid BGP announcements, enabling us to infer the validity of IRR objects not covered by RPKI.

Inspired by these observations, we leverage machine learning (ML) techniques to identify and flag invalid IRR objects. We then compare and evaluate our approach with other proposed techniques that share the same objectives. Our main findings and contributions are as follows:

- We conduct an extensive longitudinal study spanning 12 years and investigate the inconsistencies between RPKI and IRR using real BGP traces.
- We thoroughly analyze the BGP announcements associated with inconsistent IP prefixes, shedding light on how these prefixes are announced in the routing system.
- We develop an ML-based technique that effectively identifies and marks stale Route objects in the IRR.

To foster reproducibility and further research into improving the IRR and RPKI systems, we publicly release our analysis code, data, and machine learning model to the research community at

https://irredicator.netsecurelab.org

II. Background

A. Why it is hard to authenticate BGP announcements

Regional Internet Registries (RIRs) manage the allocation and registration of Internet number resources (e.g., IP addresses and ASNs). Currently, there are five RIRs globally: APNIC, ARIN, RIPE NCC, LACNIC, and AFRINIC [57]. These organizations allocate IP spaces and ASNs to National Internet Registries (NIRs) or Internet Service Providers (ISPs). Consequently, RIRs possess information about the *initial* organizations that are allocated IP prefixes. However, IP prefixes are *transferrable*, allowing NIRs to further allocate subsets of IP spaces to local ISPs, who can then re-allocate them to their own customers. Additionally, IP spaces can be leased or transferred between ASes based on business relationships.

Due to such characteristics, even RIRs are unable to keep track of a complete mapping between IP prefixes and the ASes authorized to announce them. This limitation has exposed the BGP to various security attacks [6], [35], [65], [14]. Attackers can exploit this vulnerability by announcing IP prefixes they do not legitimately own, leading to traffic diversion known as prefix hijacking.

B. Approaches to authenticating BGP

The Internet has dealt with this problem by building databases that store mappings between IP prefixes and authorized ASNs, allowing interested parties (e.g., ASes) to access and filter out *invalid* BGP announcements. Two mechanisms have been introduced and utilized to achieve this: Internet Routing Registry (IRR) and Resources Public Key Infrastructure (RPKI).

Internet Routing Registry is a globally *distributed* database of the routing information managed by RIRs and ISPs and was proposed in 1995 [9]. While IRR utilizes multiple objects to offer comprehensive routing information, our exposition primarily centers around the Route objects that specify the AS authorized to announce a specific IP prefix, allowing for a more focused analysis and understanding of the routing dynamics.

A Route object can carry multiple attributes, including `route`, a mandatory attribute for the IPv4 prefix to announce, `origin`, a mandatory attribute for the ASN that originates the `route`, `created`, an optional attribute for the date when the object is created, and `last-modified`, an optional attribute for the date when the object is updated. Network resource owners (e.g., ASes) can first create Route objects for mapping between IP prefixes and their corresponding origin ASes. These Route objects are then submitted to the database, which is managed either by RIRs (e.g., RIPE NCC) or by specific network operators (e.g., LEVEL3, NTT). The IRR managing entities have their own policies to verify Route objects submitted by ASes.

The objects stored in the IRR are written using the Routing Policy Specification Language (RPSL) [12], which offers a high level of expressiveness. This allows the objects to not only contain mapping information between IP prefixes and ASes but also include additional metadata related to ASes, such as AS relationships, AS sets, and more. Network operators can download these objects from the IRR for various purposes, including the validation of BGP announcements for route filtering, network troubleshooting, and router configuration. However, the voluntary effort-based nature of the IRR has led to criticism regarding the presence of *stale objects*, which reflect outdated ownership information for an IP prefix after it has been transferred to another AS. This can be primarily attributed to two main factors: (1) the lack of incentives for network operators to actively manage and update their objects and (2) the inability of third parties to remove outdated data [41].

Resource Public Key Infrastructure is a Public Key Infrastructure (PKI) that offers a cryptographically verifiable method of mapping IP prefixes to their respective origin ASes. To support this functionality, RPKI employs various objects, including:

- a CA certificate, which binds a set of number resources such as ASNs and IP prefixes to the public key of the owner.
- a Route Origin Authorization (ROA), which authorizes an AS to announce a specific set of IP prefixes[1]. These objects

[1] ROAs have `MaxLength` attributes that restrict the scope or coverage of these objects, specifically determining the upper limit for the prefix lengths they cover.

Auth. Objects	Measurement Period	# of Objects	% of Covered IPv4	% of Covered ASes
ROA	2011/01 – 2023/03	333 K	37.15	28.46
RADb	2016/08 – 2023/03	1.43 M	50.76	37.82
ALL-IRRs	2019/12 – 2023/03	2.69 M	74.23	68.80

TABLE I: Overview of the IRR and RPKI datasets as of March 1st, 2023; the number of Route objects and their coverage in IPv4 and ASes significantly surpass that of ROA objects.

are signed by the private key corresponding to the public key in the CA certificate.

In contrast to the IRR, *the trust of the objects in RPKI must be rooted from one of the RPKI trust anchors managed by the five RIRs – APNIC, RIPE NCC, AFRINIC, LACNIC, and ARIN*. Unlike IRR, where network operators have the flexibility to select any IRR database to register their Route objects, the RPKI object must ultimately be signed by the corresponding root trust anchor based on the initial assignment of an IP prefix; network operators who wish to register their ROAs must upload their objects to the repositories managed by the RIRs (or to the delegated repositories) so that they can get signed by the corresponding CAs. Thus, ASes can access the RPKI objects from the repositories or Relying Party (RP) software to authenticate BGP announcements, ensuring that the authorized ASes can be verified. Despite the benefits of RPKI, its adoption rate remains relatively low due to the complexity involved in registering and managing RPKI objects covering only a fraction of IPv4 spaces, ranging from 23.9% (AFRINIC) $\sim$ 59.2% (RIPE NCC). Furthermore, some ASes do not adopt RPKI due to compatibility issues with their hardware infrastructure [1].

C. Efforts to improve IRR

To address the issue of stale objects in IRR and enhance the overall quality of IRR objects, two approaches have been suggested:

1) IRR pruning with RPKI validation: RIPE NCC initiated the pruning of IRR objects by incorporating ROA validation. They announced their intention to eliminate Route objects from their IRR database if their IP prefixes were covered by ROA objects but had different ASNs associated with them [54]. In response, Internet Routing Registry Daemon (IRRd), a well-known IRR database server software [33], implemented this policy by validating all IRR objects with ROA objects in version 4 [56].
2) Age-based filtering: JPIRR removes Route objects that have not been updated for over a year [20] in order to encourage network operators to update their objects timely and ensure that outdated entries are removed.

However, both approaches have a limitation in terms of their coverage, as they only address a relatively small percentage of IRR objects, which will be presented in §VII.

III. Status Quo: IRR and RPKI Coverage

We first aim to determine the extent to which the IPv4 address space can be covered by either RPKI or IRR. Our focus is not solely on the registration of RPKI or IRR objects by network operators; we also pay attention to the proportion of BGP announcements that can be successfully validated using longitudinal datasets.

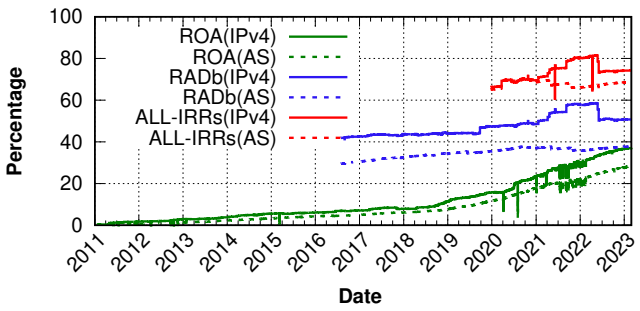

Fig. 1: The growth of RPKI and IRR in terms of the percentages of ASes and IPv4 addresses for each of these mechanisms.

A. Deployment Status of Route objects and ROAs

To gain insights into the deployment of RPKI and IRR by network operators, we analyze historical datasets of ROA and Route objects.[2] Table I shows the numbers of ROA and Route objects, and the percentages of ASes that have at least one of these objects. We use two kinds of IRR databases for our analysis: RADb, the most widely used, and ALL-IRRs, a composite database that merges RADb with other IRRs sourced from RIRs.

Route objects cover more than 74% of the total IP address space, excluding reserved IP addresses, whereas ROA objects only cover around 37% as of March 1st, 2023. However, the disparity between the two becomes more evident when considering the percentage of ASes that have adopted IRR or RPKI. Over 68% of ASes have adopted IRR, while about 28% have adopted RPKI. This indicates a higher deployment rate of IRR compared to RPKI among ASes.

We now focus on how they have been deployed over time. Figure 1 illustrates the time-varying coverage of IRR and RPKI with regard to ASes and the IPv4 address space[3]. We make a number of observations. First, we see an increasing trend of deployments. For instance, the coverage of the IPv4 address space by IRR increased from 66.3% in December 2019 to 74.2% in March 2023, while RPKI coverage grew from 15.8% to 37.2% during the same period.

Second, we observe that relying on RPKI solely to validate BGP messages may not arrive soon [29], [27]. Although the deployment rate of RPKI has notably increased since 2019, it currently covers less than 40% of the total IPv4 address space. This weighs the criticism regarding the low deployment of RPKI. It also emphasizes *the importance of leveraging IRR and enhancing its quality.*

[2]There have been multiple instances when the availability of ROA objects for downloading from the RIRs' repositories was disrupted; thus, we list such outages in §X and exclude outages from our analysis.

[3]We rely on historical datasets provided by the Number Resource Organization (NRO) to count the total numbers of IPv4 addresses and ASes [51].

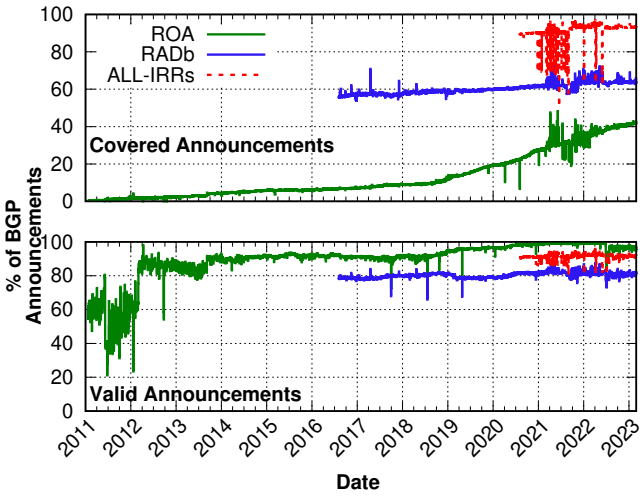

Fig. 2: Percentages of BGP announcements covered by IRR and RPKI (top) and percentages of valid announcements within each (bottom) are shown.

Third, we observe a significant decrease in the percentage of the IPv4 address space covered by IRR in mid-2022. The coverage dropped by approximately 8.3 percent points, declining from 81.5% on May 14th, 2022, to 73.2% on June 3rd, 2022, as 6.7K Route objects were removed during that period. We found that 3.4K (covering 8.1% of the IPv4 address space) Route objects belong to the DoD Network Information Center and none of them are covered by any ROA objects. This highlights a potential gap in coverage between IRR and RPKI, emphasizing the need for continuous efforts to improve the quality of IRR.

B. BGP announcements with IRR and RPKI

To understand how the Route or ROA objects are *practically* used, we measure their deployments over BGP announcements by utilizing BGP announcements collected from all vantage points of RouteViews [62].

1) BGP coverage by IRR and RPKI: We proceed to analyze the extent of BGP announcements that are *covered* by either Route or ROA objects. A BGP announcement is considered *covered* when its IP prefix matches exactly or partially at least one of the IP prefixes specified in Route or ROA objects.

Figure 2 (top) plots the percentages of BGP announcements covered by IRR and RPKI, respectively. Notably, the IRR coverage is 2.2 times higher than that of RPKI, accounting for 93.1% compared to RPKI's 42.1% as of March 1st, 2023. This indicates that RPKI is still lacking its coverage for about 58% of the total BGP announcements.

2) BGP validation with IRR and RPKI: We now examine how many of the BGP announcements are valid against either ROA or Route objects. To do so, we develop a validator that adheres to the standard logic of BGP announcement validation [44]. Figure 2 (bottom) shows the fractions of valid BGP announcements over time[4]

[4]When validating BGP announcements against IRR, we follow the same procedure as Resource Origin Validation (ROV) [44].

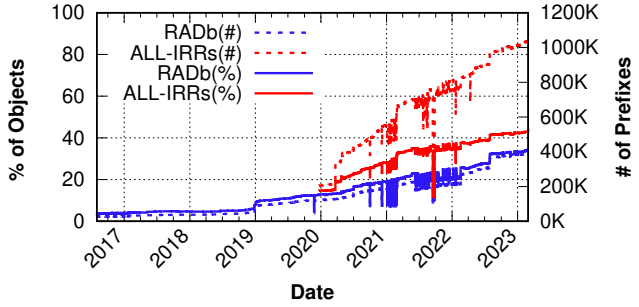

Fig. 3: The number of overlapping IP prefixes between Route and ROA objects and their portions in Route are shown.

We notice that the fraction of valid BGP announcements against IRR and RPKI has reached 92.7% and 97.4%, respectively, as of March 1st, 2023[5]. Also, we observe a significant drop in the percentages of valid BGP announcements against both RPKI and IRR on September 30th, 2017, July 19th, 2018, and April 26th, 2019. We find that the number of BGP announcements and the number of covered BGP announcements increased on those dates, but the number of valid BGP announcements was almost the same as the other days. We also find that AS37468, operated by Angola Cables, made more than 176K BGP announcements on July 19th, 2018, which it did not announce for the four weeks around that day, which aligns with the report in [17].

Key Takeaways: Our analysis reveals a superior coverage of IRR compared to RPKI from both the perspectives of object registration (74.2% vs. 37.2% of the IPv4 space) and BGP coverage (93.1% vs. 42.1%). However, a deeper examination of the validity of BGP announcements reveals the limitations of IRR, indicating that it alone may not be reliably used for the validation of BGP announcements.

IV. Inconsistencies in IRR and RPKI Validation

Our goal is to investigate approaches that utilize RPKI to provide a reliable mapping between ASNs and their announced IP prefixes, with the aim of eliminating inaccurate IRR entries. One promising strategy is to identify IP prefixes and ASNs that are covered by both Route and ROA objects and to remove Route objects that are inconsistent with RPKI validation. However, this approach may be less effective if the number of overlapping objects is too low, or if the quantity of inconsistent objects is minimal. Thus, in this section, we now examine how much of IP prefixes are covered by both Route and ROA objects and how much of their authorized ASes are *inconsistent*.

A. Overlap between Route and ROA Objects

Initially, we examine the extent of overlap between ROA and Route objects by evaluating two metrics: (1) the count of IP prefixes that are covered by both RPKI and IRR and (2) the proportion of Route objects with IP prefixes that are also

[5]Note that not all BGP announcements are valid due to attacks (e.g., BGP hijacking) or misconfigurations; thus it would be infeasible for the valid ratio to reach 100%.

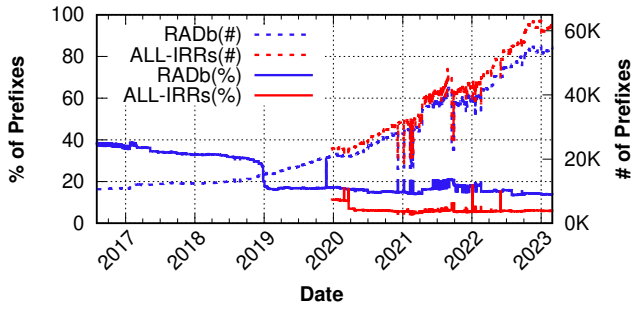

Fig. 4: The number and the percentage of inconsistent prefixes are shown.

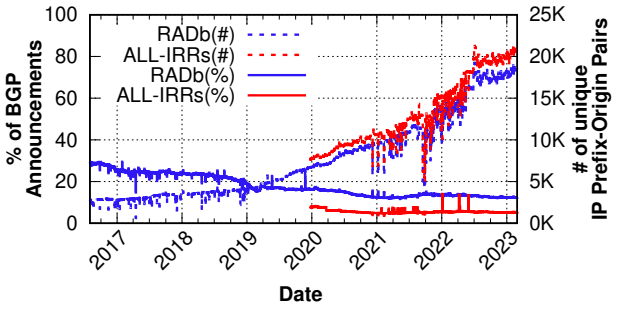

Fig. 5: The number and the percentage of BGP announcements covered by inconsistent Route objects.

covered by ROA objects relative to the total number of Route objects. Figure 3 presents the findings, from which we derive several observations.

First, the number of overlapping *IP prefixes* across the ROA and Route objects increases overall; we observe over 1.08 M IP prefixes in ALL-IRR are also registered in RPKI as of March 1st, 2023.

Also, the percentage of *Route objects* that can be covered by both ROA and Route objects has increased rapidly since 2019. This trend has been more clear as the number of registered ROA objects notably increases starting from 2019 (Figure 1). For example, 44.6% (1.2 M) of Route objects in ALL-IRR can be matched by ROA objects as of March 1st, 2023. This implies that many IP prefixes that have already been registered in IRR now begin to be also covered by ROA objects as the RPKI deployment rate grows; for example, we observe a sharp increase in the number of IP prefixes covered by both between December 22nd, 2018 and January 12th, 2019. This was because three large Taiwanese ASes[6], which had already managed their Route objects since 2000, started deploying RPKI during the above dates by registering 1,354 ROA objects covering 36,481 IP prefixes.

B. Inconsistent origin ASes of IP prefixes

As the number of IP prefixes covered by both ROA and Route objects increases, it is critical to maintain consistent origin ASNs for such IP prefixes. In practice, however, it is challenging because (1) network operators do not always keep their Route objects up to date, and (2) Route objects do not have any validity period. Thus, it may bring operational issues when the validation results across ROA and Route objects *fail*. For example, routers that use only Route objects to filter invalid BGP announcements could reject BGP announcements, that are actually valid according to RPKI, and vice versa.

We now try to understand how many inconsistent IP prefixes exist and whether they grow over time in terms of numbers, especially with regard to BGP announcements. To do so, we first consider only IP prefixes for which there are both Route and ROA objects that match. Then we classify a given IP prefix to be *consistent* if their origin ASes are the same; otherwise, we call it *inconsistent*. We obtain inconsistent IP prefixes from our collected ROA and Route objects and Figure 4 shows the fraction and the number of such IP prefixes over time.

We make a number of interesting observations; first, the percentage of the inconsistent IP prefixes decreases from 11.2% to 5.7% (from 38.9% to 13.8% in RADb) during our measurement period.

However, we also notice that *the number of such inconsistent IP prefixes keeps increasing*; for example, the number of such IP prefixes has been increased by a factor of 2.7 (5.2 for RADb) during our measurement period. At first glance, this may look contradictory; however, it is because of a combination of two interesting phenomena. First, the percentage of inconsistent IP prefixes can decrease as network operators who had already registered the Route objects for their IP prefixes started to create ROA objects with the same IP prefixes (that are consistent); the three Taiwanese ISPs who already had their Route objects registered deployed their ROA objects to the APNIC RPKI repository. Figure 1 explains the sharp drop between December 22nd, 2018 and January 12th, 2019.

On the other hand, when new ROA objects are registered by the authorized ASes, their IP prefixes may have a conflict with the stale Route objects with the outdated ASNs, making the number of inconsistent IP prefixes grow[7]. Based on the slow but increasing trend of the RPKI adoption rate, we can expect that the number of inconsistent IP prefixes might keep increasing, making the situation worse.

C. Inconsistent IP Prefixes in BGP announcements

As we observe the gap between validation results against IRR and RPKI, a natural question that arises is how many BGP announcements are affected by the inconsistent IP prefixes, which might have an impact on routing table construction. Figure 5 plots the number of BGP announcements covered by the inconsistent IP prefixes and their percentage over time. We first immediately notice that 5.3% (12.3% for RADb) of the BGP announcements covered by both ROA and Route objects are inconsistent in our latest snapshots *even though its percentage has been decreased from 7.8% (29.5% for RADb)*. Similar to the prior findings, the number of *inconsistent* BGP

[6]AS9674, AS4780, and AS9919

[7]Considering that ROA objects can only be registered by the real owner of IP prefixes, we can assume that normally the conflicted Route objects are stale.

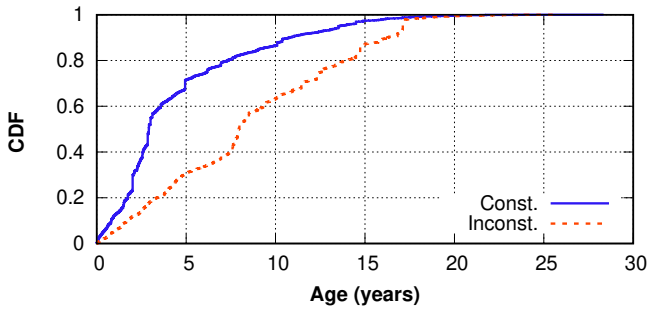

Fig. 6: The cumulative distributions of the ages of consistent and inconsistent Route objects are shown.

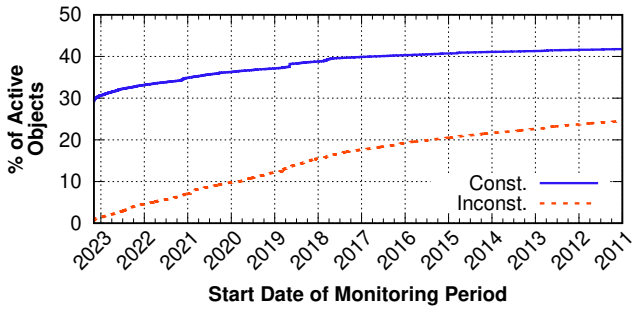

Fig. 7: The percentages of active Route objects are plotted. The end date of the monitoring period is March 1st, 2023. The x-axis is the start date of the monitoring window in a reverse direction. A Route object is active if at least one BGP announcement corresponding to the object is observed within a monitoring period; otherwise, it is inactive. Thus, the majority of the inconsistent Route objects (75.5%) have not been announced since 2011.

announcements increased during our measurement period from 7.7K to 20.8K (from 2.8K to 18.6K for RADb).

Key Takeaways: We discover that 1.2 M (44.6%) IRR objects, as of March 1st, 2023, are covered by corresponding ROA objects, primarily attributable to the rising adoption of RPKI. However, among these, 61.6 K (5.7%) are found to be inconsistent with ROA validation outcomes, suggesting their removal. A point of concern is the escalating count of such IRR objects and the number of BGP announcements they cover, indicating a trend of increase.

V. Characterizing Inconsistent Route Objects

We observed a better coverage of IRR in terms of the IPv4 space it encompasses, as well as its actual utilization in BGP announcements; however, we also noted that 61.6 K (5.7%) of those entries that are also covered by RPKI are *inconsistent*, thus invalid. This suggests that employing existing RPKI validation methodologies [54], [56] to prune IRR entries would yield only a marginal improvement (less than 5.7%), leaving other stale entries undetected. Our ultimate objective is to characterize such objects and further devise a methodology to identify those not covered by RPKI. In this section, we explore the characteristics of inconsistent Route objects by comparing them with *consistent* objects.

A. Ages of inconsistent Route objects

There has been anecdotal evidence that stale Route objects contribute to *inconsistent* IP prefixes [34], prompting some network operators to make efforts to keep their Route objects up-to-date. For instance, JPIRR (Japanese IRR) follows a policy of removing Route objects that have not had their `last-modified` attributes changed for over a year [20]. In our analysis, we use the `last-modified` attribute for measuring the age of Route objects to identify any notable patterns or trends related to their age.

Figure 6 shows the cumulative distribution of ages of consistent and inconsistent Route objects. As expected, inconsistent Route objects tend to be older than consistent Route objects. The median and 90th percentile ages of inconsistent Route objects are 7.9 years (2,899 days) and 16.3 years (5,944 days), respectively, while those of consistent Route objects are 2.9 years (1,064 days) and 10.8 years (3,928 days). While older ages are more commonly associated with inconsistent Route objects, relying solely on age to identify potentially stale Route objects may have unintended consequences. For example, there are 1.67% of consistent Route objects that have an age greater than the 90th percentile age of inconsistent Route objects. This indicates that a valid Route object, whose IP prefix has been legitimately advertised for a significant period by an authorized origin AS, may also be mistakenly categorized as stale.

B. Activeness of inconsistent Route objects

In order to understand how inconsistent IP prefixes are announced in BGP, we shift our attention to analyzing the activeness of these prefixes. We introduce an activeness metric that determines whether a specific prefix-origin pair is actively used in BGP or not. To measure activeness, we examine consecutive BGP snapshots within a designated `monitoring window`.

We consider the AS active if it has *ever* announced that prefix within a monitoring window extending from time x to the latest available snapshot. To gain insights into the behavior of ASes, we gradually widen the monitoring window toward the oldest snapshot. Our hypothesis is that origin ASes in stale Route objects are unlikely to continue announcing the same IP prefixes as they previously did. Consequently, while these stale Route objects may appear inactive within a small monitoring window, they may become active as the monitoring window grows. We apply this metric for all the pairs of IP prefixes and their ASes in Route objects and calculate the portion of active pairs as the monitoring window is varied. Also, we plot the activeness metric for both consistent and inconsistent Route objects (those covered by ROA objects) as well.

Figure 7 shows how the percentages of active objects change as we expand the monitoring window. We find interesting observations.

Firstly, inconsistent Route objects exhibit extremely low activity. When considering a monitoring window of just a single day (i.e., the latest snapshot), we find that only 0.6% of IP prefix-origin pairs matching inconsistent Route objects

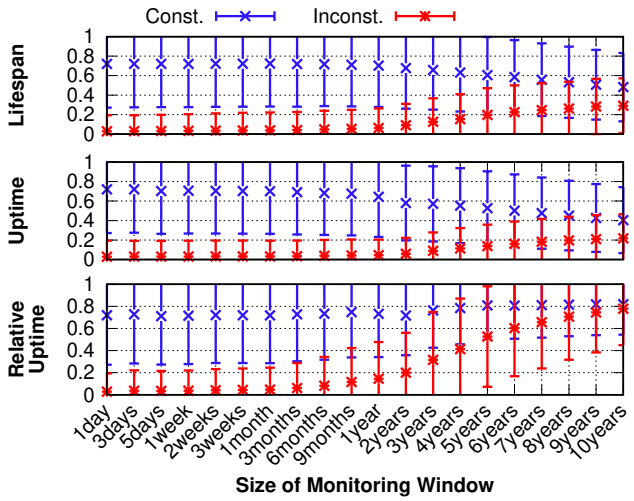

Fig. 8: The average `Lifespan`, `Uptime`, and `Relative Uptime` of consistent and inconsistent Route objects are plotted with the monitoring window size increased.

are observed in BGP announcements. This increases to 24.5% when the monitoring window is extended to the longest (12 years).

Secondly, the consistent Route objects are generally more active than inconsistent Route objects. Nevertheless, the majority of them remain unannounced in BGP for over 12 years. 29.3% consistent IP prefixes are observed when considering a single-day monitoring window, and 41.8% when considering the longest monitoring window.

Thirdly, we notice that the curves of the percentages of active objects become flattened as the monitoring window size increases. For example, we see the percentage of active IP prefixes in consistent Route objects increases by 9.4 percent points for the first five years from the latest snapshot, but it only increases by 3.1 percent points for the remainder of the measurement period (7.2 years).

On average, consistent Route objects are more likely to be active than inconsistent Route objects. However, it is important to note that classifying Route objects based solely on activeness introduces the risk of making mistakes, regardless of the size of the monitoring window.

C. BGP patterns of inconsistent IP prefixes

Similar to a previous work [2] that characterizes BGP announcements, we introduce three metrics to capture BGP patterns more comprehensively: `Lifespan`, `Uptime`, and `Relative Uptime`. These metrics are extensions of the age and activeness metrics, and they provide a more nuanced understanding of BGP announcements over time. We first encode the observations of BGP announcements for each prefix-origin pair to a bit vector v where $v[i]$ is 1 if the corresponding BGP announcements are observed before i days from the current date, otherwise 0. Let v be a bit vector, w be a monitoring window, $v[:l] = (v[i])_{0 \le i \le l}$ denote a subset of a bit vector, and $UpIndices(v) = \{x \mid 0 \le x \le size(v) \wedge v[x] = 1\}$ denote a set of indices whose the corresponding bits equal to 1, then

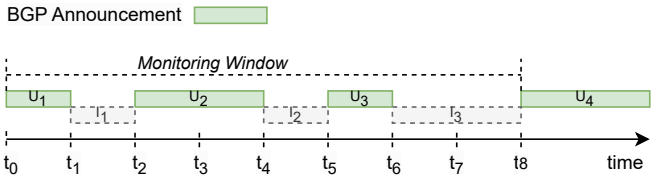

Fig. 9: Illustration of the metrics for the BGP model. `Uptime` is the sum of the days for U_1, U_2, and U_3; `Lifespan` is calculated as $t_6 - t_0 + 1$; Up is 2; $Down$ is 3; $ActiveDays$ consists of U_1, U_2, and U_3; $InactiveDays$ includes I_1, I_2, and I_3.

`Lifespan`, `Uptime`, and `Relative Uptime` can be defined as follows.

- $Uptime(v, w) = sum(v[:w])/w$
  `Uptime` extends the activeness metric to offer a more quantitative measure. It calculates the number of days that the BGP announcement is observed within the monitoring window.
- $Lifespan(v, w) = (max(UpIndices(v[:w])) - min(UpIndices(v[:w])) + 1)/w$
  `Lifespan` extends the age metric to account for the monitoring window. Unlike age, which measures the date difference between creation and the latest analysis, `Lifespan` calculates the span between the first and last days of observed BGP announcements within the window.
- $RelativeUptime(v, w) = Uptime(v, w)/Lifespan(v, w)$
  `Relative Uptime` integrates `Lifespan` and `Uptime` to gauge BGP announcement frequency relative to lifespan. It is the quotient of uptime and `Lifespan`, offering insight into how often BGP announcements for a prefix were observed across its lifespan.

These three metrics will be measured using twenty monitoring windows of varying sizes ranging from one day to ten years. By characterizing Route objects using these metrics, we can gain a deeper understanding of how BGP announcements for an IP prefix have been made over time. We analyze these metrics for all the BGP announcements in Route objects, including consistent and inconsistent Route objects as shown in Figure 8. We find interesting observations.

First, consistent Route objects have higher `Lifespans` and `Uptimes` compared to inconsistent Route objects. This indicates that consistent Route objects have been observed for a longer period of time and have been more frequently announced in BGP compared to inconsistent Route objects.

Second, as the monitoring window increases, we observe a slight increase in the `Lifespans` and `Uptimes` of inconsistent Route objects, while consistent Route objects show a decrease. This implies that inconsistent Route objects were actively used in the past but have become less relevant over time. In contrast, consistent Route objects have more recent usage patterns.

Third, the `Relative Uptimes` of inconsistent Route objects steadily increase with the growth of the monitoring window, reaching a value of 0.78. This value becomes almost comparable to those of consistent Route objects (0.82).

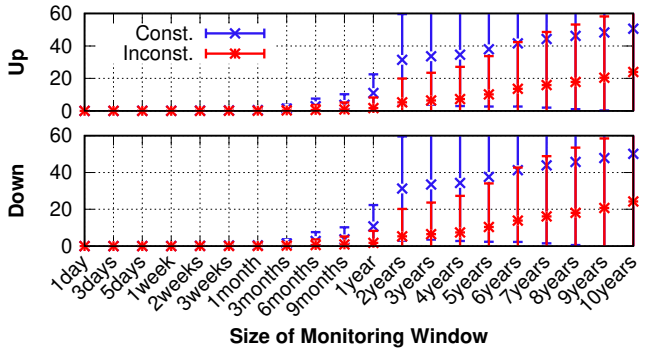

Fig. 10: Ups and Downs of consistent and inconsistent Route objects are plotted.

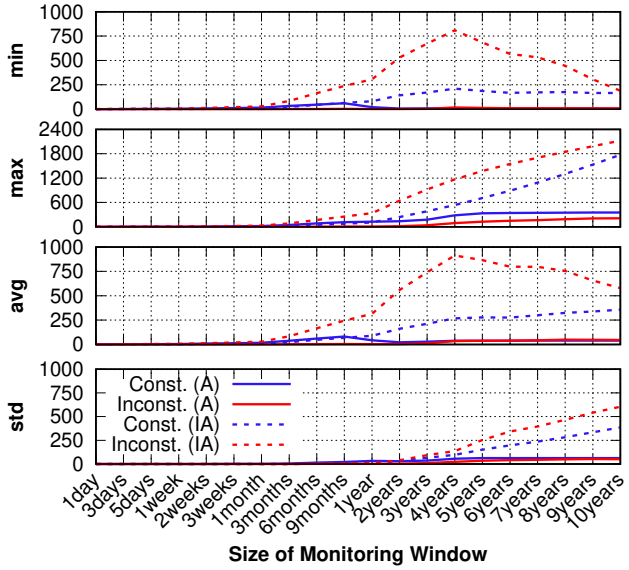

Fig. 11: The statistics of Active Days and Inactive Days of consistent and inconsistent Route objects are plotted.

This finding indicates that the IP prefixes in consistent Route objects have been actively announced and used in BGP, while inconsistent Route objects were initially correct but have become outdated and are no longer announced.

D. BGP dynamics of inconsistent IP prefixes

To capture the dynamic nature of prefix-origin pairs in BGP, we define ten metrics: Up, Down, minimum, maximum, average, and standard deviation for both Active Days and Inactive Days as depicted in Figure 9. These metrics help us understand the changes in the activeness of Route objects over time.

- The Up metric measures the number of transitions from inactive to active state.
- The Down metric measures the number of transitions from active to inactive state.
- The Active Days metric refers to a list of consecutive days during which BGP announcements are made. We measure the minimum, maximum, average, and standard deviation of these active days.
- The Inactive Days metric refers to a list of consecutive days during which BGP announcements are not made. We measure the minimum, maximum, average, and standard deviation of these inactive days.

In Figure 10, the average of Ups and Downs of consistent and inconsistent Route objects are shown. Both consistent and inconsistent Route objects show an increase in Up and Down metrics as the monitoring window size increases, while consistent Route objects exhibit higher values compared to inconsistent Route objects. This implies that consistent Route objects exhibit a higher degree of dynamic behavior in their BGP announcements, as they are more active, as depicted in Figure 7. In our analysis of the four statistics of Active Days and Inactive Days as shown in Figure 11, we make the following interesting observations. In general, consistent Route objects exhibit higher values for all statistics related to active days compared to inconsistent Route objects. Conversely, inconsistent Route objects tend to have higher values for all statistics related to inactive days compared to consistent Route objects. One particularly noteworthy finding is that inconsistent Route objects have significantly longer inactive periods than consistent Route objects, especially when considering the minimum and average values. This indicates that inconsistent Route objects experience longer periods of inactivity.

Key Takeaways: We noted distinct patterns between consistent and inconsistent Route objects, both in their lifetime and in the dynamics of their BGP announcements; for example, a mere 0.6% of IP prefix-origin pairs corresponding to inconsistent Route objects are observed in BGP announcements, whereas for consistent ones, this figure rises to around 30%. These pronounced differences indicate that utilizing such patterns could facilitate the identification of inconsistent objects, *even when they do not intersect with ROA objects*. This will be further elaborated in the subsequent section.

VI. Improving the Quality of Route Authorization

We now leverage the unique characteristics between *inconsistent* and *consistent* Route objects to categorize other potentially stale objects, *which are not in overlap with ROA objects* by using machine learning techniques.

A. Datasets

1) Dataset Construction: To train the candidate models, we use a training dataset $\{\boldsymbol{x_i}, y_i\}_{i=1}^{n}$, where $\boldsymbol{x_i}$ is a feature vector and y_i is the label assigned to the i-th Route object.

The feature vector $\boldsymbol{x_i}$ consists of a total of 312 features, which can be categorized into two groups:

- Window-based Features: These features are derived from our metrics. We measure the value of each of 13 metrics[8] across 20 different monitoring window sizes, yielding a total of 260 features.

[8]These include Lifespan, Uptime, Relative Uptime, Up, Down, along with the min, max, avg, and std for both Active Days and Inactive Days.

- Statistical Features: For each metric, there is a set of 20 measured values for the 20 monitoring window sizes. For each set, we compute statistical measures such as the minimum, maximum, average, and standard deviation, which results in total 52 features. These statistical features provide insights into the distribution of the 20 measured values (for each metric) over the monitoring windows, enhancing the representation of the Route objects' characteristics.

The label for a Route object (y_i) is assigned based on its RPKI validation status; the label is set to 1 if the object is RPKI-valid and 0 otherwise. Route objects that are not covered by ROA objects are left unlabeled and are consequently excluded from the training dataset.

2) Dataset Filtering: We refine our dataset by removing mislabeled Route objects, focusing on two main reasons.

The first reason involves mislabeled Route objects resulting from errors in the configuration of the `MaxLength` attribute in the corresponding ROA objects, as reported in [17]. For instance, a Route object covered by a ROA object with a valid origin AS may be incorrectly labeled as invalid due to a misconfigured `MaxLength` attribute.

The second reason pertains to invalid Route objects with different ASNs from those of the corresponding ROA objects, even though both ASNs belong to the same ISP. For example, a Route object authorizing AS 133480 to announce 103.131.235.0/24 is labeled invalid (which is wrong) due to a ROA object authorizing AS 9910 to announce 103.131.235.0/22. However, both ASes belong to the same ISP, Intergrid Group Pty Ltd.

To summarize, we refine our datasets by taking into account the `MaxLength` attributes of ROA objects (to mitigate the first problem) and establishing AS relationships between the two ASNs (to mitigate the second problem).

B. Model Selection

The classification task could be handled by setting reasonable thresholds for each metric and comparing the features of a Route object with these thresholds to determine if it is stale or not (i.e., heuristic approach). However, we use a machine learning (ML) model instead of using a set of rules or thresholds as it offers several advantages for classifying *inconsistent* Route objects:

- Automated threshold selection: ML models can automatically determine an optimal threshold for each feature during the training process. This eliminates the need for manual selection and fine-tuning of thresholds, which can be time-consuming and subjective.
- Generalization to unseen data: ML models can be generalized to unseen data by learning underlying patterns and relationships from the training set. They can capture the inherent variability and diversity in the data, allowing for accurate classification of new Route objects that were not a part of the training set.
- Robustness to noises and outliers: ML models are generally more robust to noises and outliers in the data compared to rule-based approaches. They can learn from a large number of training examples, identify relevant patterns, and handle noisy or inconsistent data points more effectively.

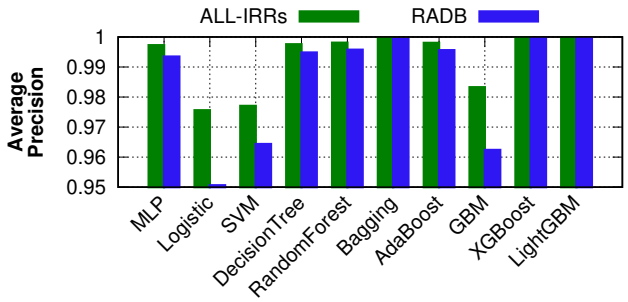

Fig. 12: The average precision of of each candidate ML model is shown. For each candidate model, we measure the average precision of testsets consisting of Route objects from ALL-IRR and RADb, respectively.

Overall, the use of an ML model offers greater accuracy, generalizability, and robustness in classifying *inconsistent* Route objects compared to rule-based approaches using thresholds.

Thus, we compare various ML models to identify the one that performs the best on the features obtained using our metrics. Specifically, we consider ten candidate models for comparison purposes: multi-layer perceptron (MLP), logistic regression, support vector machine (SVM), decision tree, random forest, bagging, AdaBoost, gradient boosting machine (GBM), XGBoost, and LightGBM. By comparing the performance of these models on the features, we can select the most effective one for classifying Route objects. We evaluate the performance of candidate models using the average precision, i.e., the area under the precision-recall curve. The ideal model should exhibit high precision to effectively identify stale Route objects and high recall to ensure practical reliability. Figure 12 illustrates the average precision value of each candidate model. Among the models considered, LightGBM demonstrates the highest performance, achieving an average precision of 0.9998 for the ALL-IRR testsets. and 0.9996 for the RADb testset.[9]

C. Further Design Considerations

When identifying stale Route objects, it is crucial to exercise caution in order to prevent incorrect predictions that could lead to the elimination of valid objects, particularly when the model is uncertain of their staleness. To mitigate this risk, we integrate the *classification with rejection* technique into our model; this approach refrains from making a prediction when the model lacks adequate confidence in classifying a given instance. This technique offers two main advantages. Firstly, it helps mitigate the risks of false positives or false negatives by avoiding potentially erroneous predictions. Secondly, it enables expert intervention in the decision-making process as predictions are deferred.

[9]Among the evaluated models, Bagging, XGBoost, and LightGBM demonstrated high performance. However, LightGBM stands out as the most efficient in terms of training time. This efficiency is particularly beneficial given the continuously expanding size of the global routing table.

In particular, we apply the *classification with rejection* technique [16] to the LightGBM model. This technique provides the model with the ability to reject a prediction when it encounters uncertainty (i.e., there is no dominant class candidate) or ambiguity (i.e., there are multiple dominant class candidates) for a given instance. For implementation, we apply a custom loss function as defined in [16] and integrate with our model; specifically, we train the model with the loss function $\mathcal{L}(\boldsymbol{g}; \boldsymbol{x}, y) = c\phi(g_y(\boldsymbol{x})) + (1-c)\phi(-g_{y'}(\boldsymbol{x}))$, where $\boldsymbol{g}(\boldsymbol{x}) = [g_0(\boldsymbol{x}), g_1(\boldsymbol{x})]^T$, $g_y(\boldsymbol{x})$ is the score function for class y, ϕ is a binary margin surrogate loss [10], $c \in (0, 0.5)$ is the rejection cost, and $y' \neq y$. Since we need a score output for each class to calculate a loss, we set the attributes of the LightGBM model as follows: objective = 'multiclass' and 'num_class' = 2. The prediction output for a Route object by our model is calculated by applying the softmax function to the scores, e.g., the prediction value of class 1 is given by $e^{g_1}/(e^{g_0} + e^{g_1})$. Finally, we can reject a prediction if either $\max_y g_y(\boldsymbol{x}) \leq 0$ (indicating uncertainty) or $\exists y, y'\ s.t.\ y \neq y' \wedge\ g_y(\boldsymbol{x}), g_{y'}(\boldsymbol{x}) > 0$ (indicating ambiguity).

D. Applying into Practice

We assume that our approach is adopted by IRR administrators, such as RADb and RIRs. The validation process of each Route object by IRR operators follows these steps:

1) If the prefix-origin pair of the Route object has not been observed by any vantage points within the largest monitoring window, it is marked as inactive. Otherwise, proceed to step 2.
2) If the rejection conditions, as described in §VI-C, are met, the Route object is marked as rejected. Otherwise, proceed to step 3.
3) The Route object is labeled with a predicted value based on our classification model. A higher value indicates a higher probability of being valid.

Due to the flexible structure for RPSL and IRR objects, we can add extra attributes to IRR objects [12]. For instance, IRRd version 4, a widely used IRR database server software [56], introduces 'rpki-ov-state' attribute into Route objects to store the validation result obtained from RPKI. In a similar manner, the validation result obtained from our technique can be stored in a new attribute within Route objects. This allows network operators, such as ISPs, to apply their own policies when interpreting the validation results. For example, an operator may choose to disregard or assign lower priority to the origin information of a prefix in a Route object if its predicted value (from our ML model) falls below a threshold set by the operator.

VII. Evaluation

Now, we evaluate our model in terms of (1) classification performance, (2) generalizability, and (3) comparison with other IRR filtering approaches. For each evaluation, we harness Bayesian optimization [22] techniques with 5-fold cross-validation to autonomously fine-tune a spectrum of hyper-parameters, which is one of our design goals (§VI-B).

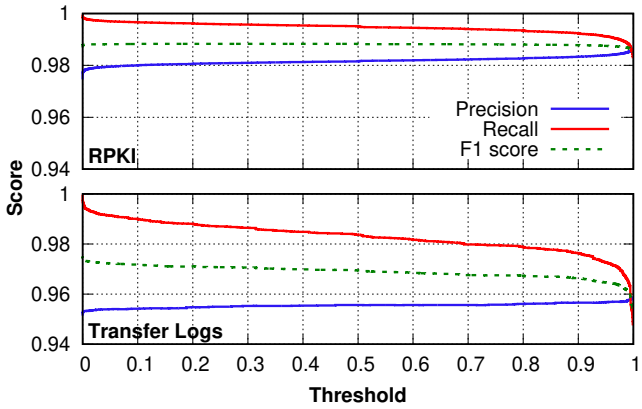

Fig. 13: The evaluation results of our model using the testset labeled with validation results against RPKI (top) and transfer logs (bottom).

A. Model Performance

First, we evaluate the classification performance of our model using a testset labeled with validation results against RPKI. We evaluate our model ten times and aggregate the prediction results to compute the precision, recall, and F1 scores, as depicted in Figure 13 (top). For each evaluation, we perform a stratified sampling [49] on the dataset collected on March 31, 2023. Specifically, we partitioned the dataset, utilizing 80% of it to create a training dataset through stratified sampling, and the remaining 20% was set aside as a test dataset. We measure the `precision` and `recall` of our model by varying the `threshold`. A Route object is considered stale if our model does not reject the prediction and the prediction output is lower than the `threshold`. We observe that our model achieves remarkably high performance; for a threshold value of 0.202, we achieve a maximum F1-score of 0.988, with precision of 0.981 and recall of 0.996. Our model performs well across a wide range of threshold values, with both precision and recall consistently above 0.98 when choosing a threshold between 0.1 and 0.999.

B. Model Generalizability

To test the *generalizability* of our model, we evaluate its performance on a testset labeled with a ground truth dataset *unrelated to RPKI*, which is transfer logs.

IP prefixes and other Internet resources can be transferred from one organization to another under the RIR's supervision. When an IP prefix is transferred, the corresponding RIR makes the transfer information (e.g., IP prefix, old and new organizations[11] participating in the transition, and the transition date) publicly available [4], [55], [5], [39], [3]. The latest transfer log corresponding to an IP prefix gives information that (1) the source is no longer the owner of the prefix and (2) the recipient is the owner of the prefix after the transfer date.

[10] We use a logistic loss, $\phi(z) = log(1 + e^{-z})$

[11] Note that transfer logs do not contain any ASN information because an organization (such as an ISP) can announce its IP prefixes from one of their ASes that they manage or can simply lease its IP prefixes to another AS to let them announce.

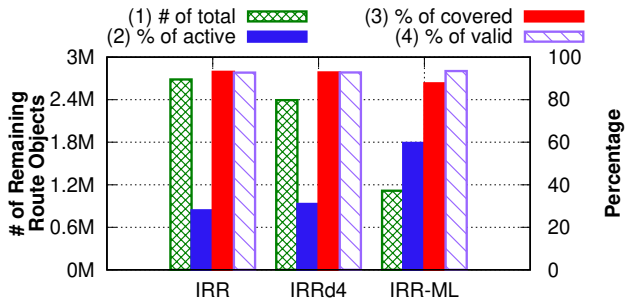

Fig. 14: The three approaches (as to how to filter the IRR data) are compared in terms of the number of remaining Route objects, the ratio of active Route objects, the ratio of covered BGP announcements, and the ratio of valid BGP announcements.

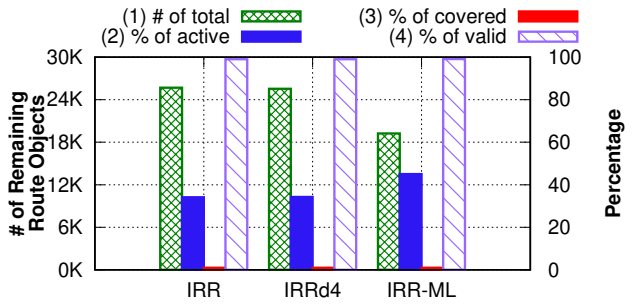

Fig. 15: The three approaches are compared in terms of the number of remaining Route objects, the ratio of active Route objects, the ratio of covered BGP announcements, and the ratio of valid BGP announcements. Note that the whole IRR contains only the Route objects in JPIRR.

To validate a Route object using transfer logs, we apply the following criteria: (1) a Route object is considered stale if the organization specified as the source in the most recent transfer log is the ISP of the AS identified in the Route object; (2) a Route object is categorized as non-stale if it the organization listed as the recipient in the latest transfer log corresponds to the ISP associated with the AS in the Route object *and* no ASes tied to any other ISP have consistently announced the prefix for a pre-defined period (e.g., two weeks). All the other Route objects that do not meet either of the above criteria remains unlabeled.

Using this approach, we identify 25,143 valid Route objects and 1,323 stale (or invalid) Route objects. We evaluate our model with these datasets. As shown in Figure 13 (bottom), our model still exhibits strong performance across different ground truth datasets; we achieve a maximum F1-score of 0.975, with precision of 0.952 and recall of 0.999 with the 0.000003 threshold. When using the same threshold value as in the previous evaluation (0.202), we achieve a precision of 0.955, recall of 0.988, and F1-score of 0.971. These results highlight the generalizability of our model in identifying stale Route objects, even with the dataset unrelated to RPKI.

C. Comparison with other IRR filtering approaches

We now compare the proposed ML model in comparison with (i) IRR without filtering and (ii) IRR pruning with RPKI validation. We evaluate the three approaches in terms of (1) the number of remaining Route objects after being filtered by each approach, (2) the ratio of *active*[12] Route objects to remaining Route objects, (3) the ratio of the *covered* BGP announcements to the whole BGP announcements, and (4) the ratio of valid BGP announcements to the *covered* BGP announcements. Here, *covered* BGP announcements means the number of BGP announcements that are covered by the remaining Route objects filtered by each approach.

We first compare the whole IRR (without any filtering), IRRd4 (filtered by RPKI validation), and IRR-ML (filtered by our ML model) in terms of the four criteria, which is shown in Figure 14. We find that IRRd4 exhibits a comparable performance with the whole IRR in terms of BGP-related criteria (3) and (4) even if it filters out 0.29 M (10.8%) Route objects [13]. IRR-ML filters out 1.57M (58.5%) Route objects from the whole IRR, and yet it reveals a slight decrease in the ratio of the covered BGP announcements; these findings could prompt questions about the effectiveness of RPKI validation, which filters a mere 10.8% of ALL-IRR objects, while IRR-ML filters 58.5% of ALL-IRR objects (Figure 14).

To delve deeper into these numbers, we analyze the BGP announcement patterns of the filtered prefixes. Our analysis reveals that only 3.12% of Route objects deemed invalid by IRRd4 (and thus, by RPKI) are still actively announced. On the other hand, only 0.07% of Route objects filtered by IRR-ML are observed in BGP announcements. *This suggests that IRR-ML is more effective in identifying and filtering out stale or unused Route objects*. Further supporting this claim, the average `Uptime` for Route objects filtered by IRR-ML is a mere 0.027 when examined over a 14-day monitoring window, equating to an average active period of approximately 0.38 days in the last two weeks.

We now show the benefits of IRR-ML compared to age-based filtering, we compare the three approaches only using JPIRR as shown in Figure 15. IRRd4 filters out a small number of Route objects from JPIRR and hence shows almost identical performance with the JPIRR in all criteria. On the other hand, IRR-ML filters out 6.5K (25.1%) Route objects from JPIRR. While there is a negligible decrease in the ratio of the covered BGP announcements, the percentage of active Route objects increases by 11.0 percent points achieving 45.0%. These results highlight that IRR-ML can improve the quality of the IRR database managed by the age-based filtering policy.

VIII. Related Work

In this section, we review the literature for understanding the security challenges in BGP and the approaches for enhancing BGP security.

[12]Observed more than once during the last 14 days.

[13]The percentage of filtered Route objects (10.8%) is greater than the percentage of *inconsistent* prefixes in ALL-IRR (5.7%) since a prefix can be associated with multiple Route objects.

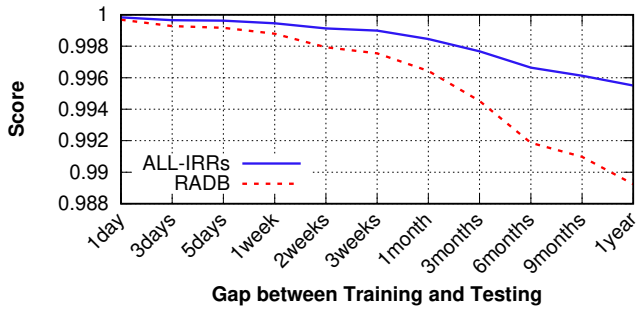

Fig. 16: As the gap between training and testing times increases, its performance is gradually decreased.

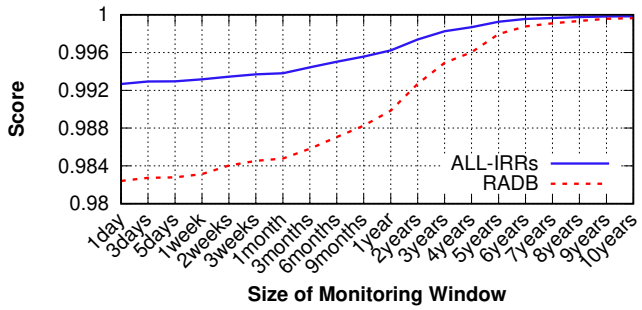

Fig. 17: As the size of the monitoring window increases, its performance is gradually increased.

A. *Efforts to improve BGP routing security*

There is a large body of work that focused on security issues in BGP [7], [28], investigated common misconfigurations [45], or identified overall challenges to securing interdomain routing [24]. To address its vulnerabilities, many security extensions to BGP were proposed such as soBGP [64], S-BGP[36], and BGPsec [26]. Some studies [25], [13], [27] focused on quantifying the effectiveness of these security protocols. However, due to the massively distributed nature of the Internet, these protocols have not been deployed much. To tackle the low deployment problem, Subramarian et al. proposed a methodology to verify bogus route advertisements by using cryptographic signatures in the control plane [59], Cheng et al. proposed a classification model based on LSTM to capture BGP anomalies [15], and Gill et al. suggested a strategy to encourage the adoption of BGP security protocols (e.g., BGPsec) by providing some financial incentives to ISPs [23].

B. *Efforts to sanitize interdomain routing information*

IRR was proposed in 1995 and has been actively used by network operators such as Google [21], Cloudflare [11], and Internet Exchange Points (IXPs) such as AMX-IX [32] and DE-CIX [10]. It is also recommended by network governance organizations such as MANRS [48] because of its extensive coverage and expressiveness. However, the issues of IRR regarding its staleness have been raised multiple times [41], [34], [50], [18]. For example, Du et al. [18] quantifies the consistency between Route and ROA objects using IRR databases collected a monthly granularity and reported that 20% of Route in RADb are covered by ROA objects and only 38% of them are *consistent* with ROA objects as of October 2021[14]. Some studies focused on improving the security by suggesting incentives for network operators to maintain the objects up to date [34] and by removing IRR objects once they have not been updated longer than a year [20]. However, they were not successful mainly because of the missing ground truth information about legitimate objects.

C. *Efforts to build a PKI for BGP*

RPKI [37] was introduced in 2008 but has been deployed slowly. Chung et al. [17] reported that the deployment is growing, but also there are deployment disparities across the RIRs. To tackle the low deployment problem, Hlavacek et al. proposes a method to register ROAs automatically [29] to encourage network operators. Some studies [31], [30] focused on identifying vulnerabilities that can give insights into how to further strengthen the security of RPKI.

Our study extends these prior studies in three ways. *First*, we examine the deployment of *all* ROAs fetched from all the RIRs since its inception and analyze them with Route objects collected from RADb, all five RIR's IRRs, and JPIRR, curated with a *daily* granularity. *Second*, we *analyze the usage pattern of Route objects within BGP* by introducing various metrics. This enables us to discern the BGP patterns of Route objects and compare those of *consistent* and *inconsistent* Route objects, furthering our objective of enhancing IRR quality. *Third*, we present an ML-based approach to *enhance the quality of IRR* by leveraging the analyzed patterns of BGP announcements.

IX. Discussion

In this section, we discuss various aspects of our model and challenges.

A. *Model Reliability*

Our primary question is whether the model will maintain reliable performance over time *without* requiring updates post-training. To investigate this, we utilize a one-day snapshot taken at t_0 (March 1st, 2023) for testing, while training the model on a snapshot from t_{0-x}. Figure 16 presents the evaluation results as we progressively increase the time gap between the training and testing snapshots; as anticipated, testing immediately after training yields the best performance, which only marginally decreases over a one-month gap (achieving an AUC score of 0.998 for ALL-IRR and 0.996 for RADb). Remarkably, even when using a one-year-old dataset for training, the model continues to exhibit high accuracy, maintaining performance levels greater than 0.989.

[14]While the percentage of overlapping Route objects aligns with our analysis, there is a discrepancy in the reported percentage of consistency. As of October 2021, our analysis indicated a consistency in RADb of 81.9%, whereas the study reports 38%. This difference can be attributed to the fact that they measured the percentage based on the number of Route objects, whereas we measured it based on the number of prefixes. It suggests that inconsistent prefixes are associated with multiple Route objects in the dataset, resulting in a lower consistency percentage when measured at the Route object level.

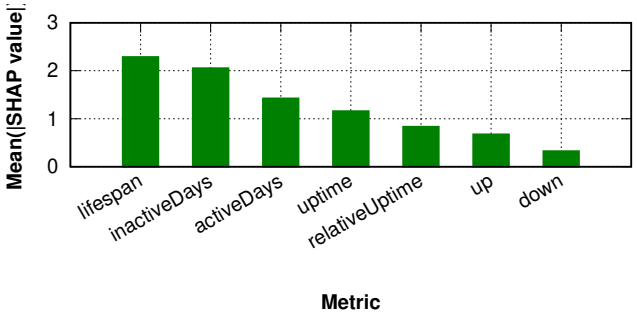

Fig. 18: The mean of the absolute SHAP values aggregated by metrics. Metrics with high absolute SHAP value are more influential in the prediction of the model.

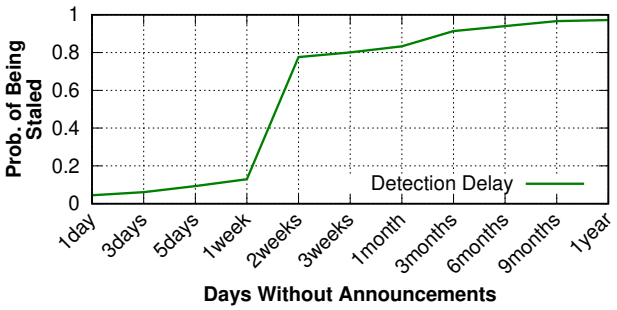

Fig. 19: As the prefix owner stops making announcements for an extended period, the likelihood of the object being considered stale increases.

Next, we assess the model's performance by varying the size of the monitoring window. As shown in Figure 17, we observe that extending the monitoring window contributes to performance improvement by utilizing a more comprehensive longitudinal dataset. However, for IRR organizers, the continuous monitoring of all BGP announcements could be operationally taxing. Remarkably, even with a monitoring window as brief as one day, the model still exhibits excellent performance, achieving an average precision of 0.993 for ALL-IRR and 0.982 for RADb. The performance degradation is relatively minimal when the monitoring window is reduced.

B. Model Explainability

To provide the IRR operators with insight into how the classification model operates, we employ SHapley Additive exPlanations (SHAP) [46] to elucidate the influence of individual features on the model's output. The SHAP values assess the contribution of each feature to the model predictions on a set of given inputs. As demonstrated in Figure 18, we aggregate gain and SHAP values by metrics. Notably, the metrics `Lifespan` and `Uptime` emerge as particularly important. This indicates that both the age of the Route object and the consistency of its corresponding BGP advertisements are crucial factors. This is further corroborated by the gain score associated with the metric `Inactive Days` and `Active Days`. This underscores the importance of the duration for which a prefix-origin is observable in BGP data for the effective classification of stale Route objects.

C. Model Resiliency

The proposed ML model is designed to be resilient against attacks that manipulate the inputs of the model, such as adversarial examples and poisoning. The nature of our model's features, which are based on historical public BGP announcements collected from multiple vantage points, makes it difficult for attackers to successfully manipulate the inputs. To launch adversarial attacks, attackers would need to craft inputs that lead to misclassification during the testing phase. In our case, this would involve announcing invalid BGP announcements that match stale Route objects to cause false positives or discarding a series of valid BGP announcements that match valid Route objects to cause false negatives. However, these manipulations are highly impractical for several reasons. First, many network operators have implemented route origin validation with RPKI or employed other techniques like BGP filters to filter out invalid origins. This makes it challenging for attackers to consistently announce invalid BGP announcements that match stale Route objects, as they would likely be blocked by these filtering mechanisms. Second, launching a successful poisoning attack typically requires corrupting a substantial portion of the training data. In our case, this would mean manipulating the features of over 26.8K Route objects that constitute 1% of the total 2.68 million Route objects as of March 1st, 2023, which is highly unlikely to go unnoticed. As a result, the practical feasibility of launching effective attacks against our model is extremely low.

To assess the resiliency of our model, we conduct an evaluation to detect real-world BGP attacks. We employ a dataset of prefix hijacks from BGPStream [53], which consists of (prefix, AS-PATH) pairs that potentially signify hijack incidents. Our strategy involves extracting the features from the (prefix, hijacker's origin) pairs and subsequently predicting using our model. The outcomes provide insights into the model's efficacy in identifying evasion attacks. Out of a total of 1,014 potential hijack incidents, we exclude 93 cases where the origin is valid against RPKI. Within the remaining 921 potential hijack incidents, our model successfully identified 608 of them (66.0%) as invalid.

D. Detection Delay

Our methodology depends on features that can be adaptively defined within a *monitoring window*, such as 2 weeks or 1 year. As a result, there might be a delay in identifying a stale Route object, particularly if the object was actively advertised but has recently ceased announcements due to being transferred to another origin, without deleting the Route object. For example, if the legitimate holder of a Route object has announced the prefix for a year and subsequently transfers it to another AS, a noticeable detection delay could occur in identifying these changes as indicative of stale objects.

Inspired by previous work [29] that focused on the *de-facto-owner* of a prefix, we incorporate *label-flipping augmentation* [42], [66] to address this challenge. The approach is straightforward: we create a label-flipped dataset by perturbing the features of a valid Route object and reversing its label; for example, if the original label is 'valid', we manipulate their

BGP announcement patterns to mimic a lack of announcements over the last x days, altering relevant features. We adhere to previous research that defines the *de-facto-owner* of a prefix as an entity that has announced for at least 11 consecutive days, meaning a prefix-origin that has not announced in the last 11 days is no longer considered a de-facto owner. We then compute the perturbed feature from this altered bit vector and change the label to 'invalid'. This allows the IRR operator to identify stale entries that have not been used in BGP announcements for a specified period, like 11 days.

By applying this approach, as shown in Figure 19, it becomes evident that a stale object can be identified if it has not been announced for two weeks. However, it is worth noting that network operators might have legitimate reasons for not announcing IP prefixes, such as traffic engineering or security concerns. In such cases, the introduced technique could inadvertently delete valid entries.

X. Conclusion

In this paper, we conducted a large-scale, longitudinal study of inconsistent Route objects in Internet Routing Registry (IRR), which are validated against Route Origin Authorization (ROA) objects in Resource Public Key Infrastructure (RPKI). We find that such inconsistent Route objects have increased over time even if the RPKI adoption rate slowly grows. Moreover, the number of inconsistent IP prefixes has been increased by a factor of 2.7 (5.2 for RADb) during our measurement period. We characterize how inconsistent IP prefixes are announced over BGP and propose a technique that identifies stale Route objects by leveraging a machine learning (ML) algorithm. Our proposed ML model, IRR-ML, improves the quality of filtered Route objects compared to previous methodologies such as RPKI-based filtering and age-based filtering. Specifically, IRR-ML effectively filters out stale Route objects that are rarely used in BGP advertisements. Specifically, IRR-ML filters out 1.55 M (57.8%) stale Route objects from the entire IRR, while RPKI-based filtering eliminates only 0.29 M (10.8%). Our technique can offer advantages in comparison to the status quo by providing more trustworthy routing information.

Acknowledgment

We extend our heartfelt gratitude to the anonymous reviewers and for their invaluable insights. This research was supported in part by NSF grant CNS-2323137, Google, the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2021-0-02048) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT). (NRF-2022R1A2C2011221, No. RS-2023-00220985).

References

[1] 16.2R2-S9: Software Release Notification for Junos Software Service Release version 16.2R2-S9. https://supportportal.juniper.net/s/article/16-2R2-S9-Software-Release-Notification-for-Junos-Software-Service-Release-version-16-2R2-S9?language=en_US.

[2] B. Andre, N. Evi, and C. KC. Internet expansion, refinement and churn. *European Transactions on Telecommunications*, 13(1), Wiley Online Library, 2002.

[3] AFRINIC Transfer Logs. https://ftp.afrinic.net/pub/stats/afrinic/transfers/.

[4] APNIC Transfer Logs. https://ftp.apnic.net/pub/stats/apnic/transfers/.

[5] ARIN Transfer Logs. https://ftp.arin.net/pub/stats/arin/transfers/.

[6] H. Ballani, P. Francis, and X. Zhang. A study of prefix hijacking and interception in the Internet. *CCR*, 37(4), 2007.

[7] K. Butler, T. R. Farley, P. McDaniel, and J. Rexford. A survey of BGP security issues and solutions. *Proceedings of the IEEE*, 98(1), IEEE, 2010.

[8] R. Brandom. Hackers emptied Ethereum wallets by breaking the basic infrastructure of the internet. 2018. https://www.theverge.com/2018/4/24/17275982/myetherwallet-hack-bgp-dns-hijacking-stolen-ethereum.

[9] T. Bates, E. Gerich, L. Joncheray, J. Jouanigot, D. Karrenberg, M. Terpstra, and J. Yu. Representation of IP Routing Policies in a Routing Registry (ripe-81++). RFC 1786, IETF, 1995.

[10] BGP Announcement Filtering – Extract from the route server guides. https://www.de-cix.net/en/about-de-cix/news/insights-how-and-what-the-de-cix-route-servers-filter.

[11] BGP filtering best practice. https://twnog.tw/wp-content/uploads/2019/06/7-Cloudflare-Jimmy-twnog3.jimmylim.bgpfilteringbestpractice_1555600440.pdf.

[12] A. Cengiz, V. Curtis, G. Elise, K. David, M. D. M, B. Tony, K. Daniel, and T. Marten. Routing Policy Specification Language (RPSL). RFC 2622, IETF, 1999.

[13] H. Chan, D. Dash, A. Perrig, and H. Zhang. Modeling adoptability of secure BGP protocol. *SIGCOMM*, 2006.

[14] J. Cowie. China's 18-Minute Mystery. 2010. https://dyn.com/blog/chinas-18-minute-mystery/.

[15] M. Cheng, Q. Li, J. Lv, W. Liu, and J. Wang. Multi-scale LSTM model for BGP anomaly classification. *IEEE Transactions on Services Computing*, 14(3), IEEE, 2018.

[16] N. Charoenphakdee, Z. Cui, Y. Zhang, and M. Sugiyama. Classification with rejection based on cost-sensitive classification. *ICML*, 2021.

[17] T. Chung, E. Aben, T. Bruijnzeels, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, R. van Rijswijk-Deij, J. P. Rula, and N. Sullivan. RPKI is Coming of Age: A Longitudinal Study of RPKI Deployment and Invalid Route Origins. *IMC*, 2019.

[18] B. Du, G. Akiwate, T. Krenc, C. Testart, A. Marder, B. Huffaker, A. C. Snoeren, and K. C. Claffy. IRR Hygiene in the RPKI Era. *PAM*, 2022.

[19] L. Doe. China Telecom suffers internet outage. 2019. https://www.capacitymedia.com/articles/3823594/china-telecom-suffers-internet-outage.

[20] Detailed rules for object registration of JPIRR service. 2006. https://www.nic.ad.jp/doc/jpnic-01045.html.

[21] Expanding our commitment to secure Internet routing. https://cloud.google.com/blog/products/networking/how-google-is-working-to-improve-internet-routing-security.

[22] P. I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[23] P. Gill, M. Schapira, and S. Goldberg. Let the market drive deployment: a strategy for transitioning to BGP security. *SIGCOMM*, 2011.

[24] S. Goldberg. Why is It Taking So Long to Secure Internet Routing? *ACM Queue*, 12(8), 2014.

[25] S. Goldberg, M. Schapira, P. Hummon, and J. Rexford. How secure are secure interdomain routing protocols? *SIGCOMM*, 2010.

[26] W. George and S. Murphy. https://tools.ietf.org/html/rfc8206. RFC 8206, IETF, 2017.

[27] Y. Gilad, A. Cohen, A. Herzberg, M. Schapira, and H. Shulman. Are We There Yet? On RPKI's Deployment and Security. *NDSS*, 2017.

[28] A. Herzberg, M. Hollick, and A. Perrig. Secure Routing for Future Communication Networks (Dagstuhl Seminar 15102). 2015. http://drops.dagstuhl.de/opus/volltexte/2015/5267/.

[29] T. Hlavacek, I. Cunha, Y. Gilad, A. Herzberg, E. Katz-Bassett, M. Schapira, and H. Shulman. Disco: Sidestepping rpki's deployment barriers. *NDSS*, 2020.

[30] T. Hlavacek, P. Jeitner, D. Mirdita, H. Shulman, and M. Waidner. Behind the scenes of RPKI. *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022.

[31] T. Hlavacek, P. Jeitner, D. Mirdita, H. Shulman, and M. Waidner. Stalloris: RPKI Downgrade Attack. *USENIX Security*, 2022.

[32] Implementation of RPKI and IRR filtering on the AMS-IX platform. https://www.ripe.net/support/training/ripe-ncc-educa/presentations/use-cases-stavros-konstantaras.pdf.

[33] Internet Routing Registry Daemon (IRRd) Version 4. https://github.com/irrdnet/irrd.

[34] B. Kuerbis and M. Mueller. Internet routing registries, data governance, and security. *Journal of Cyber Policy*, 2(1), 2017.

[35] B. Kevin, F. T. R, M. Patrick, and R. Jennifer. A survey of BGP security issues and solutions. *Proceedings of the IEEE*, 98(1), 2009.

[36] C. Lynn, J. Mikkelson, and K. Seo. Secure BGP (S-BGP). IETF, 2003.

[37] M. Lepinski and S. Kent. An Infrastructure to Support Secure Internet Routing. RFC 6480, IETF, 2012.

[38] LACNIC Develops Its Internet Routing Registry (IRR). 2019. https://prensa.lacnic.net/news/en/institutional/lacnic-develops-its-internet-routing-registry-irr.

[39] LACNIC Transfer Logs. https://ftp.lacnic.net/pub/stats/lacnic/transfers/.

[40] A. Medina. CenturyLink / Level 3 Outage Analysis. 2020. https://www.thousandeyes.com/blog/centurylink-level-3-outage-analysis.

[41] D. McPherson, S. Amante, E. Osterweil, L. Blunk, and D. Mitchell. Considerations for internet routing registries (IRRs) and routing policy configuration. RFC 7682, IETF, 2015.

[42] G. Matt, A. Yoav, B. Victoria, B. Jonathan, B. Ben, C. Sihao, D. Pradeep, D. Dheeru, E. Yanai, G. Ananth, and others. Evaluating models' local decision boundaries via contrast sets. *arXiv preprint arXiv:2004.02709*, 2020.

[43] O. Moll. Border Gateway Protocol Hijacking - Examples and Solutions. 2020. https://www.anapaya.net/blog/border-gateway-protocol-hijacking-examples-and-solutions.

[44] P. Mohapatra, J. Scudder, D. Ward, R. Bush, and R. Austein. BGP Prefix Origin Validation. RFC 6811, IETF, 2013.

[45] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP misconfiguration. *SIGCOMM*, 2002.

[46] S. M. maLundberg and S. Lee. A unified approach to interpreting model predictions. *NIPS*, 2017.

[47] MANRS IXP Programme. https://www.manrs.org/ixps/actions/.

[48] MANRS for Network Operators. 2021. https://www.manrs.org/netops/network-operator-actions/.

[49] J. Neyman. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. *Breakthroughs in Statistics: Methodology and Distribution*, Springer, 1992.

[50] K. Nagahashi and H. Esaki. Research on Routing Consistency in the Internet RoutingRegistry. *Electronics and Communications in Japan*, 89(9), 2006.

[51] NRO RIR Statistics. https://www.nro.net/about/rirs/statistics/.

[52] Network Operator Participants. 2023. https://www.manrs.org/netops/participants/.

[53] C. Orsini, A. King, D. Giordano, V. Giotsas, and A. Dainotti. BGPStream: A Software Framework for Live and Historical BGP Data Analysis. *IMC*, 2016.

[54] RIPE NCC IRR Database Non-Authoritative Route Object Clean-up. https://www.ripe.net/participate/policies/proposals/2018-06.

[55] RIPENCC Transfer Logs. https://ftp.ripe.net/pub/stats/ripencc/transfers/.

[56] RPKI integration. https://irrd.readthedocs.io/en/stable/admins/rpki/.

[57] Regional Internet Registries. https://www.nro.net/about/rirs/.

[58] A. Siddiqui. A Major BGP Hijack by AS55410-Vodafone Idea Ltd. 2020. https://www.manrs.org/2021/04/a-major-bgp-hijack-by-as55410-vodafone-idea-ltd/.

[59] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. H. Katz. Listen and whisper: security mechanisms for BGP. *NSDI*, 2003.

[60] M. Stucchi. Mutually Agreed Norms for Routing Security (MANRS) Implementation Guide. https://github.com/manrs-tools/manrs-docs/blob/main/pdf/MANRS-Network-Implementation-Guide.pdf.

[61] L. Tung. iCloud goes down: Apple joins the Google, Facebook, Cloudflare cloud outage club. 2019. https://www.zdnet.com/article/icloud-goes-down-apple-joins-the-google-facebook-cloudflare-cloud-outage-club/.

[62] University of Oregon: The Route Views Project. http://www.routeviews.org.

[63] Updating Internet Routing Registry (IRR) data to peer with Google. 2019. https://support.google.com/interconnect/answer/9368848.

[64] R. White. Architecture and Deployment Considerations for Secure Origin BGP (soBGP). IETF, 2006.

[65] YouTube Hijacking: A RIPE NCC RIS case study. 2008. https://www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study.

[66] J. Zhou, Y. Zheng, J. Tang, J. Li, and Z. Yang. Flipda: Effective and robust data augmentation for few-shot learning. *arXiv preprint arXiv:2108.06332*, 2021.

APPENDIX

A. RPKI Repository Outage

Through the entire snapshot of RPKI objects, we observe that ROA objects from the APNIC repository were inaccessible on the following dates: November 24, 2019, August 3, 2020, January 4, July 15, 19, 23, 31, August 10, 2021, and 21 days in September 2021. Similarly, for the RIPENCC repository, we experienced difficulties accessing ROA objects on March 7, 8, 2015, April 6, and August 3, 2020. Furthermore, ROA objects

from the LACNIC repository were unavailable on February 28, 2021.