Clearing Clouds from the Horizon: Latency Characterization of Public Cloud Service Platforms

Rita Ingabire*†, Antonio Bazco-Nogueras†, Vincenzo Mancuso†, Luis M. Contreras‡ and Jesus Folgueira‡
*Universidad Carlos III de Madrid, Madrid, Spain,
†IMDEA Networks Institute, Madrid, Spain,
‡Telefónica Innovación Digital, Madrid, Spain

***Abstract*—Services rely more and more on cloud platforms to offer their products to end users. This implies that being able to estimate the latency required to reach those cloud platforms is of growing importance. To shed light on this crucial aspect, we perform a three-month measurement campaign involving traceroute measurements every 30 minutes over 256 pairs of source-destination probes, where the vantage points are located in different Cloud Service Providers (CSPs) and the destination probes belong to one of the main Infrastructure Operators (IOs) of Spain. We provide interesting insights obtained from analyzing the data resulting from this campaign. Among them, we observe that, as expected, distance is the unavoidable factor impacting cloud latency. Yet, other results are less anticipated, such as the great stability of the network, or the lack of performance difference when comparing standard and premium network service tiers. We also analyze the potential of forecasting the cloud latency both for future samples but also for unobserved connections.**

***Index Terms*—latency, measurements, RIPE Atlas, forecasting, cloud services, cloud performance**

I. Introduction

In today's world, Internet has become an indispensable component of the daily life of most people, permeating nearly every aspect of society as an enabler of new services. Among the many components that compose and interact with Internet, cloud services have emerged as one of the key factors that have facilitated the recent flourishing of new applications. As a result, if we want to ensure that those services are successfully provided, we must be able to characterize and anticipate the performance of cloud services through the network.

Mapping the network performance is a challenging and crucial task, yet we have only been able to complete partial analyses [1]. Network latency is one of the most impactful Key Performance Indicators (KPIs) affecting user Quality of Service (QoS) in online applications [2], and its significance is growing as interactive cloud-based applications continue to develop. However, characterizing network latency between cloud servers and end users accessing the cloud-based service is a complex task, because this latency depends on user location, server distance, time of access, technology of access, and network topology, among other aspects. As for cloud-based services, there have been works focused on characterizing their performance from their emergence [3] to recent years [4]–[7].

This work is supported by the project AEON-CPS (TSI-063000-2021-38), funded by the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union NextGeneration-EU in the framework of the Spanish Recovery, Transformation and Resilience Plan. A. Bazco-Nogueras is supported by the grant 2020-T2/TIC-20710 for Talent Attraction, Madrid Region, Spain.

In this work, we aim to characterize and predict the network latency required to reach the cloud. For that, we focus on the cloud services ecosystem in Spain, a mid-side, developed country that presents some characteristics that can be extrapolated to other regions of the world. In particular, we consider three main CSPs: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). We make use of the RIPE Atlas platform [8] to perform a `traceroute` measurement campaign from probes located in these CSPs towards probes that are connected to the network of Telefónica, one of the main IOs in Spain. The objective of restricting to a specific IO is twofold: we limit the impact that potential diversity of network planning strategies may have in the measurements, and, moreover, Telefónica is the operator with the greatest set of probes connected to RIPE Atlas, which improves the quality of the derived results.

This work aims at shedding light on the performance of cloud-based services and their dependency on latency. Through the `traceroute` measurements, we generate a database for latency characterization and forecasting. Our approach is non-cooperative since it does not require a privileged point of view or privileges inside the IO's infrastructure, and it makes use of active probing. Our main findings are:

- We characterize the temporal and spatial distribution of the Round Trip Time (RTT) between three different CSPs (AWS, Azure, and GCP) and Telefónica's end users. We obtain the RTT density distribution for each one of the cloud vantage points, as a function of the hour of the day, as a function of the distance between endpoints, and depending on the specific path that the route follows.
- From the previous analysis, we show that the latency presents a notable stability over time, not dependent on the moment of the day, and mainly based on distance.
- We found that placing cloud access at the IO's edge achieves better RTT than using prevalent CSPs' platforms.
- We provide methods to forecast the future connectivity performance of the public cloud infrastructure for two scenarios: (i) predicting from past latency measurements on the same link, and (ii) predicting the performance of a new link from measurements from other connections between devices located at different places. For that, we leverage both statistical and Machine Learning (ML) algorithms, also providing explainable methods to understand which features matter the most in making latency predictions.

II. Related Work on Public Cloud Measurements

Network latency and cloud performance have been measured and analyzed throughout the years, with works focusing on different aspects of the problem. We summarize the state of the art for the topics that are close to our work, and we provide a discussion and comparison based on our findings in Section VI.

1) Tools and platform for cloud measurements: CloudBeacon was introduced in [9]; this JavaScript tool was designed to evaluate how CSPs should strategize their edge networks or determine optimal locations for satellite data centres. CLAudit, a prototype planetary-scale cloud-latency auditing platform, was introduced in [10]; it utilizes the experimental PlanetLab [11] network to measure latency on Microsoft's Azure cloud service. A subsequent investigation leveraged CLAudit to interpret identified suspicious latency events over two one-month time intervals [12]. In [13], the authors designed CLASP to assess the speed from GCP for three distinct speed-test servers (Okla, MLab, and Comcast). The research focused on detecting network congestion from throughput and latency variability from a five-month measurement campaign.

2) Comparisons between cloud services: Palumbo *et al.* presented in [4] a comparative analysis of two of the main CSPs: Microsoft Azure and AWS. The study involved a 14-day experimental campaign utilizing 25 vantage points on the PlanetLab platform. We refer to [4, Table I] for a detailed comparison of the works dealing with latency measurements in cloud networks until 2020. In [3], the authors developed Cloudcmp, which allows users to select the most suitable CSP based on performance and cost. Michelinakis *et al.* evaluated in [6] the interconnection among apps, network operators, and CSPs, to understand the relationship between cloud computing and mobile network operators, and gauge users' perceptions of these services. The data was collected using the now-discontinued MONROE database [14], and the findings revealed that 85% of the mobile apps relied heavily on CSPs.

3) Network analysis through RIPE Atlas: RIPE Atlas [8] is one of the most prominent tools enabling world-wide Internet measurements. It is consistently used by the Internet measurement community to perform distinct studies such as measuring last-mile congestion [15], suggesting where to place the edge servers [16], geolocating IP infrastructure [17], characterizing the lack of data centres in Africa [18], or analyzing the impact of the non-optimal internet topology [19].

In [5], Mohan *et al.* examined the reachability and latency of cloud centres using the RIPE Atlas platform, to understand whether edge computing is necessary to satisfy the latency requirements of modern applications. For that, they employed over 3200 RIPE Atlas probes distributed across 166 countries as vantage points for measurements. The study considered seven CSPs and consisted of `ping` measurements every 3 hours. This work was extended in [20], where they also incorporated `traceroute` measurements every 24 hours.

Tools such as RIPE Atlas also bear some limitations [21], [22]. While RIPE Atlas comprises thousands of probes, it naturally presents some bias: many probes enjoy favoured network connectivity due to who are the entities and individuals that install them [21], and there are geographic biases towards certain countries and urban locations [22]. Furthermore, it is intrinsically complex to quantify such bias [22].

III. Measurement Methodology

Accessing public clouds services requires us to traverse the public Internet network, which implies passing across different Autonomous Systems (ASs) and Internet service providers (ISPs), although the CSPs are increasingly relying on their private networks to better control their end users QoS. In this situation, obtaining rich data about the configuration or topology of the network is often unfeasible, and a thorough investigation of cloud latency requires a large number of measurement points, which must be also geographically spread.

The scientific community has made several attempts to develop platforms for carrying out latency analysis in this context. However, this task requires a taxing effort that needs to be sustained over time. Indeed, platforms such as [11], [14] were discontinued after some years of successful functioning.

The measurement campaign has been conducted through the RIPE Atlas platform [8], an open and collaborative measurement platform developed and maintained by RIPE NCC [23]. It is a global network composed of devices acting as vantage points (also called *probes*) that actively measure Internet connectivity. Anyone can access this database, there are APIs to facilitate data management, and RIPE Atlas users can also perform customised measurements.

A. RIPE Atlas platform

RIPE is the Regional Internet Registry (RIR) for Europe, the Middle East, and parts of Central Asia, whereas RIPE Atlas is RIPE's data collection and active measurement platform.

According to the latest information available at the time of crafting this article [24], RIPE Atlas has more than 12,000 *active* probes worldwide, covering more than 90% of the countries. In Spain, the country to which this study is restricted, the networks that contain RIPE Atlas probes cover more than 92% of the almost 38 million Internet users [25] in the country, and it currently has 224 *active* probes in Spain. For more details on RIPE Atlas architecture, we refer to [26].

RIPE Atlas platform allows its users to carry out `ping`, `traceroute`, DNS, SSL, HTTP and NTP measurements. If, for example, a CSP needs to know whether you can run a certain application — that is, whether the RTT latency is below some threshold — from a specific network in which there is at least one probe, then a simple measurement to the server with the most convenient of the previously mentioned choices would suffice to have a well-informed estimate. RIPE Atlas works on a credit-per-use system, where you can spend the credits that you accumulate by contributing to the collaborative network (by, for example, maintaining a probe in your network) to run customized experiments. It also provides several tools for creating and retrieving measurements; in particular, there exists a command line toolset (`Magellan`) for setting up, streaming, filtering, and querying your measurements, a Python library to manage measurements (`Sagan`), and a Python wrapper around the RIPE Atlas API (`Cousteau`).

B. Measurement campaign details

Next, we describe the details of the conducted measurement campaign. For clarity, we separately explain the different components of the campaign: destination and source probe selection to run the measurements, and the main parameters of the campaign, such as periodicity, size, and duration.

The geographical scope of the campaign is circumscribed to Spain, with the only exception of a few cloud-based vantage points located in France; this exception stems from the fact that some CSPs do not have a cloud region in Spain, and the closest one is the region located in France.

We measure the RTT via `traceroute`. The measurements are initiated from the probes located in the cloud to the selection of end-user RIPE Atlas probes located across Spain, because otherwise `traceroute` packets would be dropped by the cloud infrastructure with high probability.

1) End points (Destination probes representing user locations): RIPE Atlas contains 624 probes located in Spain (in May 2024), 224 of which are active probes. We first filter the probes to restrict ourselves to those with stable connections that also belong to the relevant AS. As previously mentioned, we are interested in analyzing the latency performance of different CSPs for a specific IO, to better understand the corresponding impact of the IO's network and the CSP's network in the perceived latency. Hence, among those 224 probes in Spain, we select those that (i) belong to Telefónica's AS Number (ASN) 3352, (ii) have been online (reachable) at least 99% of the time during the last week before the start of the campaign, (iii) have been online at least 99% of the time during the last month before the start of the campaign, *and* (iv) have been online at least 80% of the time since they were registered in RIPE Atlas. The metrics regarding online time were obtained by scrapping RIPE Atlas' web, since the platform does not allow for retrieving this information from the available APIs.

This filtering leaves us with 36 probes, although 4 of them were finally discarded, as later explained in Section III-C. We recall that this set corresponds to the destination probes that act as end users. The spatial distribution of this set of probes is illustrated in Fig. 1. There are two probes placed in the Canary Islands, which are located 1500 km away from the south tip of the Spanish continental area in a southwest direction. We have pictorially relocated them to the bottom-right corner of Fig. 1 to maintain a high enough spatial resolution. The probes' spatial distribution coarsely illustrates the underlying population density distribution of Spain, mostly condensed in the coastal regions with Madrid and Zaragoza as major exceptions.

2) Source probes (cloud server locations): We make use of 8 cloud-located vantage points to represent cloud server locations. Besides locating them in some of the main CSP platforms, we also consider vantage points directly located within Telefónica's infrastructure: In this manner, we can study the possible differences between CSP's and IO's cloud services. Since the main scope of this work is not to rank or compare different CSPs, but rather characterizing the generic performance that end users can experience when services are located at (any) cloud, we anonymize the name of the

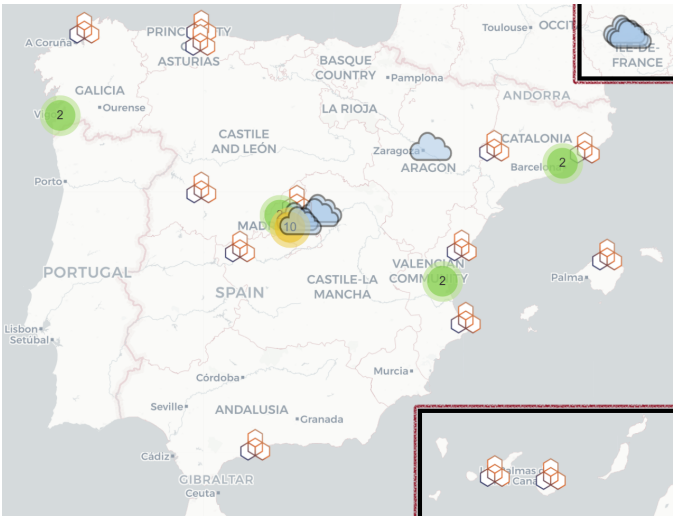

Fig. 1. Geographical distribution of probes taking part in the experiment. The cloud server probes (source probes) are represented by the icon ☁, whereas the destination probes are represented by the RIPE Atlas' icon. Circles with a number inside indicate the number of destination probes that are located in their inner area to avoid overimposing symbols. Canary Islands (bottom right) and the Paris metropolitan area (top right) are not in their actual location.

specific CSP the hosts each probe. Consequently, from this point forward we refer to the three considered leading CSPs as C1, C2, and C3, where the order has been selected arbitrarily. The main characteristics of the vantage points are detailed as follows, where "ID" refers to the probe RIPE Atlas ID.

1) IO_{int}: We installed this probe (ID: 1006162) in one of the datacenters of *Distrito Telefónica*, the headquarters of the IO Telefónica in Madrid, Spain.
2) C1_{io}: This probe is located at the IO's edge as part of C1 deployment, such that the compute and storage cloud services are placed at the edge of Telefónica's network.
3) C1_{sp}: This probe (ID: 1004991) is installed in the cloud region that C1 has deployed in Spain.
4) C1_{fr}: This probe (ID: 1003375) is installed in the cloud region that C1 has deployed in France.
5) $\text{C3}_{\text{sp-std}}$: We installed this probe (ID: 1007394) in the cloud region that C3 has deployed in Spain. It is configured with the basic network service of C3, such that traffic uses peering, ISP, or transit to reach users.
6) $\text{C3}_{\text{sp-prm}}$: We also installed this probe (ID: 1007393) in the cloud region that C3 has deployed in Spain. But, unlike $\text{C3}_{\text{sp-std}}$, it uses C3's premium network service, and thus traffic between the Internet and the cloud application travels within the C3 private network to reach users.
7) $\text{C3}_{\text{fr-prm}}$: We installed this probe (ID: 1007409) in the cloud region that C3 has deployed in France, also configured with the premium network service.
8) C2_{fr}: We installed this probe (ID: 1007405) in the cloud region of C2 sited in Paris area (France).

The spatial distribution of the cloud-based vantage points is also shown in Fig. 1, where we have added on the top-right corner of the figure the location of the France-based cloud probes, which are all closely located in the Paris area.

TABLE I
SUCCESSFUL TRACEROUTE SAMPLES PER DESTINATION PROBE

Probe ID	341	3712	3726	13881	14866	15118	15618	15632	21537	26072	30039	30392	33627	33818	33948	33971	34344	51265
% loses	0%	0.03%	0.01%	78.56%	0.49%	5.89%	5.61%	0%	0.07%	0%	22.78%	1.47%	0.01%	0%	0.16%	2.75%	7.71%	0.08%
Probe ID	51352	52511	55661	60494	60561	61357	61547	62363	62588	62799	1003090	1003970	1004200	1005149	1005642	1007387	1007401	
% loses	0%	6.28%	0.04%	0.31%	0%	0%	1.34%	0.43%	2.03%	0.03%	0%	1.5%	1.22%	0%	0.03%	0.5%	0.04%	

TABLE II
SUCCESSFUL TRACEROUTE SAMPLES PER CLOUD SERVER

Probe ID (cloud)	Cloud label	% loses (35 probes)	% loses (32 probes)
1003375	$C1_{fr}$	3.68%	0.93%
1004991	$C1_{sp}$	4.06%	1.02%
1006085	$C1_{io}$	3.88%	1.01%
1006162	IO_{int}	4.00%	1.01%
1007393	$C3_{sp\text{-}prm}$	4.09%	1.12%
1007394	$C3_{sp\text{-}std}$	4.19%	1.15%
1007409	$C3_{fr\text{-}prm}$	3.96%	1.11%
1007405	$C2_{fr}$	5.80%	1.00%

We hence observe that the set of vantage points contains: (i) from the spatial perspective, 5 probes in Spain (4 in Madrid, one in Zaragoza) and 3 in France (Paris area); (ii) from the ownership dimension, one at Telefónica's network, 3 at C1's platform, 3 in C3, and one belonging to C2. C2 has not (yet) deployed a cloud region in Spain; hence, we consider the closest cloud region: France. We also wanted to analyze the impact of deploying a cloud region in a country, so we also consider the cloud regions located in France for the CSPs that have already deployed cloud regions in Spain (C1 and C3).

3) Measurement setup: We performed ICMP `traceroute` measurements from each one of the 8 cloud-based vantage points towards each of the 36 destination points. Each `traceroute` measurement sends 3 packets, each of them of size 1499 kB, excluding any header for network and transport layer protocols. These measurements were repeated *every 30 minutes* during almost three months: they were initiated on January 30th 2024 and have been consistently running until April 24th 2024. That amounts to a total of more than 4000 `traceroute` measurements per each CSP–end-point pair.

C. Dataset pre-processing and enhancement

Measurement results were retrieved from RIPE Atlas database in JSON format, where they remain publicly available. Besides the information provided in these0 files, we enhance the resulting dataset with additional information that allows us to better characterize the data. We also realize a preliminary evaluation of the results to analyze the connectivity of the probes and remove those that were not correctly functioning.

1) Additional information added to the dataset: To enhance the dataset, we integrated some further features. Specifically, we retrieved all the probe metadata available through RIPE Atlas APIs, which contains information such as spatial coordinates, ASN, IP addresses, and network prefix. We also computed and added the distance (in km) between probes for each one of the 288 source-destination probe pairs (8 CSP sources and 36 end points). Furthermore, to provide a more comprehensive understanding of the `traceroute` experiments, we compute and incorporate the average round-trip time over the 3 packets sent at each measurement, for each hop in each `traceroute`.

2) Stability of the involved end points and drop out: Carrying out a preliminary analysis is essential to prevent corrupted data or wrongly configured tests from compromising the later study.

From the 36 probes used as destinations that we selected, we evaluated their performance in terms of successful `traceroute` measurements during the whole campaign. That is, whether we obtain a response to the `traceroute` command and whether that response corresponds to a timeout. First, we removed one end point because none of the `traceroute` measurements was successful, likely because of a firewall configuration on the destination network. The results for the remaining 35 destination probes are represented in Table I, where we show the percentage of packets that do not receive a response from the final destination for each one of the selected probes. We can observe that there are two probes that were not correctly functioning for a considerable amount of time, namely 22.78% and 78.56%, respectively. After verification in RIPE Atlas platform, we found that those probes were not reachable (offline) for significant periods of time during the measurement campaign, so we removed their data from the dataset. In addition, another probe (ID 52511) was found to return values that range between 2 s and 3 s of latency for about 85% of the samples, even if its percentage of lost packets is in line with many other probes. Consequently, we also removed this probe from the dataset, which was hence composed of 32 probes (and hence 8×32 = 256 source-destination pairs). We note that Fig. 1 only represents those post-filtered 32 probes.

To better understand the impact of these flawed probes, we show in Table II the same metric, the successful measurements, this time aggregated per source cloud server. The metric was computed with the 35 probes of Table I and with the 32 probes left after removing the 3 malfunctioning probes highlighted in Table I. We observe how unsuccessful samples drastically drop for all cloud servers after removing the malfunctioning probes.

IV. CLOUD LATENCY CHARACTERIZATION

After explaining the measurement campaign, we analyze the cloud latency from the obtained measurements. We proceed to characterize the collected information from different perspectives: (i) the temporal analysis over every day, (ii) the probability distribution for each CSP and for each source-destination pair, (iii), the correlation between RTT and distance, and (iv) the variability and statistics of IP paths.

A. Characterization of RTT across time, servers, and users

1) Mean Hourly RTT: We evaluate whether the experienced RTT varies throughout the day. For that, we cluster all the

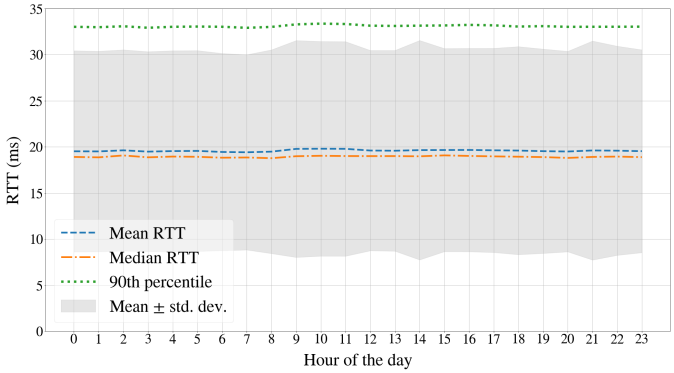

Fig. 2. RTT statistics as a function of the hour of the day, aggregated over all the source and destination probes.

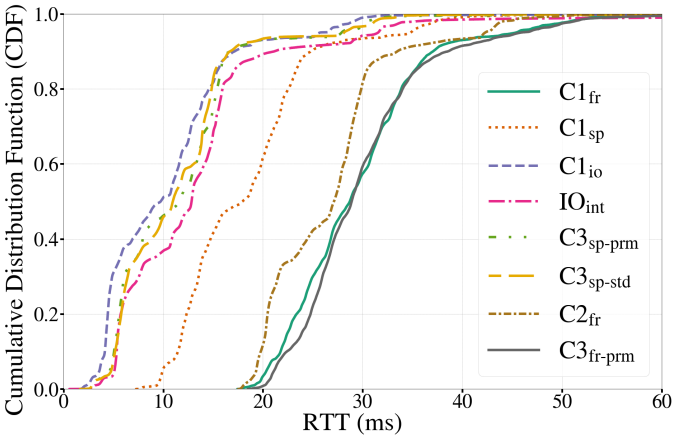

Fig. 3. CDFs of the 8 cloud source probes, aggregating data to all destinations.

RTT measurements based on the hour of the day at which they were performed, and we obtain the main statistics of those clusters over the three months of campaign. The results are shown in Fig. 2, where we depict the mean, median, and 90th percentile of the RTT for each of the 24 hours of the day. In addition, the shadowed area represents the range given by mean $\pm$ std. deviation. We observe that the most important statistics remain remarkably constant across the different hours of the day. This behavior, already pointed out in [4], is also consistent across vantage points. The only visible variation is that the RTT variance subtly increases during peak hours of the day (morning, 14 p.m. and 21 p.m.), but this hike is only of around 1 or 2 ms, which does not alter the overall performance of most of the applications. This behavior is partially explained by the bias of RIPE Atlas networks towards great-connectivity devices, and where wireless devices, for example, are prominently underrepresented. In contrast, cellular wireless networks are more prone to suffer from variable delays due to the variable congestion at the access network [27].

2) Cumulative distribution function of RTT: Next, we focus on the density function of the RTT and analyze it with different granularity. First, we obtain the Cumulative Distribution Function (CDF) of the RTT for each one of the vantage points, aggregating over the 32 destination probes.

The results are shown in Fig. 3, and there are several important insights that we can obtain from this figure. The first observation, which is certainly expected, is that distance-induced delay is not avoidable: indeed, the three servers located in France are the ones with higher RTT; more importantly, this latency gap is quite constant across the CDFs and is approximately of 20 ms, which is a significant increment that can disrupt the QoS of some applications such as Extended Reality (XR). Another finding, perhaps more surprising, is that each of the eight CDFs presents a steep silhouette, where around 90% of the samples lie within a narrow range of 15 ms. This fact indicates that the network presents a quite stable behavior. Other aspect to remark is that the performance for $C3_{\text{sp-prm}}$ and $C3_{\text{sp-std}}$ is basically equivalent. Actually, previous works [13] found that, for some CSPs, public-network-through plans demonstrated higher variability in latency compared to the premium CSP-private-network counterpart, but this conclusion cannot be replicated with our measurements; conversely, it seems that both tiers achieve an equivalent performance in terms of variability for C3. We also observe from Fig. 3 that the best performance is obtained by $C1_{\text{io}}$, i.e., the only edge-based vantage point. Although the difference concerning IO_{int}, $C3_{\text{sp-prm}}$ and $C3_{\text{sp-std}}$ is not significant, $C1_{\text{sp}}$ under-performs with respect to the other cloud points located in Spain.

We dig into the details of the distribution of the RTT for each of the source-destination probe pairs. In Fig. 4, we represent the per-pair CDFs for the 32 destinations of 4 different cloud vantage points: IO_{int}, $C1_{\text{io}}$, $C2_{\text{fr}}$, and $C3_{\text{fr-prm}}$.. For the sake of readability, we highlight 3 CDFs in each plot: We calculate the mean RTT of the 32 pairs, we sort the destination probes by mean RTT in ascending order, and we select those probes located in the 10th, 50th, and 90th percentile of mean RTT. This visualization has a twofold objective: (i) facilitate the visualization of single CDFs, and (ii) show the narrow range at which most of those CDFs are.

We observe several remarkable behaviors in Fig. 4. First, the individual CDFs are even steeper than the aggregated ones in Fig. 3, which implies that the stable behavior previously mentioned is much more stable when we look into any specific pair. Second, some pairs present a step-wise shape, as the 10th percentile for $C2_{\text{fr}}$. Yet, such steps are distributed in a very narrow range of around 5 ms, which means that those steps are not impacting considerably the performance of the network. Besides this, most of the destination probes are concentrated in a close range of 15 ms, which aligns with the results highlighted for Fig. 3. We can distinguish two probes that present a distinct behavior, whose CDF is shifted towards the high RTT values; those probes are the two probes located in the Canary Islands, which reproduce the same problem seen for the cloud regions located in France: distance plays an important role.

The latter comment implies a consequence that is generally applicable for other areas of the world and IOs: In regions like the Canary Islands, the edge paradigm is necessary to deliver services that require less than 40 ms of network latency.

B. Characterization of RTT as a function of the distance

Next, we aim to answer the question about whether the fundamental assumption that distance is the possibly most

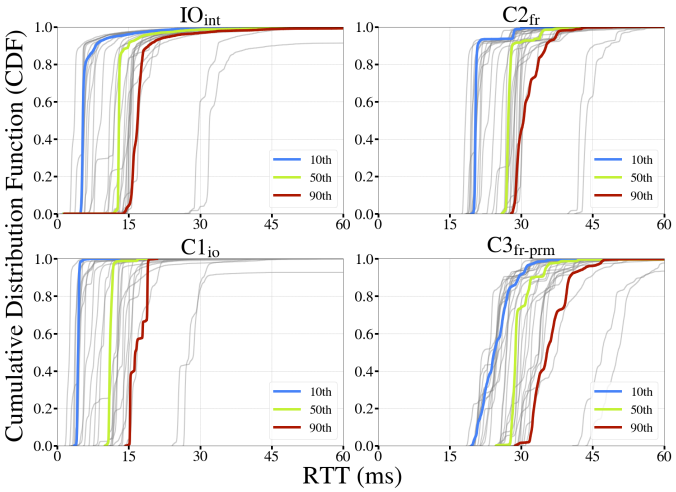

Fig. 4. CDF for all pairs of source and destination probes, represented for four source cloud servers. The highlighted CDFs correspond to the destination probes that occupy the 10th, 50th and 90th percentile of mean RTT among the 32 probes, i.e., the green lines (50%) is the CDF of the probe whose mean RTT is above the mean RTT of the 50% of the probes.

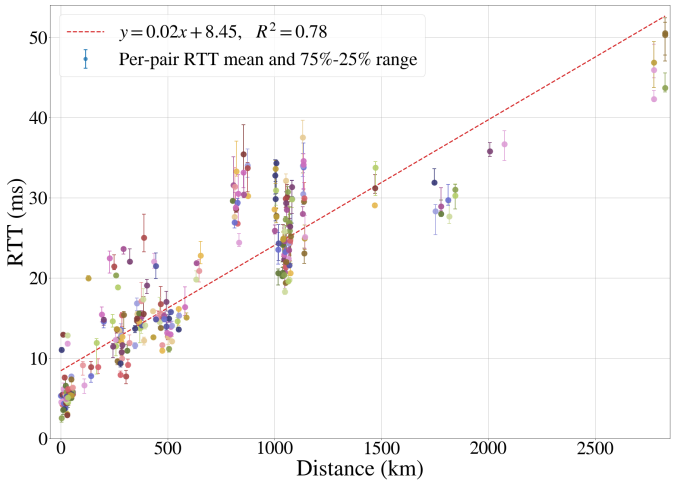

Fig. 5. RTT statistics for all the source-destination pairs as a function of the distance. Colors are randomly selected to distinguish close points and density.

important element impacting latency between a cloud user and the cloud is true. To ascertain whether a direct correlation exists between RTT and distance, we model the relationship between these two magnitudes as a linear regression. We first compute the mean RTT for each source-destination pair. Then, we apply the method of least squares to estimate the best fitting parameters of the linear regression. The results are depicted in Fig. 5. In this figure, we locate the median and the interquartile range of the RTT for each of the 256 pairs, as a function of the distance in kilometers. The fitting line indicates that RTT increases about 2 ms per 100 km (i.e., 20 ms per 1000 km), with an additive minimum delay of 8.45 ms. This result is aligned with the gap of 15 ms appearing between the cloud points located in Madrid and those located in Paris, and may partially justify the under-performance of C1$_{sp}$ with respect to the other cloud points located in Spain, due to the concentration of destination probes in Madrid area and the fact that C1$_{sp}$ is

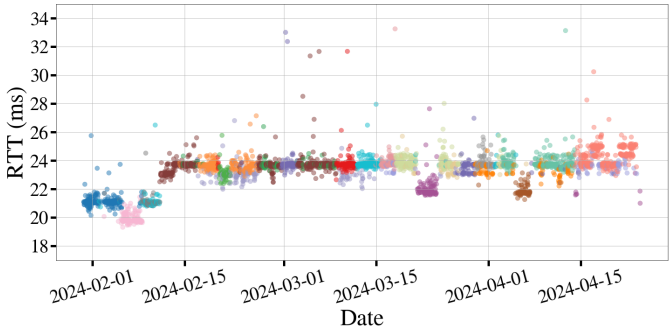

Fig. 6. RTT per IP path from C1$_{sp}$ to destination probe with ID 1003090.

located in Zaragoza, 350 km away from the capital of Spain.

A coefficient of determination $R^2 = 0.78$ suggests that there is a clear linear correlation, yet there are other factors (and randomness) that also notably impact the RTT. This finding motivates us to conduct additional analysis to fully comprehend the main aspects driving user-to-cloud latency.

C. Impact of IP path changes

We have seen that distance fails to faithfully determine the mean RTT, even if there exists an important correlation between the two variables. Therefore, we focus on analyzing whether the specific IP path used (i.e., the hops that the measurement traverses) yields deeper insights on the RTT. We zoom in on a specific source-destination pair due to space constraints. We select the measurement from C1$_{sp}$ to probe 1003090.

We show the results of this analysis in Fig. 6 and Fig. 7. In Fig. 6, we represent the RTT of each measurement for the selected pair as a function of the time at which the measurement was realized. We indicate the distinct IP paths with different colors. We observe that there exists certain stability of the most frequent routes, because a different route assumes the leading role every few days. We also note that the specific IP path selected influences the RTT. We quantify such influence by obtaining the main statistics for the most frequent routes, which is illustrated in Fig. 7. There, we represent all the IP paths that are used for at least 1% of the samples, and the paths are sorted by increasing percentage of samples. We can clearly observe that the RTT variability per path is considerably smaller than the variability of all paths jointly considered. Numerically, the standard deviation of the RTT over all samples is 1.3 ms, while the mean standard deviation averaged over paths is less than half: 0.63 ms. Thus, the specific IP path does impact the latency, and path stability can reduce RTT variability.

V. Cloud Latency Forecasting

After characterizing the RTT performance, we aim to answer (i) if we can forecast the future connectivity performance based on the previous characterization to know in advance the RTT, (ii) what are the most suitable models for this task, and (iii) if we can infer which features are more important.

We consider two different scenarios: *time series forecasting*, i.e., predicting the future behavior of an already existing link, and *spatial forecast*, i.e., infer the performance of a connection *before it actually exists* based on other connections.

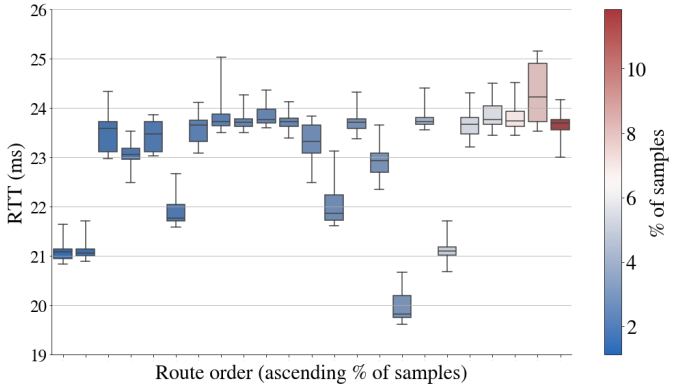

Fig. 7. Statistical distribution of the RTT for distinct IP paths from $C1_{sp}$ to destination probe with ID 1003090. Boxes represent the median and the inter-quartile range, whiskers indicate the 5th and 95th percentile, and color shows the percentage of samples that use the corresponding IP path.

A. Time series forecasting

We first re-structure the dataset as time series of RTT values, one series for each source-destination pair. We employ a range of well-established forecasting methods. Namely, *Naive forecast*, which simply matches the prediction to the latest available RTT value, *simple exponential smoothing (SES)*, *double exponential smoothing* (Holt), also known as Holt's linear, *Auto-ARIMA* (AutoRegressive Integrated Moving Average), which automatically identifies the optimal parameters for an ARIMA model, Facebook's *PROPHET*, and *Long Short Term Memory (LSTM) neural networks*, which are a type of recurrent neural network (RNN) designed to handle sequential data, making them well-suited for time series prediction tasks. We note that *simple exponential smoothing* can be also expressed as an ARIMA(0,1,1) without constant model, and *double exponential smoothing* is equivalent to an ARIMA (0,2,2) without constant model.

The performance of these methods is summarized in Table III, where we provide both the unitless normalized mean squared error (MSE) and the actual MSE in milliseconds. We first observe that naive forecasting, SES, Holt, and ARIMA give similarly accurate predictions. Conversely, Prophet's and LSTM's performance is one order of magnitude less accurate. This is at first surprising, as those are the most advanced algorithms, but it can be explained by the results of Section IV: Latency is quite constant, with spare leaps that are not always predictable. Thus, applying simple methods that maintain the main trends can provide better results, since complex models aim at estimating complex (sometimes nonexistent) patterns. A careful tuning of Prophet and LSTM could provide better performance. However, it is not our intend to perform such optimization, but rather highlight that simple methods, which do not require this optimization, achieve a high accuracy.

B. Spatial forecasting

After evaluating the potential of time-series forecasting methods, we perform analysis on spatial forecasting. Specifically, we ask ourselves whether we can predict the performance of a new not-yet-established link from data obtained from other links. There exist many forecasting methods that we can use (we refer to [28] for a complete description of ML algorithms according to how interpretable their solutions are). In light of this, we evaluated several machine-learning methods that are suitable for our problem. We consider six forecasting supervised-learning methods: Linear model, Decision Trees, Random Forests, Extreme Gradient Boosting (XGBoost), K-nearest neighbors (KNN), and Support Vector Machines (SVM). We analyze two scenarios: (i) a case where distance is the sole feature considered for forecasting, and (ii) a case where all features from the dataset are considered for the prediction.

TABLE III
TIME-SERIES FORECASTING RESULTS

Method	Loss (NMSE)	Loss (ms)
Naive forecast	0.074	0.792
Simple exponential smoothing	0.074	0.792
Double exponential smoothing	0.085	0.901
Auto-ARIMA	0.064	0.678
PROPHET	0.411	4.366
LSTM	0.778	8.274

TABLE IV
SPATIAL FORECASTING RESULTS

	Only Distance		All-features	
Method	NMSE	Loss (ms)	NMSE	Loss (ms)
Linear model	0.533	5.673	0.413	4.401
Decision tree	0.465	4.961	0.496	5.291
Random forest	0.462	4.923	0.355	3.780
XGBoost	0.464	4.948	0.313	3.334
SVM	0.556	5.923	0.431	4.590
KNN	0.486	5.183	0.475	5.054

The results are summarized in Table IV. We observe that the distance-only prediction is quite accurate, which aligns with the characterization of Section IV. The MSE remains close to 5 ms for all the considered algorithms, which is a good precision if we consider that both CSPs and IOs are mostly interested on knowing if the RTT belongs to a certain range of values. In addition, we also observe that adding the other features in the dataset allows us to further improve the forecasting accuracy, as we are able to reduce the RTT error down to 3.3 ms with XGBoost, which is the most accurate algorithm.

We also show the error PDF for the six considered methods in Fig. 8. We can learn more about the model's performance from the PDF shape: the models' narrow distribution around zero implies that error variance is minimal, and estimates are at most 10 ms far from the actual value with high probability.

C. Feature importance: Applying interpretable methods

We look into the key features affecting latency by employing explainable ML techniques. These techniques are often applied after the model is trained to gain insights of what the model has learnt, they are also known as "post-hoc" explainable approaches. In this work, we shall use Local Interpretable Model-agnostic Explanations (LIME) [29] and Shapley Additive Explanations (SHAP) [30] approaches, which

TABLE V
LIME AND SHAP BASED EXPLANATIONS FOR XGBOOST AND RANDOM FOREST ALGORITHMS

	XGBOOST				RANDOM FOREST			
	LIME		SHAP		LIME		SHAP	
Order	*Feature*	*Weight*	*Feature*	*Weight*	*Feature*	*Weight*	*Feature*	*Weight*
1	Distance	0.797	Distance	0.758	Distance	0.750	Distance	0.770
2	Source probe ASN	0.053	Source probe ASN	0.076	Hop count	0.036	Source probe ASN	0.060
3	Hop count	0.039	Hop count	0.051	Source probe ASN	0.030	Hop count	0.052
4	Destination reached	0.013	Timestamp	0.042	Minute of hour	0.023	Timestamp	0.034
5	Minute of hour	0.009	Destination reached	0.038	Timestamp	0.014	Last synchronization	0.015

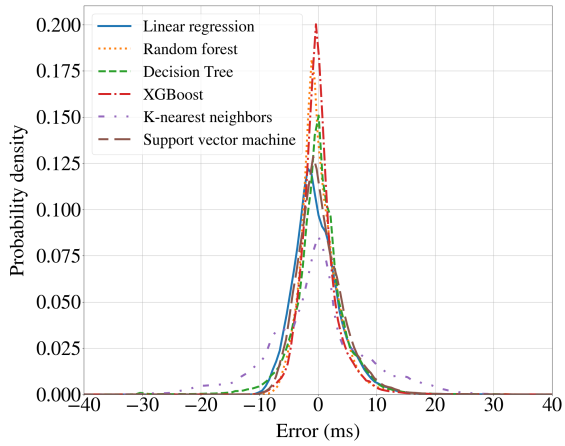

Fig. 8. PDF of the forecasting errors for each one of the considered algorithms.

are the most prevalent "post-hoc" explainable approaches. LIME focuses on creating local explanations for individual predictions, while SHAP is a model-agnostic approach used for generating explanations on samples of predictions.

To use LIME, we select an instance from the data that requires explanation. LIME is used to perturb the data sample to produce fresh perturbed samples surrounding the instance of interest. The updated data is then fed into the black-box model, allowing it to generate predictions based on these new examples. The samples are then ordered by degree of similarity to the original sample. LIME constructs a linear model for these weighted samples using the black box predictions on the altered data as targets. The features with the greatest impact on the final prediction can then be determined via LIME.

In contrast, SHAP applies a game-theoretic method. The feature importance is determined by calculating the marginal contribution of each feature, taking into account every scenario in which that specific feature could have been added to the set of features that the model was trained on, as well as the model prediction for each of these feature combinations. Subsequently, the average contribution of each feature to the prediction over all feasible feature combinations is its Shapely value.

Explaining forecast decisions: We evaluate both LIME and SHAP to determine which attributes eventually impacted the most the RTT forecast of the two best models: XGBoost and Random forest. We perform twenty runs for each one of the experiments, and we average the results over the runs. For LIME, we choose instances at random, whereas for SHAP we choose random groups of samples. We show the results in Table V, where we present the top five features sorted by their weight in the forecast explanation.

SHAP and LIME weights are unitless. They indicate the degree to which each feature influences the model's output. These values can be negative or positive, and the sign denotes whether the impact is proportional or inverse. As we are interested only in the magnitude of the impact, we provide the average *absolute* value over the twenty runs.

We observe in Table V that both explainable methods show a similar result for any of the two methods: The distance between the source and the destination of the connection is the most important feature with a weight that lies between 0.75 and 0.797 for the four considered cases, whereas the weight of the second feature is only of 0.05–0.07. For the second to the fifth features, we observe that the weights are closer to each other, and we find different features in this top-five list. The second and third most important features are always (but with alternating order) the source probe's (i.e., the CSP's) ASN and the hop count. The CSP's ASN is thus important but only to a limited extent, which tells that different providers are well and smoothly connected, whereas the hop count indicates that stabilizing paths and minimizing hops is also important to provide adequate service, as also illustrated in Fig. 6.

VI. DISCUSSION AND COMPARISON

This work adds an extensive campaign to the existing measurement research to help the community to pave the way towards a more comprehensive understanding of latency performance of cloud systems. Next, we discuss our main findings and we put them into perspective with previous works.

Measurement campaigns realized on the public Internet to monitor cloud latency performance have many degrees of freedom in terms of parameter selection. Since resources are limited, there exists a compromise between the number of source points, destination points, geographical resolution, temporal resolution, service providers, countries, technologies, and other aspects. Thus, comparing findings of different works is often complex because the selected parameters differ. For example, while [5], [20] are the works whose analysis is the closest to ours, there are still significant differences: They sacrificed temporal resolution (`ping` every 3 hours and `traceroute` every 24 hours) for the sake of geographical scope (101 cloud regions over the world) and temporal duration,

whereas we improve the temporal granularity to 30-minute intervals but at the expense of limiting it to a single country. In terms of spatial resolution, they considered 32 probes in Spain, which matches the number of probes that we consider; yet, they presented a country-level analysis, whereas we present a more detailed per-pair and per-route analysis, and we also provide forecasting and explainable methods to better comprehend the complex relationship between features and latency. We also find important setup differences with other works such as [4] (25 end-user vantage points, 8 cloud datacenters, 14 days, `ping` packets), [13] (from 6 GCP regions to 458 speed test servers, focusing on throughput rather than latency), or [16] (30 datacenters in U.S.A., focused on edge deployment).

Our findings highlighted in Fig. 2 and Fig. 3 align with those of [4], which found that there are no clear temporal patterns and that latency was more influenced by geographical regions than by the choice of provider. However, we also obtain results that deviate from previous works. For example, [13] found that, for some CSPs, public-network-through plans demonstrated higher variability in latency compared to the premium CSP-private-network counterpart, but this conclusion cannot be replicated with our measurements; Fig. 3 shows that standard and premium plans achieve the same performance, with matching CDFs. This illustrates how measurement campaigns provide snapshots of latency behavior, and we must continue and expand the research on this area.

VII. Conclusions

We have performed a measurement campaign to characterize the latency between some of the major CSPs and end users connected to the network of one of the main IOs in Spain. We made use of RIPE Atlas platform to realize this campaign that consisted of traceroute measurements every thirty minutes and for 256 different cloud-user pairs. From this dataset, we have characterized the RTT behavior of cloud service platforms, showing that RTT does not change over the duration of the day. We also proved that distance is the most important feature that determines the mean RTT, although it does not suffice to explain all the RTT variability. In addition, we observe that placing the cloud servers at the edge of the IO's network can improve the latency, although the values remain comparable. This work contributes to the continuous task of understanding cloud-based services' performance and provides insights on what are the key features to take into account for latency characterization and forecasting.

Acknowledgements

We acknowledge the RIPE Atlas team for providing us access to their platform, and Christian Elsen for maintaining the public RIPE Atlas probes located at different AWS regions [31].

References

[1] T. Koch, W. Jiang, T. Luo *et al.*, "Towards a traffic map of the internet: Connecting the dots between popular services and users," in *Proc. ACM Workshop on Hot Topics in Networks (HotNets)*. ACM, 2021, p. 23–30.

[2] P. Casas and R. Schatz, "Quality of Experience in Cloud services: Survey and measurements," *Computer Networks*, vol. 68, pp. 149–165, 2014.

[3] A. Li, X. Yang, S. Kandula, and M. Zhang, "Cloudcmp: Comparing public cloud providers," in *Proc. ACM Internet Measurement Conf. (ACM IMC)*. ACM, 2010, p. 1–14.

[4] F. Palumbo, G. Aceto, A. Botta *et al.*, "Characterization and analysis of cloud-to-user latency: The case of Azure and AWS," *Computer Networks*, vol. 184, p. 107693, 2021.

[5] N. Mohan, L. Corneo, A. Zavodovski *et al.*, "Pruning edge research with latency shears," in *Proc. ACM Workshop on Hot Topics in Networks (HotNets)*. ACM, 2020, p. 182–189.

[6] F. Michelinakis, H. Doroud, A. Razaghpanah *et al.*, "The cloud that runs the mobile internet: A measurement study of mobile cloud services," in *IEEE Conf. Computer Commun. (INFOCOM)*, 2018, pp. 1619–1627.

[7] Y. Jin, S. Renganathan, G. Ananthanarayanan *et al.*, "Zooming in on wide-area latencies to a global cloud provider," in *Proc. ACM SIGCOMM Conf.* ACM, 2019, p. 104–116.

[8] "RIPE Atlas," https://atlas.ripe.net, Accessed April 2024.

[9] Y. A. Wang, C. Huang, J. Li, and K. W. Ross, "Estimating the performance of hypothetical cloud service deployments: A measurement-based approach," in *IEEE Conf. Computer Commun. (INFOCOM)*, 2011, pp. 2372–2380.

[10] O. Tomanek and L. Kencl, "CLAudit: Planetary-scale cloud latency auditing platform," in *Proc. IEEE Int. Conf. Cloud Networking (CloudNet)*, 2013, pp. 138–146.

[11] "Planetlab," https://planetlab.cs.princeton.edu, [accessed April 2024].

[12] O. Tomanek, P. Mulinka, and L. Kencl, "Multidimensional cloud latency monitoring and evaluation," *Computer Networks*, vol. 107, Oct. 2016.

[13] R. K. P. Mok, H. Zou, R. Yang *et al.*, "Measuring the network performance of google cloud platform," *Proc. ACM Internet Measurement Conf. (ACM IMC)*, 2021.

[14] O. Alay, A. Lutu, R. García *et al.*, "Measuring and assessing mobile broadband networks with MONROE," in *IEEE Int. Symp. on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2016.

[15] R. Fontugne, A. Shah, and K. Cho, "Persistent last-mile congestion: Not so uncommon," in *Proc. ACM Internet Measurement Conf. (ACM IMC)*. ACM, 2020, p. 420–427.

[16] L. Corneo, N. Mohan, A. Zavodovski *et al.*, "(how much) can edge computing change network latency?" in *IFIP Networking Conf.*, 2021.

[17] M. Candela, E. Gregori, V. Luconi, and A. Vecchio, "Using RIPE Atlas for geolocating IP infrastructure," *IEEE Access*, vol. 7, 2019.

[18] O. Victor Babasanmi and J. Chavula, "Measuring cloud latency in Africa," in *Proc. IEEE Int. Conf. Cloud Networking (CloudNet)*, 2022, pp. 61–66.

[19] A. Kedia, A. Ganesh, and A. Aggarwal, "Examining lower latency routing with overlay networks," 2023, arxiv preprint: 2306.15174.

[20] L. Corneo, M. Eder, N. Mohan *et al.*, "Surrounded by the clouds: A comprehensive cloud reachability study," in *Proc. Int. World Wide Web Conf. (WWW)*. ACM, 2021, p. 295–304.

[21] V. Bajpai, S. J. Eravuchira, and J. Schönwälder, "Lessons learned from using the RIPE Atlas platform for measurement research," *ACN SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 3, p. 35–42, jul 2015.

[22] P. Sermpezis, L. Prehn, S. Kostoglou *et al.*, "Bias in internet measurement platforms," in *Netw. Traffic Meas. and Analysis Conf. (TMA)*, 2023.

[23] "RIPE NCC," https://www.ripe.net/, online; accessed 16-April-2024.

[24] RIPE Atlas, "Coverage and statistics," https://atlas.ripe.net/coverage/, 2024, [Online; accessed 01-May-2024].

[25] E. A. Petros Gigis, Vasileios Kotronis, "RIPE Atlas population coverage," https://sg-pub.ripe.net/petros/population_coverage/, 2024, [Online; accessed 01-May-2024].

[26] R. Kistel, "RIPE Atlas architecture - how we manage our probes," https://labs.ripe.net/author/kistel/ripe-atlas-architecture-how-we-manage-our-probes/, 2023, online; accessed 01-May-2024.

[27] A. F. Zanella, A. Bazco-Nogueras, C. Ziemlicki, and M. Fiore, "Characterizing and modeling session-level mobile traffic demands from large-scale measurements," in *Proc. ACM Internet Measurement Conf. (ACM IMC)*, 2023, p. 696–709.

[28] C. Rudin, C. Chen, Z. Chen *et al.*, "Interpretable machine learning: Fundamental principles and 10 grand challenges," *Statistics Surveys*, vol. 16, pp. 1–85, 2022.

[29] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[30] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Inf. Processing Systems*, vol. 30, 2017.

[31] C. Elsen, "RIPE Atlas probes in AWS," https://github.com/chriselsen/RIPE-Atlas-in-AWSs, [Online; accessed 15-Jan-2024].