Explainable AI to Understand the Latency Behavior of Public Cloud Service Platforms

Rita Ingabire, *Student Member, IEEE,* Antonio Bazco-Nogueras, *Member, IEEE,* Vincenzo Mancuso, *Member, IEEE,* Luis M. Contreras, *Member, IEEE,* Jesús Folgueira, *Member, IEEE*

***Abstract*—Cloud platforms have become a core component of the Internet because most services and products rely on them to host their backends. Estimating and understanding the latency experienced when accessing those cloud platforms is a challenge of growing importance that has not been sufficiently studied. To address this relevant matter, we conducted a three-month measurement campaign, collecting traceroute data every 30 min across 256 source–destination probe pairs. Our specific goal is to analyze whether the current network is able to provide adequate performance for emerging applications and services. We use this dataset to evaluate the performance of forecasting algorithms when predicting cloud latency from both temporal and spatial perspectives, and we leverage post-hoc explainability methods to identify the drivers affecting latency. Several prior studies provide public cloud-latency datasets, but these datasets are generally analyzed in isolation. To close this gap and provide a cross-dataset comparative analysis of cloud-latency measurements, we analyzed the related publicly available datasets and applied a common forecasting and explainability workflow to compare their findings. Our analysis reveals that operators do not require complex methods to predict latency and that distance and a few other simple features are sufficient to achieve operationally accurate predictions. We find latency to be remarkably stable from the user's perspective, both over the duration of the campaign and across hours of the day, which contrasts with previous findings, and we show that the specific path traversed has a significant impact on latency.**

***Index Terms*—explainability, cloud, latency, RIPE Atlas, forecasting, comparative analysis, measurement, LIME, SHAP**

I. Introduction

SOCIETY deeply relies on services that run on remote datacenters. Internet access often provides seamless connectivity that enables these remote services. However, next-generation services demand very stringent technical requirements, mainly in terms of latency and throughput, and current networks may fall short of these requirements. Therefore, the development of these emerging latency-sensitive applications demands a thorough understanding of cloud access time.

Latency is one of the most influential factors affecting user Quality of Service (QoS) in online applications. However, characterizing network latency between cloud servers and users is a challenging task, because this latency depends on user location, server distance, access technology, and network topology, among other aspects. Several studies have contributed to the characterization of cloud-service latency [2]–[9], although, naturally, they are constrained by the limitations of measurement campaigns (e.g., temporal or geographical span), and only partial assessments can be obtained [10].

In this work, our objective is to characterize, predict, and explain the time it takes for a geographically distributed population to reach the cloud. To achieve that, we first conducted an extensive measurement campaign of the cloud ecosystem in Spain, which is a mid-sized, developed country whose characteristics may be representative of other regions. We have considered three main Cloud Service Providers (CSPs): Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). The measurements are conducted using the RIPE Atlas platform [11], and they consist of `traceroute` measurements from probes located in these CSPs toward probes that are connected to the network of Telefónica, one of the main Infrastructure Operators (IOs) and Internet service providers (ISPs) in Spain. The decision to restrict the analysis to a specific ISP serves two purposes: first, it minimizes the potential influence of diverse network planning strategies on the measurements; second, Telefónica is the operator with the largest set of probes connected to RIPE Atlas, thereby enhancing the quality of the derived results. We performed a multi-ISP validation campaign to assess the ISP-specific effect. Our approach is non-cooperative since it does not require a privileged vantage point or access privileges within the ISP's infrastructure, and it employs active probing.

Using the collected data, we characterize cloud latency, forecast its temporal and spatial behavior, and explain the importance of the forecasting features. We apply the same analysis to relevant public datasets [7]–[9], providing the first cross-dataset comparative analysis of cloud-latency measurement campaigns. Our main contributions are as follows:

- We release a 3-month traceroute campaign covering 256 links between RIPE Atlas and cloud vantage points for AWS, GCP, and Azure, with 30-min granularity. This dataset enriches public knowledge of cloud latency performance and provides an updated snapshot of it.
- We characterize the Round Trip Time (RTT) distribution across these links in both temporal and spatial dimensions, examining how time of day, distance, and routing

R. Ingabire, A. Bazco-Nogueras and V. Mancuso are with IMDEA Networks Institute, Madrid, Spain. R. Ingabire is also with Universidad Carlos III de Madrid, Spain, and V. Mancuso is also with the University of Palermo, Italy. L. M. Contreras and J. Folgueira are with Telefónica, Madrid, Spain.

This work was partly funded by TUCAN6-CM (TEC-2024/COM-460), funded by the Madrid Regional Government (ORDEN 5696/2024). A. Bazco-Nogueras was supported by the grant 2020-T2/TIC-20710 for Talent Attraction, funded by Madrid Region, Spain.

A preliminary version of part of the work presented in this article appeared in the proceedings of ICCCN 2024 [1].

path each influence latency. Our results reveal that latency is largely stable over time, driven primarily by physical distance rather than time of day, and that deploying cloud access at the IO's edge achieves lower latency than mainstream CSP deployments.
- We evaluate cloud latency forecasting under (i) prediction of future latency from historical data, and (ii) spatial regression for unseen links. We show that the inherent stability of latency enables even simple algorithms to achieve operationally accurate predictions.
- We perform an extensive post-hoc Explainable Artificial Intelligence (XAI) analysis to understand the entanglement among the considered features. To generalize our measurement campaign findings, we provide the first-of-its-kind cross-dataset comparative analysis of public datasets on cloud latency measurements [7]–[9]. This comparison shows some, but not complete, consistency across datasets, with distance and (to a lesser degree) user features being the most relevant factors. We observe that latency variability becomes significant only when considering several countries and ISPs.
- We release the repository that contains the code needed to retrieve our measurements and to perform all the analyses shown in this work to allow for full reproducibility [12].

A preliminary version of this work was presented in [1], where we presented the characterization and regression analysis of the data collected during our measurement campaign. We have thoroughly extended the analysis by (i) providing new insights on the impact of path variability on the latency, and (ii) performing a detailed comparison with all the relevant datasets available in the literature on the topic of cloud latency.

The remainder of this paper is organized as follows. Section II reviews cloud-latency measurements, tools, and public datasets. Section III describes the measurement methodology, and Section IV presents the latency characterization. Section V evaluates temporal and spatial forecasting and explainability. Section VI applies this analysis to public datasets, while Section VII discusses the resulting cross-dataset comparative analysis. Section VIII concludes the paper.

II. Related Work on Public Cloud Measurements

We summarize the state of the art on related topics, and we describe the related datasets, which will later be evaluated in Section VI-C. The discussion of findings and replicability of previous studies will be presented in Section VI-A.

A. State of the art

1) Tools & platforms for cloud measurements: Cloud measurements have previously been performed using custom tools or platforms. CloudBeacon [13] was a JavaScript tool that collected measurements when users accessed web services. These measurements help CSPs evaluate potential changes to their edge networks and identify suitable locations for satellite datacenters. Another tool, CLAudit [14], was built on PlanetLab [15], a distributed measurement platform that operated from 2002 to 2022. Using PlanetLab nodes as distributed vantage points, CLAudit measured latency toward Azure cloud services. It was subsequently used to compare measurements from two one-month campaigns and investigate suspicious latency events [16]. A later effort introduced CLASP [17], which combined cloud and speedtest infrastructure. Over five months, it measured connectivity from GCP to Ookla, MLab, and Comcast speedtest servers and used latency and throughput variability to detect network congestion. Other distributed infrastructures supporting Internet measurements include MLab [18], BISmark [19], and CAIDA Ark [20]. Early studies relied on internally developed custom tools [13], [14] due to the limited availability of cloud measurement platforms, while some more recent studies [17] relied on external measurement services. Therefore, replicability and generalizability remain challenging tasks.

2) Network analysis through RIPE Atlas: RIPE Atlas [11] is a prominent platform for worldwide Internet measurements. It has supported studies of last-mile congestion [22], edge server placement [21], IP infrastructure geolocation [7], and the scarcity of datacenters in Africa [25]. Mohan *et al.* examined whether edge computing is necessary to satisfy the latency requirements of modern applications [8]. They employed over 3200 RIPE Atlas probes distributed across 166 countries as measurement vantage points. The study considered seven CSPs and performed `ping` measurements every 3 hours. Corneo *et al.* extended this work by incorporating `traceroute` measurements every 24 hours [9]. These large-scale studies aggregate measurements across diverse geographies, operators, and cloud providers, making it difficult to separate the effects of distance, operator infrastructure, and temporal variation. They also emphasize descriptive analysis rather than prediction or the systematic identification of latency drivers, and no empirical synthesis and comparison of cloud latency measurements using a common methodology exists. We address these gaps through a homogeneous single-operator design and a common forecasting and explainability analysis of our measurements and public datasets, allowing us to identify latency drivers and assess model generalizability. Tools such as RIPE Atlas also have some limitations [26], [27]. While RIPE Atlas comprises thousands of probes, it naturally presents some bias: many probes enjoy favored network connectivity because of the types of entities and individuals that install them [26], and there are geographic biases toward certain countries and urban locations [27]. Furthermore, it is intrinsically complex to quantify such bias [27].

3) Latency Forecasting and XAI for cloud services: For large-scale traffic forecasting, Koumar *et al.* found that convolutional Long Short-Term Memory (LSTM) hybrids performed best for ISP traffic and observed no strong seasonality [28]. Existing XAI research focuses on operational metrics rather than user latency. At the network level, Moghadas *et al.* used XAI to identify the base stations that most influenced spatio-temporal traffic forecasts and to produce compact explanations [29]. In cloud environments, Gebreyesus *et al.* used SHapley Additive exPlanations (SHAP) to explain datacenter energy use, cooling, and environmental behavior [30], while Viradia *et al.* compared SHAP and Local Interpretable Model-agnostic Explanations (LIME) for optimization across cost, workload, and performance [31]. Sim *et al.* applied XAI to

TABLE I
PUBLICLY AVAILABLE CLOUD-REACHABILITY DATASETS

Ref. Name	Year	Paper	Purpose of study	Platform	Cloud Service Provider	Granularity
CandelaAcc19	2019	[7]	Geolocating IP Infrastructure	RIPE Atlas	RIPE Atlas & NLNOG Ring	1 hour, 2434 probes, 825 targets, 97 countries, 425 cities
MohanHNets19	2020	[8]	Edge computing vs. pure cloud computing	RIPE Atlas	GCP, AWS, Azure, Digital Ocean, Linode, Alibaba and Vultr	Ping every 3 hours, 9 months, worldwide, 3200+ probes, 101 cloud points
CorneoWebC21	2021	[9]	Suitability of modern cloud environments for current and predicted applications	RIPE Atlas	AWS, GCP, Azure, Digital Ocean, Alibaba, Vultr, Linode, Oracle, IBM	3-hour ping, 24-hour traceroute, 8500 probes to 189 cloud targets
ComNet21	2021	[3]	Investigate the cloud-to-user latency	PlanetLab	AWS, Azure	Every min., 25 vantage points, 4 cloud regions, 14 days, ICMP, TCP, HTTP
IFIPNet21	2021	[21]	Placement of general-purpose edge computing resources on Internet-wide scale	RIPE Atlas	Amazon, Microsoft, Google, Digital Ocean, Linode, Vultr and Alibaba	934 probes, 30 datacenters, only U.S.; periodicity not indicated
IMC20	2020	[22]	Investigate last-mile congestion in public clouds	RIPE Atlas	None	30-min intervals, 216 samples per probe, several long time periods
Conext23	2023	[23]	Digital Divide Study	RIPE Atlas	AWS, Azure, GCP	Traceroute 4 times every 3 hours, 4 days
TMA25	2025	[24]	Latency from users to CDN edge servers	RIPE Atlas	CDN edge servers	Ping every 2 hours, 10 days, 79 src, 100 targets

input selection for energy consumption forecasting [32].

Together, these studies demonstrate the value of XAI for analyzing network and cloud systems. However, none jointly applies forecasting and explainability to latency between users and cloud services or evaluates the consistency of explanations across measurement campaigns. In summary, we address these gaps by (a) conducting a 3-month measurement campaign that provides an updated snapshot of cloud service latency; (b) characterizing its spatio-temporal behavior and predictability; (c) systematically applying explainability methods to identify important features; and (d) providing a cross-dataset comparative analysis.

B. Publicly Available Datasets

Table I summarizes the related publicly available datasets. We provide a detailed comparison between our campaign and the most relevant existing ones in Section VI-A.

1) `MohanHNets19` [8]: Mohan *et al.* created this dataset to compare edge and pure cloud computing. They collected worldwide `ping` measurements every 3 hours from more than 3200 probes to 101 datacenters operated by GCP, AWS, Microsoft Azure, Digital Ocean, Linode, Alibaba, and Vultr, beginning in September 2019. They released the cleaned dataset in MySQL format.[1]

2) `CandelaAcc19` [7]: This dataset originated from a study that aimed to geolocate Internet backbone infrastructure. This study used RIPE Atlas anchors and NLNOG ring nodes [33] as destinations representing the Internet infrastructure. It provided one dataset with RIPE Atlas anchors as source points to represent a sample of the infrastructure and another dataset with RIPE Atlas probes as source points to characterize the Internet edge and end users. The dataset is available on RIPE Atlas under the label "`mcgl`."

3) `CorneoWebC21` [9]: Corneo *et al.* collected `ping` measurements every 3 hours and daily `traceroute` measurements from 8500 RIPE Atlas probes to 189 datacenters operated by GCP, AWS, Microsoft Azure, Digital Ocean, Linode, Alibaba, and Vultr to analyze client-to-cloud latency and assess the suitability of cloud platforms for general-purpose applications. The dataset is available in SQLite format.[2]

[1]Dataset: https://dataserv.ub.tum.de/index.php/s/m1574595

[2]Dataset: https://mediatum.ub.tum.de/1593899

4) Other Datasets: Palumbo *et al.* measured cloud-to-user latency from 25 PlanetLab vantage points to AWS and Microsoft Azure every minute for 14 days [3], [15]. We exclude this dataset because it uses PlanetLab rather than RIPE Atlas. We also exclude the Corneo edge dataset because it overlaps with `MohanHNets19` [21]. Fontugne *et al.* used RIPE Atlas measurements toward root DNS servers and random addresses to study last-mile congestion [22]. We exclude these measurements because they do not target commercial CSPs. Later, [24] collected ping data between RIPE Anchors and CDN edge servers every 2 hours for 10 days, using 79 anchors as sources and 100 edge servers as targets, while [23] executed RIPE measurements to AWS Global Accelerator every 3 hours for 4 days. We do not include them in the cross-dataset comparative analysis because of their limited duration.

III. MEASUREMENT METHODOLOGY

Cloud traffic traverses several public Autonomous Systems (ASs) and ISPs, as well as private CSP networks. Since no complete map of this infrastructure exists, cloud latency studies require geographically distributed measurement points. Previous measurement platforms often required maintenance that volunteer contributions could not sustain, and some eventually ceased operation [15], [34]. We conduct our campaign through RIPE Atlas [11], an open measurement platform maintained by RIPE NCC [35]. It has globally distributed probes that actively measure Internet connectivity and support custom experiments through public APIs.

A. RIPE Atlas platform

RIPE NCC serves as the Regional Internet Registry for Europe, the Middle East, and parts of Central Asia and operates RIPE Atlas as its active measurement platform [11]. At the time of writing, RIPE Atlas included more than 12,000 active probes across over 90% of all countries [36]. In Spain, it maintained 224 active probes in networks serving more than 90% of the country's 38 million Internet users [37].

RIPE Atlas supports `ping`, `traceroute`, DNS, SSL, HTTP, and NTP measurements. Users who contribute resources, such as hosted probes, earn credits that they can use to conduct custom experiments. The platform also provides APIs

and supporting software: `Magellan`, a command-line interface for configuring, streaming, and querying measurements; `Sagan`, a Python library for parsing measurement results; and `Cousteau`, a Python wrapper for the RIPE Atlas API. The platform architecture is detailed in [38].

B. Main measurement campaign details

Our geographical scope is limited to Spain, with the exception of some cloud-based vantage points located in France, because some CSPs do not have a cloud region in Spain, and their closest region is in France. We will refer to this campaign as `singleAS24`.

The measurements were performed using traceroute and initiated from cloud-based probes toward end-user RIPE Atlas probes across Spain, because preliminary attempts in the opposite user→cloud direction showed the cloud infrastructure frequently blocked, dropped, or failed to respond to the messages. We separately explain the different components of the campaign: destination probes, source probes, and the main campaign parameters, such as periodicity, size, and duration.

1) Destination vantage points: The destination probes represent end users. In May 2024, RIPE Atlas contained 624 probes in Spain, of which 224 were active. To isolate the contributions of ISPs and CSPs to latency, we selected stable probes belonging to Telefónica's AS Number (ASN) 3352.

We required each probe to have remained online for at least (i) 99% of the week before the campaign, (ii) 99% of the preceding month, and (iii) 80% of the time since its registration with RIPE Atlas. We obtained these availability metrics by scraping the RIPE Atlas website because its APIs do not provide access to this information.

These criteria produced 36 destination probes. We later discarded four during preprocessing, as explained in Section III-D, leaving the 32 probes shown in Fig. 1. Their distribution broadly follows Spain's population density, with concentrations in coastal regions and around Madrid and Zaragoza. The Canary Islands actually lie approximately 1500 km southwest of the Iberian Peninsula.

2) Source vantage points: Eight cloud probes initiate the traceroute measurements. Alongside probes hosted on CSPs, we include probes within Telefónica's infrastructure to examine cloud services operated by both CSPs and the ISP. Since we aim to characterize general cloud performance rather than rank providers, we anonymize the CSPs as C1, C2, and C3.

The source set comprises IO_{int} (ID 1006162), located at Telefónica's headquarters in Madrid; C1_{io} (ID 1006085), hosted in a C1 deployment at the edge of Telefónica's network; C1_{sp} (ID 1004991) and C1_{fr} (ID 1003375), hosted in C1 regions in Spain and France; $\text{C3}_{\text{sp-std}}$ (ID 1007394) and $\text{C3}_{\text{sp-prm}}$ (ID 1007393), hosted in the Spanish C3 region with basic and premium network services; $\text{C3}_{\text{fr-prm}}$ (ID 1007409), hosted in the French C3 region with premium service; and C2_{fr} (ID 1007405), hosted in the French C2 region.

As Fig. 1 shows, five source probes are located in Spain, four in Madrid and one in Zaragoza, while three are located near Paris. By ownership, the set contains one probe in Telefónica's network, three in C1, three in C3, and one in C2.

Fig. 1. Geographical distribution of the selected probes. The cloud (source) probes are denoted by the icon , whereas the destination probes are denoted by the RIPE Atlas icon . Circled numbers indicate the number of closely located destination probes; this notation avoids overlapping symbols. The Canary Islands (bottom right) and the Paris region (top right) are not shown in their actual locations.

Because C2 has no Spanish cloud region, we use its closest region in France. We also include French regions for C1 and C3 to assess the effect of locating cloud infrastructure in the same country as the users.

3) Measurement setup: We performed ICMP traceroute measurements from each of the eight cloud-based vantage points to each of the 36 destination points. Each `traceroute` measurement sent three packets, each with a 1,499-Byte payload, excluding network- and transport-layer protocol headers (IP, ICMP, UDP, TCP, etc.). The experiments were performed using standard ICMP (Paris=0), with the *Don't Fragment (DF)* bit unset, thereby allowing fragmentation; we verified that no fragmentation occurred anywhere in the dataset. Because ICMP lacks transport-layer port fields, ECMP (Equal-Cost Multipath) hashing causes all packets in a measurement to follow the same path, making repeated measurements directly comparable. Nevertheless, these biases are systematic and consistent across all paths, preserving cross-path comparability within our dataset.

These measurements were repeated *every 30 min* for almost three months: they were initiated on January 30, 2024, and ran consistently until April 24, 2024. That amounts to a total of more than 4000 `traceroute` measurements for each of the 288 CSP–endpoint pairs. The 36 RIPE Atlas measurements (one per destination probe) are publicly available at https://atlas.ripe.net/measurements.[3]

The RIPE Atlas credit system imposes a hard trade-off among the dimensions of the tuple {probes, time granularity, time span, geographical scope}, since increasing one requires reducing another. The optimal balance depends on the campaign objective. In the literature, campaigns range from a few

[3]For the sake of reproducibility, the RIPE Atlas measurement IDs start as `66888XXX`, with the last three digits being: 490, 491, 492, 494, 495, 497, 499, 500, 501, 502, 503, 505, 507, 509, 511, 512, 513, 515, 516, 517, 518, 520, 521, 524, 525, 527, 529, 530, 532, 533, 535, 536, 538, 539, 541, 542.

TABLE II
TRACEROUTE PACKET-LOSS RATE BY DESTINATION PROBE/CLOUD SERVER

Probe ID	341	3712	3726	13881	14866	15118	15618	15632	21537	26072	30039	30392	33627	33818	33948	33971	34344	51265
% loss	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	
% loss	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%	

days [3], [22], [24] (with a higher number of probes) to one year [8], [9] (with sparser temporal granularity). We selected a 30-min granularity and a 3-month span as the optimal temporal trade-off for our study, because ISPs and CSPs are typically interested in forecasting performance over the upcoming days, weeks, or months. This configuration is sufficient to capture daily and weekly patterns (which are the strongest patterns in network traffic), as well as infrequent events occurring on a monthly scale. This work does not target long-term seasonal variations, BGP routing changes, or low-probability network anomalies, which would require a longer observation window at the cost of temporal resolution. Instead, our focus is on whether cloud latency performance can be (temporally or spatially) predicted and explained under normal operating conditions.

C. Additional multi-ISP measurement campaign

The `singleAS24` measurement campaign deliberately considered probes belonging to a single ISP (Telefónica) because our goal was to understand cloud providers' contributions to latency and variability. Thus, selecting a single ISP allowed us to remove the inter-ISP variability.

To address generalizability, we conducted an additional measurement campaign, referred to as `multiISP26`, from April 17 to 24, 2026, with 5 *cloud* vantage points in 4 countries and 40 probes across Spain and France, representing 15 ISPs.

To select the probes, we (i) searched for the probes in Spain and France that had at least one of these tags: wireless, 3g, 4g, 5g, lte, mobile, wifi, or system-wifi. We found only 2 in Spain, and 16 in France; (ii) randomly selected 10 probes from the main campaign (cf. Section III-B), (iii) selected 13 additional random probes from Spain; (iv) added the 2 Spanish probes with a *wireless*-related tag (Wi-Fi); and (v) selected 10 random probes in France and 5 random probes among the 16 probes in France with a *wireless* tag (LTE).

The 40 destination probes span 15 ISPs (8 Spanish, 7 French). Of these, 33 (82.5%) use fiber and 7 (17.5%) are connected through wireless technologies. Only 26 destination probes returned data; all probes were online, so the most likely reason is the presence of firewalls or other security measures. We remove outliers above 500 ms (0.15%), which correspond to timeouts. All the other configuration parameters are the same as in `singleAS24`. `multiISP26` is included as a sanity-check dataset. Our analysis focuses on `singleAS24`. Results from `multiISP26` are included for validation, and the two campaigns are discussed in Section VII-A.

D. Dataset pre-processing

The results were retrieved from RIPE Atlas in JSON format. We enhanced the dataset with additional features to supplement the available information about probes and measurements, and we performed a preliminary analysis of the probes' connectivity to remove those that were not functioning correctly. The following details correspond to `singleAS24`.

1) Additional Information: We retrieved all the probe metadata available through RIPE Atlas APIs, including spatial coordinates, ASN, IP addresses, and network prefixes, which are not directly provided in the measurements. We computed and incorporated the distance for each source–destination pair. Finally, we included in the dataset the average RTT of the three packets sent simultaneously at each hop of each traceroute.

2) Probe Responsiveness: We examined traceroute responses and timeouts before conducting the analysis. First, we removed one destination probe that returned no successful measurements, likely because its network blocked traceroute traffic. Table II reports packet loss for the remaining 35 probes. Two probes remained unreachable for substantial portions of the campaign, producing loss rates of 78.56% and 22.78%, so we removed them. We also removed probe 52511 because it reported latencies between 2 s and 3 s for approximately 85% of its samples. The final dataset therefore contains 32 destination probes and 256 source–destination pairs, as shown in Fig. 1. Table III aggregates packet loss by cloud source before and after these removals. We observe that excluding the three malfunctioning probes substantially reduces unsuccessful measurements for every cloud source.

TABLE III
SUCCESSFUL TRACEROUTE SAMPLES PER CLOUD SERVER

Probe ID (cloud)	Cloud label	% loss (35 probes)	% loss (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%

IV. CLOUD LATENCY CHARACTERIZATION

We characterize cloud latency in the `singleAS24` campaign. We study (i) the temporal patterns over a typical day, (ii) the probability distribution for each CSP and source–destination pair, (iii) the correlation between RTT and distance, and (iv) the variability and statistics of IP paths, i.e., the alternative sequences of IP addresses traversed for each source–destination pair, as observed in traceroute samples.

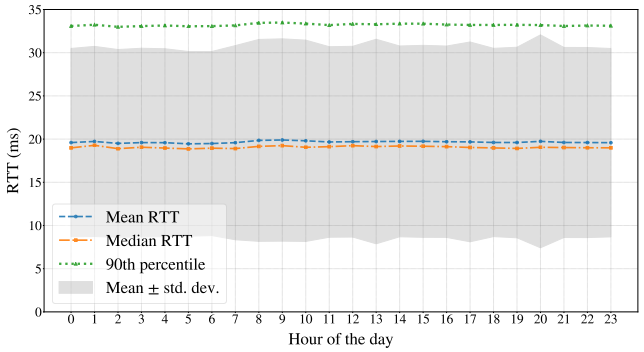

Fig. 2. Daily RTT statistics, aggregated over all source and destination probes.

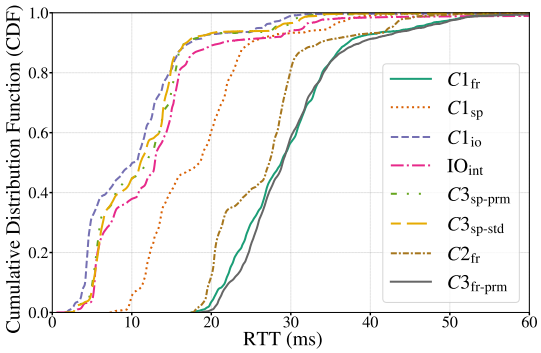

Fig. 3. CDFs of the 8 cloud probes, aggregating data across all destinations.

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

1) Daily RTT patterns: An aspect of interest in latency characterization is the identification of daily patterns in the RTT. As a first step, we cluster the 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-month period. We represent the mean, median, and 90th percentile of the RTT for each hour in Fig. 2. In addition, the shaded area represents the range given by mean $\pm$ one standard deviation. We observe that these statistics remain remarkably constant across the whole day and across the whole 3-month period measured. This behavior, previously noticed in [3], is also consistent across vantage points. The only visible variation is the subtle variance increase during peak hours (i.e., morning, 2 p.m., 9 p.m., mainland Spanish time). However, these spikes are negligible (1–2 ms) and do not affect the performance of the underlying services. This finding is supported by Bhat *et al.* [24], who stated that "End-to-end latencies of paths between end users and CDN edge servers are typically quite stable, spikes are rare, and variations are small, if any."

These remarks are supported by formal statistical tests. The ADF (Augmented Dickey-Fuller) test [39] confirms mean-reverting behavior in RTT for 93.2% of paths (median $p = 9.9 \cdot 10^{-11}$), confirming that RTT does not drift or follow a random walk, whereas the KPSS (Kwiatkowski-Phillips-Schmidt-Shin) test [40] confirms level-stationarity for only 31.8% of paths, which at first appears contradictory. However, this pattern is known to indicate evidence of piecewise stationarity with occasional structural breaks rather than genuine long-term instability [41]. This interpretation is corroborated by the small practical effect sizes reported below and by the per-IP path analysis in Section IV-C. To formally assess latency stability across the day, we applied Mann-Whitney U tests [42] comparing day (07:00–20:59) and night (21:00–06:59) RTT distributions per pair. No significant difference was found for 62.1% of paths (median $p = 0.148$), and, even where significant, the effect is negligible (median $|\Delta median| = 0.061$ ms, $|r| = 0.056$). Levene's test [43] confirms that variance differs across paths ($p \approx 0$), reflecting heterogeneous link distances and routing conditions rather than temporal instability. Overall, the uniformly small effect sizes support the claim that latency is notably stable over time.

This steady behavior is partially explained by the bias of RIPE Atlas networks toward well-connected devices and the subsequent under-representation of wireless devices, since cellular wireless networks are more likely to experience variable delays due to the variable congestion at the access network [44]. Although we have omitted the visualization of the latency characterization results for the `multiISP26` campaign due to space constraints, the stability across the day observed in Fig. 2 is maintained, with the only difference being an increase in the mean and median RTT and in the 90th percentile (5 ms each, approx.), which stems from the longer distances. Thus, including several ISPs and wireless access technologies does not appear to materially affect these results.

2) Cumulative Distribution Function (CDF) of RTT: We first aggregate the 32 destination probes for each cloud source in Fig. 3. All eight CDFs have steep slopes, with approximately 90% of samples falling within a 15 ms range, which indicates stable latency. The three French source vantage points have the highest RTT, with a nearly constant gap of approximately 20 ms that can affect the QoS of latency-sensitive applications such as extended reality.

The basic and premium C3 services, $C3_{\text{sp-std}}$ and $C3_{\text{sp-prm}}$, show equivalent variability. This differs from previous findings that public-network services can exhibit greater variability than premium services [17]. $C1_{\text{io}}$, the only edge source probe, achieves the lowest RTT, although its advantage over IO_{int}, $C3_{\text{sp-prm}}$, and $C3_{\text{sp-std}}$ is not significant. In contrast, $C1_{\text{sp}}$ underperforms the other Spanish cloud sources.

Fig. 4 presents the 32 individual links from each source and highlights the 10th, 50th, and 90th percentiles of mean RTT. The bottom plot provides a detailed view of IO_{int}, $C1_{\text{io}}$, $C2_{\text{fr}}$, and $C3_{\text{fr-prm}}$. The individual CDFs are steeper than the aggregated curves, showing greater stability at the link level. Some links exhibit stepwise behavior, such as the 10th-percentile link for $C2_{\text{fr}}$, but the steps remain within approximately 5 ms and have little operational impact. Most destination probes also remain within a 15 ms range, which is consistent with the aggregate results.

Two probes in the Canary Islands deviate from this pattern and show higher RTT, further demonstrating the effect of distance. In similarly remote regions, edge infrastructure becomes necessary for services requiring network latency below 40 ms.

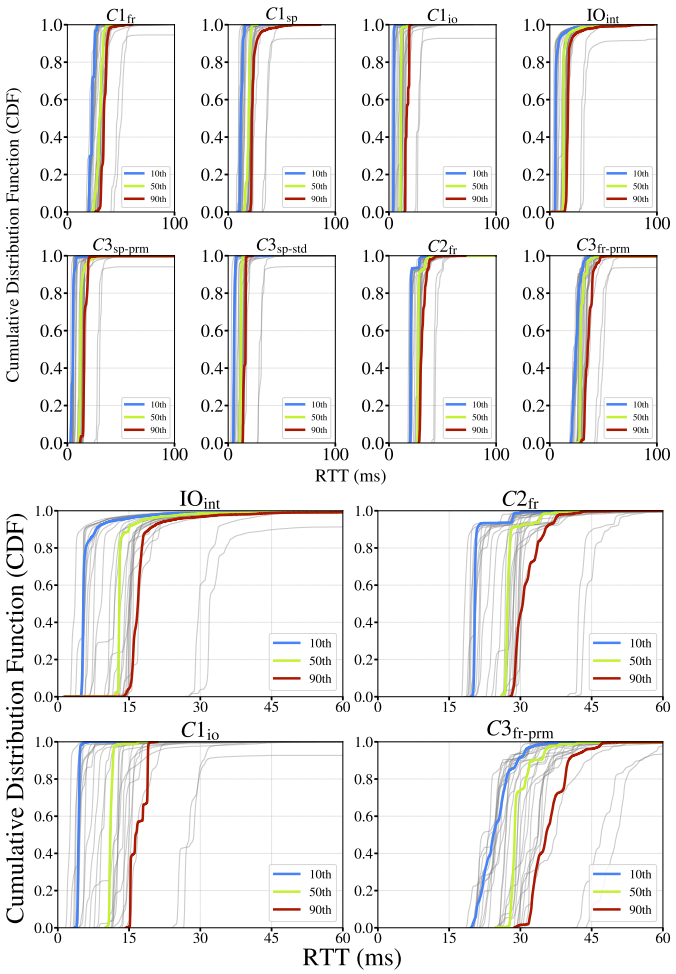

Fig. 4. Per-pair RTT CDFs grouped by cloud source (top) and detailed for four sources (bottom). Colored curves identify destination probes at the 10th, 50th, and 90th percentiles of mean RTT.

B. Characterization of RTT as a function of distance

We examine the validity of a core assumption in cloud computing: Is geographical distance truly the most critical factor influencing latency between users and cloud infrastructure?

To ascertain whether RTT and distance are linearly correlated, we fit a least-squares linear regression to their relationship. Fig. 5 shows the per-pair median RTT as a function of distance, along with the fitted regression line, the per-pair interquartile range, and the 95% confidence interval.

We find that RTT increases by about 2 ms per 100 km (i.e., 20 ms per 1000 km), with an additive minimum delay of 8.29 ms. This result is consistent with the 15 ms gap between the sources located in Madrid and those located in Paris, and may partially justify the underperformance of $C1_{sp}$ with respect to the other sources located in Spain, due to the concentration of destination probes in Madrid area and the fact that $C1_{sp}$ is located in Zaragoza, 350 km away from the capital of Spain. A coefficient of determination of $R^2 = 0.81$ suggests a strong linear correlation, although other factors, including stochastic variability, also significantly influence the RTT. This observation underscores the need for further analysis to identify the primary drivers of user-to-cloud latency.

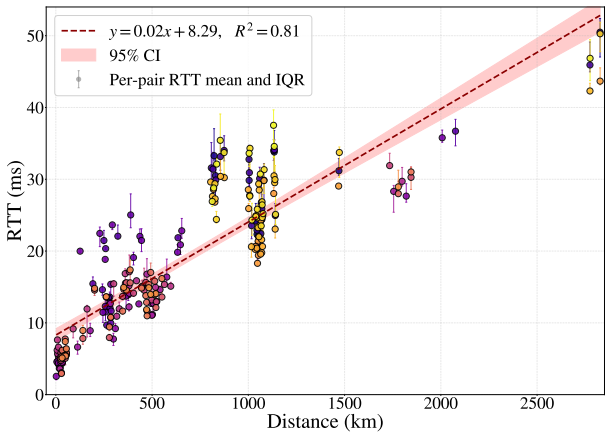

Fig. 5. RTT statistics for each source–dest. pair as a function of the distance.

C. Impact of path variability on RTT

Once we have verified that distance cannot completely determine the mean RTT to the cloud, we analyze the impact of IP path variability on latency by means of the path information provided by `traceroute`.

Fig. 6 illustrates the RTT variability across time for different IP paths. Each color represents a different route traversed by the packets. From a qualitative perspective, we can observe in Fig. 6 that each cloud probe is characterized by a stable RTT floor, higher for the French regions. We also observe that a given path experiences a stable RTT; this property is clear, for example, for $C2_{fr}$ (e.g., gray, green, or salmon colors), $C1_{io}$ (e.g., lilac, dark blue, purple, or red routes), or IO_{int} (e.g., green, red, purple, or peach cases). While less immediately apparent, the same behavior is observed across the remaining cloud probes and for most paths. We show the RTT stability for a specific path in Fig. 7, which also reveals that there exists temporal stability on the chosen routes, since a particular route becomes dominant every few days.

To quantify the influence of the IP path on the RTT, we obtain the main statistics for the most frequent routes, and we illustrate them in Fig. 8. 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 when we jointly consider all paths. Numerically, the standard deviation (StD) of the RTT over all samples is 1.3 ms, while the mean StD 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 & Explainability

We aim to determine (i) whether future connectivity performance can be forecast from the preceding characterization to estimate RTT in advance, (ii) which models are most suitable for this task, and (iii) whether the forecasting models can be explained to identify the most relevant features.

We consider two problems: *time series forecasting*, i.e., predicting the future RTT of an existing link, and *spatial forecasting*, i.e., inferring cloud performance for a new user before it is actually connected from other cloud–user sessions.

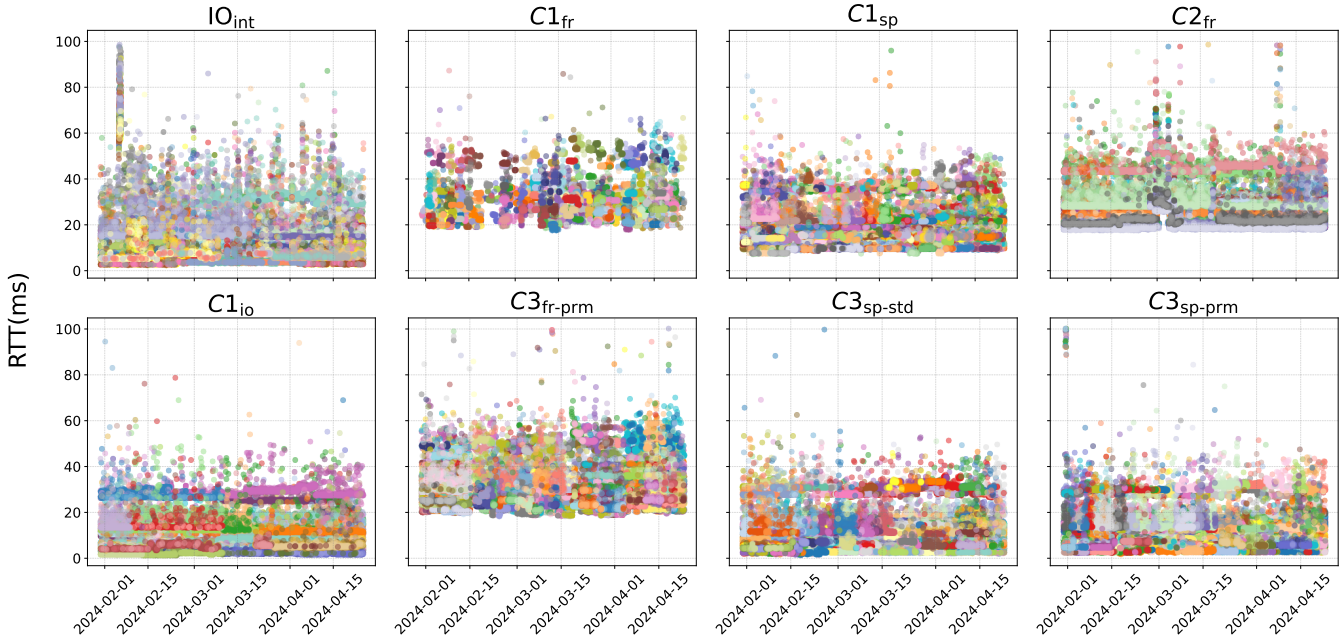

Fig. 6. RTT measurements from each cloud vantage point. Colors represent different IP paths, and only paths that occur at least 1% of the time are represented.

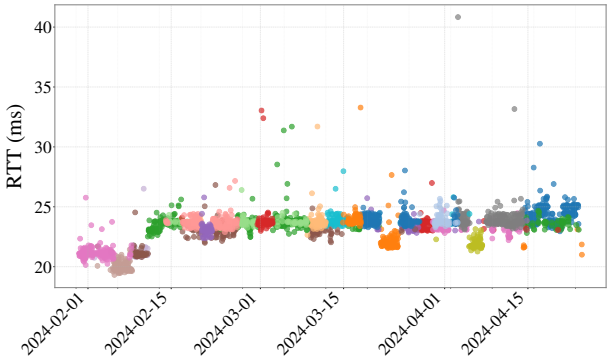

Fig. 7. RTT per IP path from C1$_{sp}$ to destination probe with ID 1003090.

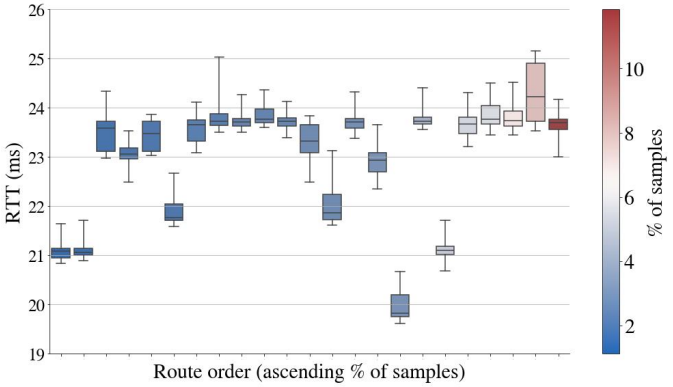

Fig. 8. Statistical distribution of the RTT for distinct IP paths from C1$_{sp}$ to destination probe with ID 1003090. Boxes represent the median and the interquartile range, whiskers indicate the 5th and 95th percentiles, and color shows the percentage of samples that use the corresponding IP path.

A. Time series forecasting

We consider a range of well-established forecasting methods, from the most basic algorithms to state-of-the-art methods, to determine whether complex Machine Learning (ML)-based methods are required. These methods are: *Naive forecast*, which matches the prediction to the latest available RTT value, *Simple Exponential Smoothing (SES)*, *Holt's linear*, *AutoRegressive Integrated Moving Average (AutoARIMA)*, which automatically identifies the optimal parameters for an ARIMA model (p,d,q) with seasonal parameters (P, D, Q, m), Facebook's *Prophet* [45], and an *LSTM neural network*. SES and Holt's linear can be expressed as ARIMA(0,1,1) and ARIMA(0,2,2), respectively. For completeness, we also consider a state-of-the-art model for *spatio-temporal* forecasting called Graph WaveNet (GWN) [46]. The graph represents the measurement paths, while the model produces forecasts for the desired time horizons. The model is composed of four blocks with two layers per block, 32 hidden channels, and a look-back window of four observations.

Traceroute data were collected from RIPE probes at 30-min intervals. We consider several forecast horizons: 30 min, 3 h, and 6 h (1, 6, and 12-step-ahead forecasts, respectively).

The performance of the considered methods is summarized in Table IV and Table V for the `singleAS24` and `multiISP26` datasets, respectively. In these tables, we provide the dimensionless normalized root mean squared error (NRMSE), the actual root mean squared error (RMSE) in milliseconds, and the 95% Prediction Interval (PI). The 95% PI represents the range between the 2.5th and 97.5th percentiles of the error distribution. We first observe that most methods (up to Prophet) provide similarly accurate predictions. The errors of LSTM and Graph WaveNet are approximately one order of magnitude larger. At first glance, this result may seem surprising, given that Prophet, LSTM, and Graph WaveNet are the most advanced methods among those considered.

TABLE IV
AVG. RTT FOR TIME-SERIES FORECASTING (singleAS24)

	30 min			3 hours			6 hours		
Method	**NRMSE**	**RMSE (ms)**	**PI (norm)**	**NRMSE**	**RMSE (ms)**	**PI (norm)**	**NRMSE**	**RMSE (ms)**	**PI (norm)**
Naive	**0.09**	**0.97**	**0.06**	**0.19**	**2.05**	**0.81**	0.23	2.50	1.03
SES	0.12	1.32	0.30	0.20	2.18	0.90	**0.23**	**2.49**	**1.01**
Holt linear	0.20	2.18	0.84	0.30	3.25	1.36	0.42	4.55	1.85
a-ARIMA	0.09	0.98	0.07	0.21	2.27	0.80	0.28	3.06	0.97
Prophet	0.30	3.28	1.12	0.34	3.76	1.27	0.39	4.25	1.44
LSTM	0.89	9.69	0.68	0.98	10.77	0.68	0.99	10.84	0.67
GWN	0.97	10.62	3.28	0.98	10.66	3.29	0.98	10.64	3.29

TABLE V
AVG. RTT FOR TIME-SERIES FORECASTING (multiISP26)

	30 min			3 hours			6 hours		
Method	**NRMSE**	**RMSE (ms)**	**PI (norm)**	**NRMSE**	**RMSE (ms)**	**PI (norm)**	**NRMSE**	**RMSE (ms)**	**PI (norm)**
Naive	**0.04**	**0.51**	**0.03**	**0.09**	**1.03**	**0.15**	0.11	1.31	0.26
SES	0.05	0.62	0.07	0.09	1.06	0.19	**0.11**	**1.30**	**0.29**
Holt linear	0.08	0.95	0.19	0.24	2.93	0.49	0.46	5.51	0.92
a-ARIMA	0.05	0.58	0.06	0.16	1.87	0.32	0.24	2.83	0.47
Prophet	0.15	1.79	0.52	0.17	2.06	0.60	0.20	2.34	0.69
LSTM	0.60	7.15	0.26	0.64	7.67	0.26	0.70	8.40	0.27
GWN	0.93	10.96	3.79	0.82	9.65	3.58	0.85	9.95	3.64

TABLE VI
SPATIAL FORECASTING PERFORMANCE (singleAS24)

	Only Distance			All Features			Without Distance		
Method	**NRMSE**	**RMSE (ms)**	**PI (norm)**	**NRMSE**	**RMSE (ms)**	**PI (norm)**	**NRMSE**	**RMSE (ms)**	**PI (norm)**
Linear	0.55	5.95	1.93	0.53	5.79	1.81	0.93	10.12	3.58
Dec. tree	**0.46**	**4.98**	**1.71**	0.52	5.62	1.93	1.05	11.41	4.39
RF	**0.46**	**4.98**	**1.71**	0.45	4.94	1.71	0.88	9.60	3.65
XGBoost	0.49	5.38	1.89	**0.41**	**4.46**	**1.58**	**0.81**	**8.81**	**3.33**
SVM	0.56	6.16	2.09	1.50	16.32	4.24	2.88	31.41	8.41
KNN	0.49	5.35	1.96	0.65	7.09	2.50	0.95	10.38	3.97

TABLE VII
SPATIAL FORECASTING PERFORMANCE (multiISP26)

	Only Distance			All Features			Without Distance		
Method	**NRMSE**	**RMSE (ms)**	**PI (norm)**	**NRMSE**	**RMSE (ms)**	**PI (norm)**	**NRMSE**	**RMSE (ms)**	**PI (norm)**
Linear	**0.68**	**7.97**	**2.55**	**0.65**	**7.66**	**2.50**	0.95	11.23	3.60
Dec. tree	0.81	9.57	3.22	0.78	9.19	2.97	1.03	12.16	4.28
RF	0.81	9.57	3.22	0.71	8.44	2.77	0.96	11.35	3.86
XGBoost	0.80	9.46	3.23	0.65	7.72	2.56	**0.83**	**9.79**	**3.44**
SVM	0.69	8.15	2.55	1.33	15.71	4.37	2.79	33.16	8.53
KNN	0.81	9.52	3.24	1.10	13.01	4.29	1.11	13.14	4.33

However, upon considering the findings from Section IV, this outcome becomes less unexpected: Latency is relatively stable, with occasional spikes that are not always predictable. Thus, applying simple methods that maintain the main trends can provide better results, since advanced models aim to estimate complex (in this context, nonexistent) patterns. While fine-tuning LSTMs could enhance their performance, it is not our intention to conduct such optimization. Instead, our goal is to emphasize that simpler methods that do not require this level of optimization can still achieve high accuracy in operational terms due to the stable RTT. Additionally, we observe similar results and trends across both datasets. Prediction error increases slightly as the forecasting horizon lengthens, although it remains within the same range.

B. Spatial forecasting

Next, we investigate whether the performance of new cloud-to-end-user connections for users in different locations and with different characteristics can be predicted from data on existing connections. Furthermore, we aim to provide explainable results that clarify which features have the greatest impact on the predictions.

There are numerous regression methods available for use, with diverse levels of explainability. For a comprehensive overview of ML algorithms, categorized by the interpretability of their solutions, we refer to [47]. We consider six supervised-learning methods: Linear regression (ordinary least squares), Decision Tree, Random Forest (RF), Extreme Gradient Boosting (XGBoost) (the last two with 100 estimators), K-nearest neighbors (KNN) ($K = 5$), and Support Vector Machines (SVM) (linear, 5k iterations). The detailed model hyperparameters can be found in the publicly available codebase [12].

We analyze 3 scenarios: (i) the case where distance is the sole feature considered for forecasting, (ii) the case where all features from the dataset are considered for the prediction, and (iii) the case where distance is removed from the input features. We split the data such that all the measurements for 75% of the *source–destination pairs* in the dataset are used for training, whereas the remaining 25% of pairs are used for testing. We ran 20 experiments for each method, and we used random seeds to select which source–destination pairs would fall into the training and test sets for each experiment.

The results for the singleAS24 campaign are summarized in Table VI. The distance-only prediction is quite accurate, consistent with the characterization in Section IV. Across all algorithms, RMSE remains close to 6 ms, which is a reasonable error margin given that CSPs and ISPs are primarily concerned with whether the RTT falls within a given range. Adding all features further reduces RMSE to 4.46 ms with XGBoost, the best-performing algorithm. We provide the error PDF per method in Fig. 9. The narrow distribution of the model errors around zero implies that error variance is minor, and estimates are within 10 ms of the actual value with high probability.

Additionally, the RMSE in the case with no distance increases by 100% on average relative to the case where *only* distance is considered. That is, the error doubles. Conversely, the error is reduced by only around 10% when we move from considering only distance to considering all the features.

Finally, we also provide the results for the multiISP26 campaign in Table VII. We observe that spatial forecasting for this campaign achieves slightly worse accuracy (+20%) than for the singleAS24 data. This result is expected due to the shorter campaign duration and more diverse spatial distribution, including sparser geographical locations.

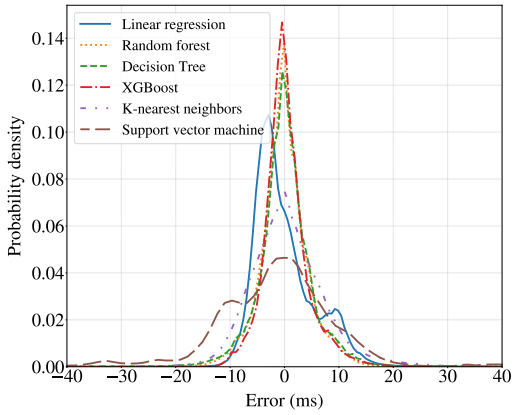

Fig. 9. PDF of the forecasting errors for each considered algorithm.

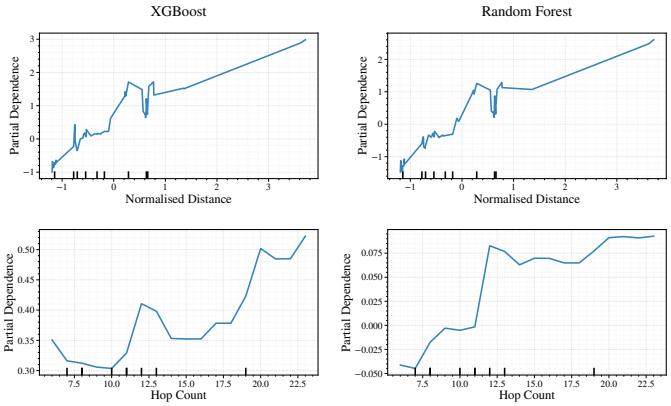

Fig. 10. PDP comparing model behavior for the most relevant features.

C. Explainability and feature importance

1) Explainability methods: We investigate the features affecting latency using post-hoc explainable ML methods [48]. Explanations may be local or global and model-agnostic or model-specific. We use LIME [49] and SHAP [50] and validate their findings using the global methods Permutation Feature Importance (PFI), Partial Dependence Plots (PDP), and Partial Dependence Range (PDR).

LIME explains an individual prediction by perturbing the corresponding sample, obtaining predictions for the generated samples, and weighting them according to their similarity to the original sample. It then fits a local linear model whose coefficients represent each feature's contribution. In contrast, SHAP uses game-theoretic Shapley values [51]. It measures a feature's marginal contribution by comparing model predictions with and without that feature across possible feature subsets, then averages these contributions to obtain the feature's Shapley value.

PFI measures the increase in prediction error when a feature's values are randomly shuffled [48, Ch. 23]. This shuffling breaks the relationship between the feature and the target. PDR represents the difference between the maximum and minimum predicted outputs as a feature varies across its observed range while averaging over the other features [48, Ch. 19]. Finally, we also show the PDPs to reveal the type of relationship between the feature and the prediction.

TABLE VIII
XAI ANALYSIS WITH BOOTSTRAPPED SHAP VARIANCE

	XGBoost				Random Forest			
Feature	LIME	SHAP±StD	PFI	PDR	LIME	SHAP±StD	PFI	PDR
Distance	**0.74**	**0.74±0.40**	**1.45**	**2.51**	**0.84**	**0.80±0.07**	**1.56**	**2.62**
Dst. ok	0.17	5e-3±6e-3	0.01	0.38	0.01	2e-3±1e-3	0.01	0.15
Hop count	0.12	0.06±0.06	0.02	0.22	0.04	0.05±0.02	0.01	0.14
Src ASN	0.08	0.07±0.06	0.03	0.22	0.03	0.06±0.02	0.03	0.17
Time	0.01	0.04±0.04	0.03	0.31	0.01	0.03±3e-3	0.02	0.16

2) Explaining latency performance: We evaluate LIME and SHAP to determine which attributes affect the spatial RTT forecast for the two best-performing models from Table VI: XGBoost and Random Forest. We perform twenty runs for each experiment, using Monte Carlo simulations to account for variability. Results are averaged across these runs to obtain a more reliable estimate of performance. In each iteration, we randomly select 10,000 dataset samples. For LIME, we choose individual instances at random, whereas for SHAP we choose random groups of samples. We present in Table VIII the top five features sorted by their weight in the forecast explanation.

SHAP and LIME weights are unitless and indicate the extent to which each feature influences the model's output. The sign of these values denotes whether the feature has a positive or negative impact. As we are interested mainly in the magnitude of the impact, rather than the direction, we provide the average *absolute* value over the twenty runs.

We observe in Table VIII that both LIME and SHAP provide similar results for the two regression methods: Distance ranks first: for XGBoost and Random Forest, respectively, LIME reports 0.74 and 0.84, while SHAP reports 0.74 and 0.78. The second-ranked weights are 0.17 and 0.04 for LIME and 0.07 and 0.06 for SHAP. From the second to the fifth feature, we notice that weights are relatively close and that the order of features varies in each case: the second and third most important features are always (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 indicates that different providers have strong, stable connectivity, whereas the hop count indicates that stabilizing paths and minimizing hops is also important for providing adequate service, as also illustrated in Fig. 7.

Table VIII includes PFI and PDR values. PFI and PDR also consistently identify distance as the dominant feature, with values exceeding those of all other features by two orders and one order of magnitude, respectively. We also show the PDP for *distance* and *hop count* in Fig. 10, where *normalized distance* represents the distance after z-score normalization.

To strengthen these results, we also evaluate the Normalized Movement Rate (NMR) metric [52] as a way to mitigate the effects of correlations among the input features. NMR checks how the importance of each feature changes when the most important feature is removed, recursively. A lower NMR score (which ranges from 0 to 1) indicates a greater stability in the feature-importance ranking, and therefore greater confidence in the stability of the SHAP explanations. For the best-performing model in our analysis, XGBoost, the overall NMR

TABLE IX
TIME-SERIES FORECASTING ACROSS LITERATURE DATASETS

	Mohan-Corneo [8]			Candela-Gregori [7]			Corneo-Eder [9]		
Method	NRMSE	RMSE (ms)	PI	NRMSE	RMSE (ms)	PI	NRMSE	RMSE (ms)	PI
Naive	0.77	76.79	2.81	0.39	41.89	1.45	0.35	38.66	0.84
SES	0.62	61.69	2.14	0.39	42.11	1.41	0.46	49.63	1.84
Holt Linear	0.72	71.10	2.64	0.47	50.86	1.81	0.59	65.11	2.61
a-ARIMA	**0.55**	**54.20**	**1.70**	**0.35**	**38.12**	**1.27**	**0.35**	**37.94**	**0.89**
Prophet	0.55	55.08	1.74	0.52	55.89	1.89	0.99	107.97	4.20
LSTM	0.57	56.22	1.69	1.04	117.79	3.55	0.97	109.89	3.54

over 20 iterations is 0.00342, indicating ranking stability and low impact of feature correlation. After removing the most important feature (distance), hop count and source ASN remained the top features.

3) Computational cost and deployment feasibility of XAI: Integrating post-hoc explainability into large-scale cloud-connection monitoring systems introduces computational overhead. Their computational cost depends on factors such as the number of features, perturbation or coalition samples, or background observations. Nevertheless, SHAP and LIME should not be applied to every latency forecast; instead, explanations are generated selectively (e.g., following anomalous forecasts, service-level objective violations, periodic audits, or explicit operator requests). This design is consistent with selective-explanation frameworks in which practitioners control the fraction of observations receiving more computationally intensive explanations, thereby balancing explanation quality against average computational cost [53].

Since explanations are decoupled from the (potentially) latency-constrained forecasting path, SHAP and LIME do not need to operate at the same throughput as the forecasting algorithm. The main constraints are instead the amortized explanation cost, the available background computational capacity, and whether explanations are returned within the time window required for diagnosis or intervention. Scalability can be further improved via batching, parallel execution, or model-specific explainers (e.g., TreeSHAP provides polynomial-time computation for tree-based models such as XGBoost [54]).

VI. COMPARATIVE ANALYSIS OF PUBLIC CLOUD-LATENCY DATASETS

In this section, we analyze the main publicly available datasets related to cloud latency. Our objective is to evaluate the generality of our findings and provide a comprehensive view of the current status of cloud latency characterization. In fact, one of the major challenges we have encountered is the heterogeneity in the formats, temporal and geographical scopes, and data structure of the dataset, which introduces some limitations in their comparison.

We compare our dataset with the existing studies and discuss our findings. Then, we apply the forecasting and explainability process described in Section V to these related datasets.

A. Comparison with related studies

We add a new measurement campaign to the existing latency measurement research to help the community develop a more comprehensive understanding of the latency performance of cloud systems. Measurement campaigns conducted on the public Internet to monitor cloud latency performance have many degrees of freedom in terms of parameter selection. Since resources are limited, there is a trade-off between the number of source and destination points, geographical resolution, temporal resolution, service providers, countries, technologies, and other aspects. Thus, comparing findings of different studies is often complex because the selected parameters differ. For example, while the studies in [8], [9] are 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 worldwide) and temporal duration, whereas we improved the temporal granularity to 30 minutes 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 explainability methods to better comprehend the complex relationship between features and latency. We also find major differences in experimental setup relative to other studies such as [3] (25 end-user vantage points, 8 cloud datacenters, 14 days, `ping` packets), [17] (from 6 GCP regions to 458 speed-test servers, focusing on throughput), or [21] (30 datacenters in U.S.A., focused on edge deployment).

Our findings highlighted in Fig. 2 and Fig. 3 align with those of [3], which found that there were 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 studies. For example, [17] found that, for some CSPs, public-network plans showed greater variability than their premium CSP-private-network counterparts, 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 result illustrates how measurement campaigns provide snapshots of latency behavior that may differ, and we must continue research in this area.

Next, we evaluate the datasets from Section II-B [7]–[9]. To set the stage for a deeper analysis, we report their mean RTT and StD as follows: (61.9 ms, 77.4 ms) for `CorneoWebC21`, (91.6 ms, 96.3 ms) for `MohanHNets19`, and (174.4 ms, 104.8 ms) for `CandelaAcc19`. As we observe, dataset means differ substantially, and the mean RTT in `CandelaAcc19` is three times higher than that in `CorneoWebC21`.

B. Cloud latency forecasting for datasets from the literature

We apply the temporal and spatial forecasting methods from Section V to the three literature datasets. For time-series forecasting, Table IX identifies AutoARIMA as the best-performing method for all three datasets. This agrees with our campaign and supports the finding that cloud latency generally remains stable except for discrete changes. However, the errors exceed those obtained for our dataset and amount to approximately half of each dataset's StD, indicating greater variation

TABLE X
SPATIAL FORECASTING PERFORMANCE FOR THE LITERATURE DATASETS. COMPLETE DATASETS

	MohanHNets19 [8]				CandelaAcc19 [7]				CorneoWebC21 [9]			
Method	NRMSE	MAPE	RMSE±StD (ms)	95%PI	NRMSE	MAPE	RMSE±StD (ms)	95%PI	NRMSE	MAPE	RMSE±StD (ms)	95%PI
Linear model	0.59	1.89	58.66 ± 118.40	1.82	0.59	3.57	64.36 ± 128.80	2.32	0.52	2.52	57.12 ± 100.60	2.17
Decision tree	0.39	1.05	38.83 ± 103.26	1.83	0.40	1.96	42.92 ± 103.61	1.55	0.36	0.96	38.93 ± 86.42	1.49
Random forest	0.33	0.91	33.06 ± 93.69	1.65	0.31	1.58	33.27 ± 82.82	1.17	0.29	0.96	32.10 ± 76.50	1.16
XGBoost	**0.30**	**0.95**	$\mathbf{29.97 \pm 85.15}$	**1.42**	**0.29**	**1.50**	$\mathbf{31.55 \pm 81.33}$	**1.08**	**0.27**	**0.87**	$\mathbf{29.13 \pm 71.60}$	**1.07**
SVM	1.00	1.00	99.08 ± 151.58	3.12	1.00	1.00	108.64 ± 133.67	3.54	0.99	1.00	107.99 ± 152.32	3.37
KNN	0.64	1.88	63.43 ± 118.70	1.67	0.86	5.60	93.73 ± 156.18	3.67	0.71	2.64	77.58 ± 135.33	3.62

TABLE XI
SPATIAL FORECASTING PERFORMANCE FOR THE LITERATURE DATASETS. EUROPEAN USERS TO CLOUD SITES IN FRANCE AND SPAIN

	MohanHNets19 [8]				CandelaAcc19 [7]				CorneoWebC21 [9]			
Method	NRMSE	MAPE	RMSE±StD (ms)	95%PI	NRMSE	MAPE	RMSE±StD (ms)	95%PI	NRMSE	MAPE	RMSE±StD (ms)	95%PI
Linear model	**0.82**	**5.30**	$\mathbf{18.46 \pm 71.02}$	**1.70**	0.94	4.02	34.40 ± 95.22	3.50	**0.87**	**10.48**	$\mathbf{20.33 \pm 68.20}$	**2.00**
Decision tree	0.91	10.55	20.62 ± 69.77	2.70	0.78	3.26	28.79 ± 99.35	2.40	1.29	7.36	30.16 ± 75.22	4.07
Random forest	0.85	7.19	19.16 ± 69.70	2.21	**0.62**	**2.44**	$\mathbf{22.89 \pm 79.25}$	**1.84**	1.10	5.46	25.75 ± 69.69	3.25
XGBoost	0.82	5.12	18.64 ± 70.39	1.95	0.63	2.87	23.00 ± 80.81	1.84	1.11	5.49	25.90 ± 69.56	3.31
SVM	1.00	1.00	22.56 ± 70.75	3.07	1.02	1.00	37.22 ± 95.94	3.78	0.99	1.00	23.13 ± 70.64	2.90
KNN	1.22	11.81	27.52 ± 71.20	4.39	1.15	6.45	42.29 ± 95.40	5.04	1.33	33.62	31.18 ± 72.52	4.35

and jitter. These differences reflect the worldwide scope of the literature datasets, whose paths traverse longer distances and more heterogeneous networks, ASNs, and exchange points.

Because geographical scope may affect spatial forecasting, we evaluate two scenarios. Our campaign examines the impact of deploying a cloud region in a mid-sized country rather than comparing CSPs or network operators, whereas the literature datasets cover a broader, mostly worldwide area. We therefore analyze the complete datasets and filtered datasets containing measurements from European user probes to cloud sites in Spain or France. The `MohanHNets19` and `CorneoWebC21` datasets contain no Spanish cloud sites, so we use their French sites. We retain user probes across Europe because restricting them to Spain would produce insufficient data for learning. For each scenario, we use all available input features and report the average performance over 20 runs using NRMSE, Mean Absolute Percentage Error (MAPE), RMSE, and its StD.

For the complete datasets, Table X ranks XGBoost first and Random Forest second across all three datasets, consistent with our findings. The MAPE remains between 1 and 2%, while the StD considerably exceeds the mean RMSE. This dispersion reflects latency spikes and outliers caused by network failures, reconfigurations, congestion, and packet loss. Because NRMSE is normalized, it supports comparisons among algorithms but does not represent absolute error.

The filtered results in Table XI preserve the main patterns. Filtering reduces RMSE but increases MAPE, particularly for `CandelaAcc19` and `CorneoWebC21`: the smaller geographical scope lowers mean latency, while the smaller sample size increases percentage error. Nevertheless, these datasets retain higher RMSE than our campaign because they cover users across Europe and therefore traverse more countries, ASs, and networks with different service quality. Our campaign covers a smaller area consisting primarily of one country, with the addition of cloud servers in France.

C. Explainability for literature datasets

We evaluate the best-performing algorithm, XGBoost, for the three datasets and for the two cases considered: the whole datasets, and the filtered datasets focused on the links between Europe and Spain-France. Due to the high computational cost of post-hoc explainability methods, we randomly select a set of 1000 instances and perform the analysis on this subset.

We show the features with the greatest weights for both LIME and SHAP in Table XII for the complete datasets and in Table XIII for the Europe-filtered datasets. We observe some consistency in the LIME and SHAP explanations across datasets and cases. Generally, *Distance* and source-related features are more impactful than other features like timestamp and destination-related features. We recall that, in this case, the cloud vantage points are the destinations, i.e., the user probe characteristics matter more than the CSP's features.

For LIME, all cases follow the same structure, with *Distance* as the most influential feature and with the subsequent features obtaining a weight one order of magnitude smaller. Hence, from a user's perspective, the distance to the cloud server is the primary determinant of the latency they experience. Concerning SHAP, it assigns *Hop Count* and the source location attributes as the most impactful features. We note that the fact that *Distance* fades in importance for SHAP is mostly explained by the sensitivity of SHAP to confounders and inter-feature correlations [55], since, e.g., the number of hops is highly correlated with the distance.

VII. DISCUSSION & INSIGHTS

A. Challenges

The cross-dataset comparative analysis presented several challenges, the most important of which are as follows:

TABLE XII
LIME AND SHAP WEIGHTS FOR XGBOOST FOR THE LITERATURE DATASETS. COMPLETE DATASETS

	LIME						SHAP					
	CorneoWebC21		CandelaAcc19		MohanHNets19		CorneoWebC21		CandelaAcc19		MohanHNets19	
Order	*Feature*	*Weight*	*Feature*	*Weight*	*Feature*	*Weight*	*Feature*	*Weight*	*Feature*	*Weight*	*Feature*	*Weight*
1	Distance	0.406	Distance	0.672	Distance	0.666	Src. Long.	0.051	Distance	0.032	Distance	0.011
2	Timestamp	0.346	Dst. Lat.	0.231	Dst. Addr.	0.110	Distance	0.029	Src. Lat.	0.023	Src. Lat.	0.010
3	Src. Addr.	0.156	Src. Lat.	0.129	Dst. Lat.	0.079	Hop count	0.021	Dst. Long.	0.007	Provider	0.001
4	Dst. Addr.	0.119	Src. Long.	0.127	Src. Lat.	0.023	Src. Lat.	0.019	Src. ASN	0.004	Src. Long.	0.001
5	Dst. Long.	0.045	Src. ASN	0.098	Provider	0.001	Src. Addr.	0.008	Dst. ASN	0.001	Src. ASN	0.000

TABLE XIII
LIME AND SHAP WEIGHTS FOR XGBOOST FOR THE LITERATURE DATASETS. USERS IN EUROPE TO CLOUD SITES IN FR & SP.

	LIME						SHAP					
	CorneoWebC21		CandelaAcc19		MohanHNets19		CorneoWebC21		CandelaAcc19		MohanHNets19	
Order	*Feature*	*Weight*	*Feature*	*Weight*	*Feature*	*Weight*	*Feature*	*Weight*	*Feature*	*Weight*	*Feature*	*Weight*
1	Src. Long.	0.320	Src. ASN	0.127	Distance	0.355	Distance	0.062	Src. Lat.	0.017	Src. Long.	0.021
2	Distance	0.083	Distance	0.093	Src. Long.	0.103	Src. Addr.	0.053	Src. Long.	0.015	Src. Lat.	0.006
3	Dst. Addr.	0.072	Dst. Addr.	0.078	ProbeIp	0.039	Src. Long.	0.011	Src. ASN	0.008	Src. ASN	0.003
4	Timestamp	0.043	Dst. ASN	0.075	Src. ASN	0.035	Hop count	0.009	Dst. Addr.	0.007	ProbeIp	0.002
5	Dst. Lat.	0.025	Src. Addr.	0.067	Src. Lat.	0.032	Dst. Addr.	0.006	Distance	0.007	Distance	0.001

a) Setup Heterogeneity: Each dataset presents distinct characteristics: The datasets consider different platforms (RIPE Atlas, NLNOG Ring), measurement types (ping or traceroute), cloud nodes (CSPs, RIPE Atlas anchors), measurement granularities and temporal ranges.

b) Temporal bias and limitation: Cloud-network architectures evolve continuously; thus, comparing measurements across time introduces an inherent and largely unquantifiable bias. The works selected for comparison [7]–[9] date from 2019–2021 and represent the most recent studies assessing cloud latency that are comparable to our campaign. Crucially, this temporal bias is analogous to the more intuitive spatial bias: just as geographically distinct datasets capture different network topologies, temporally separated datasets reflect different infrastructure states. Direct cross-dataset comparisons are therefore inherently limited. Instead, we apply a consistent methodology across all datasets and draw conclusions about general trends rather than making absolute cross-dataset claims. The absence of more recent large-scale cloud latency measurement studies makes this comparison the most rigorous currently possible, and further underscores the value of the present work as an up-to-date reference for the community.

In addition, the measurement duration (3 months) restricts the findings to that limited period of time. We may miss long-term seasonal variations or low-probability network anomalies requiring a longer time span. However, it is an adequate period for our specific goals of characterizing and predicting daily and weekly behavior, as detailed in Section III-B3.

c) RIPE Atlas bias: Our experiment relies on RIPE Atlas, which, as discussed in Section II-A2, has some limitations. Most notably, its probes tend to have high-quality, stable connectivity: only two Wi-Fi probes and no cellular (2G–5G) probes were found in Spain. Hence, our work does not explore the actual last-mile diversity, and extending this methodology to more volatile access scenarios (e.g., 5G networks) represents a natural and important extension of this work.

d) Dataset format: Each dataset is publicly available in a particular format that may not easily map to the format of the other datasets. For example, `CandelaAcc19` provides only links to the RIPE Atlas measurement IDs, so the data must be retrieved by logging in to RIPE Atlas and downloading the data from more than 8000 URLs [7]. `CorneoWebC21` provides the data in SQL format and contains tables with cross-referenced fields [9], whereas our dataset is stored in CSV files and can also be downloaded from RIPE Atlas. The dataset sizes also differ, ranging from 1.6 GB (`MohanHNets19`) to 70 GB (`CorneoWebC21`).

B. Main Insights

We split this discussion into two subsections: one on forecasting evaluation and one on explainability.

1) Forecasting: Tree-based ensemble algorithms, particularly XGBoost and Random Forest, routinely outperform other approaches for predicting spatial delay. In the `MohanHNets19` dataset, XGBoost reduces NRMSE from 0.59 to 0.30 and RMSE from 58.66 ms to 29.97 ms relative to the linear model. In our dataset, it reduces NRMSE from 0.53 to 0.41 and RMSE from 5.79 ms to 4.46 ms. Although individual decision trees do not always outperform linear models, ensembles maintain a clear advantage. This advantage is greater in large, heterogeneous datasets but remains under the more homogeneous conditions of our campaign. These results agree with previous findings on tree-based models for tabular data [56] and they offer network operators useful advice: *Implementing ensemble tree-based models can result in significant improvements in latency prediction accuracy, supporting improved resource allocation and user experience in cloud services.*

One of the crucial advantages of simpler models is that their execution time is considerably shorter, which also implies a lower computation cost. Therefore, if simpler models achieve

TABLE XIV
EXECUTION TIME BY FORECASTING METHOD (20 RUNS, IN SECONDS)

Method	Temporal	Only distance	All features
Naive forecast	0.02	–	–
SES	0.38	–	–
Holt's linear	14.24	–	–
AutoARIMA	90.39	–	–
Prophet	469.48	–	–
LSTM	432.14	–	–
Linear model	–	0.16	3.71
Decision tree	–	3.28	54.41
Random forest	–	27.04	402.67
XGBoost	–	10.08	17.35
KNN	–	971.47	919.10
SVM	–	74.29	2657.70
GWN	–	–	3360.28

an accuracy at least equivalent to that of more complex (and therefore slower and more expensive) models, then the ISP will be able to deploy these solutions at a fraction of the cost. Table XIV shows the time required to run 20 iterations of each forecasting algorithm, on the same data. The actual absolute values are not relevant here, as they are dependent on the hardware used; instead, we are interested in the relative runtime performance of the algorithms. We observe that execution times differ by several orders of magnitude. More importantly, the best-performing models (XGBoost, Random Forest, Naive, and AutoARIMA) are among the fastest models.

To verify that our findings also apply to other critical metrics beyond average latency, such as worst-case latency, we performed the temporal and spatial forecasting experiments to predict the worst RTT among the traceroute messages sent in each 30-min interval. We observe that the results remain accurate and in the same order of magnitude: For temporal forecasting, the best avg.-RTT 3-hour-ahead prediction has an RMSE of 1.867 ms, with a 95% prediction interval of 3.695 ms, whereas for the max.-RTT prediction the best RMSE is 1.804 ms, with a 95% prediction interval of 2.642 ms. For spatial forecasting, we observe that the best algorithms for max.-RTT produce only slightly higher RMSE values than the best algorithms for average RTT. Specifically, for the max.-RTT only-distance case, the decision tree model obtains an RMSE 30% higher than that of Random Forest for the corresponding avg.-RTT only-distance case (6.5 ms vs 4.984 ms); for the max.-RTT all-features case, XGBoost obtains an RMSE 15% higher than XGBoost for the corresponding avg.-RTT all-features case (5.136 ms vs 4.461 ms). These results show that the observed conclusions extend to worst-case scenarios such as the maximum-RTT case.

For time-series forecasting, simple methods such as naive forecasting, SES, and Holt's linear method provide accurate predictions. Prophet and LSTM might achieve similar performance with dedicated tuning, but optimizing individual methods lies outside this work's scope. Moreover, when simple methods already achieve errors of only a few milliseconds, more complex methods do not justify their additional computational cost. There is a significant difference in the accuracy of the methods between our dataset and those from the literature. The key to interpreting this difference probably lies in the fact that our measurement campaign comprises probe–server pairs that experience homogeneous conditions, since we focus on a reduced geographical area and a single IO. Other studies, in turn, combine heterogeneous populations without recognizing that the observed variability stems from this heterogeneity rather than from temporal fluctuations.

2) Explainability: LIME presents more consistent explanations across datasets than SHAP. One reason for this difference is that the way SHAP works, by evaluating all possible combinations of features and substituting the remaining ones with random values [48, Ch. 18], removes the *locality* of LIME's explanations but can also be detrimental inasmuch as the randomly selected values may lead to unrealistic samples that obscure the correlation between features.

We find that the most important feature is the distance between source and destination, as already illustrated in Section IV, followed by source-related features (its Longitude, Latitude, IP Address or ASN), while the destination-related features do not carry as much weight. Specifically, for LIME, Distance is the most important feature across both `ping` and `traceroute` datasets, whereas for SHAP the Hop count is the most important feature for the `traceroute` dataset (`CorneoWebC21`), while the source-related features take the lead on the `ping` datasets (`CandelaAcc19` and `MohanHNets19`). Thus, for `CorneoWebC21`, SHAP and LIME are aligned in their explanations (note that the Hop count is naturally highly correlated with distance, which also ranks second for SHAP). The different behavior of the SHAP explanations for the `ping` datasets may be related to the mentioned unrealistic sample creation, and it is worth investigating in the future.

Considerations on using post-hoc explanations: Post-hoc methods explain how a selected model uses its features, not why latency behaves as observed. Consistent explanations may indicate an underlying, possibly nonlinear, relationship between important features and latency, but they require careful interpretation. In particular, LIME and SHAP do not explicitly account for correlations among features [52]. We therefore compute Pearson correlations between every pair of features, including latency, as shown in Figs. 11–13.

In the `MohanHNets19` dataset, 12 of the 66 feature pairs show moderate correlations between 0.4 and 0.6. The `CandelaAcc19` dataset shows weaker correlations overall, although its destination-related features correlate noticeably with one another. Across all three datasets, distance and latency have a correlation of approximately 0.75, consistent with the value of 0.78 obtained for our dataset in Fig. 5.

The results obtained for NMR, together with the considerable increase in RMSE when distance is removed from the features, show that SHAP and LIME results regarding the importance of distance as a predictor are robust and not materially affected by the feature correlations. Nevertheless, the limitations of post-hoc XAI methods remain an open research challenge.

C. General implications for ISPs and users

1) Edge Deployment as a Necessity: As shown in Section IV and Fig. 5, edge infrastructure is necessary to provide

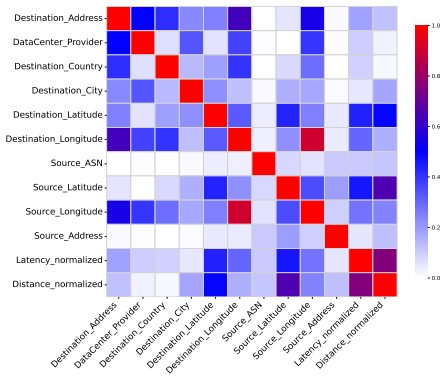

Fig. 11. Correlation matrix for Mohan [8]

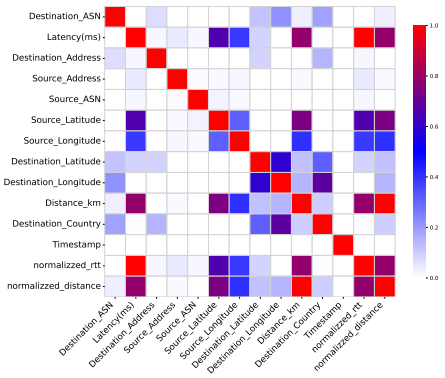

Fig. 12. Correlation matrix for Candela [7]

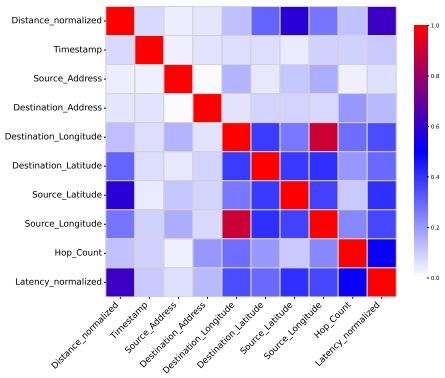

Fig. 13. Correlation matrix for Corneo [9]

high QoS homogeneously across a mid-sized country, as hundreds of kilometers are already sufficient to exceed the stringent 20-ms latency requirement of emerging applications.

2) Service Provision: For the geographical and temporal scope considered, user-specific cloud-access latency remains stable, contrary to prior studies suggesting substantial temporal variability. This stability enables ISPs and CSPs to determine whether a latency-sensitive service can provide adequate QoS before initiation and prevent users from experiencing poor performance when it cannot. Distance and a few readily available features provide accurate predictions without complex methods, and latency typically remains stable for a considerable period after shifting to a new level.

3) Edge Placement and MANO: Our explainability analysis shows that geographic distance strongly predicts latency (cf. Table VIII). Combined with high temporal stability, this finding supports simple distance-based rules for initial edge-server placement. In multi-tenant MANO environments with auction-based resource allocation [57], however, measured latency represents a lower bound that operators must balance against competition among tenants, pricing, and distributed-infrastructure costs. The importance of distance also supports mobility-aware resource management, in which edge placement and allocation adapt to user mobility to minimize access distance [58]. Spatial latency forecasts could further inform delay-sensitive orchestration; for example, age-of-information-aware VNF scheduling could use distance-based estimates to optimize where and when functions are instantiated along the service chain [59].

VIII. Conclusions

Evaluating cloud-service latency is an ever-evolving challenge. Hence, performance snapshots should be combined to gain a comprehensive understanding of the problem. Our contribution is twofold. First, our campaign measures latency from major CSPs to end users of a leading Spanish IO across 256 cloud–user pairs at 30-min granularity. We find that latency remains stable and that path changes drive most variations. Placing cloud servers at the ISP's network edge reduces latency, although the values remain comparable. The findings about latency stability and predictability are limited to the geographical and temporal scope of the measurements. In addition, the top-connectivity bias present in devices connected to the RIPE Atlas network contributes to latency stability, and latency for real user devices should also account for the inherent variability of last-mile and wireless links. Nevertheless, access-link latency and transport–core latency are complementary components. Many studies focus on the former, whereas real-world measurements of the latter are less common, as discussed in Section II. The multi-ISP campaign confirms that our findings are generalizable and do not suffer from the single-ISP restriction. Our dataset offers an updated snapshot of CSPs' connectivity for emerging applications.

Second, our evaluation of available cloud-latency datasets provides the first such cross-dataset comparative analysis. Despite dataset-specific characteristics, distance consistently emerges as the most important factor, advancing our understanding of cloud-service performance and the features relevant to latency characterization and forecasting.

As future work, enriching the feature set with finer-grained topological proxies (such as AS-path distance, inferred Internet exchange presence, last-mile access type, and temporal covariates like BGP stability indicators) could refine the explainability analysis and improve accuracy in heterogeneous scenarios.

Acknowledgments

We acknowledge RIPE Atlas for providing us with access to their platform, and Christian Elsen for maintaining the public RIPE Atlas probes located at different AWS regions [60].

References

[1] R. Ingabire, A. Bazco-Nogueras, V. Mancuso, L. M. Contreras, and J. Folgueira, "Clearing clouds from the horizon: Latency characterization of public cloud service platforms," in *Proc. IEEE Int. Conf. Comput. Commun. & Netw. (ICCCN)*, 2024.

[2] A. Li, X. Yang, S. Kandula, and M. Zhang, "CloudCmp: Comparing public cloud providers," in *Proc. ACM Internet Meas. Conf. (IMC)*, 2010.

[3] F. Palumbo, G. Aceto, A. Botta, D. Ciuonzo, V. Persico, and A. Pescapé, "Characterization and analysis of cloud-to-user latency: The case of Azure and AWS," *Comput. Netw.*, vol. 184, p. 107693, 2021.

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

[5] Y. Jin, S. Renganathan, G. Ananthanarayanan, J. Jiang, V. N. Padmanabhan, M. Schroder *et al.*, "Zooming in on wide-area latencies to a global cloud provider," in *Proc. ACM SIGCOMM Conf.*, 2019, p. 104–116.

[6] O. Hilyard, B. Cui, M. Webster, A. Muralikrishna, and A. Charapko, "Cloudy forecast: How predictable is communication latency in the cloud?" in *Proc. IEEE Int. Conf. Comput. Comm. Netw. (ICCCN)*, 2024.

[7] M. Candela, E. Gregori, V. Luconi, and A. Vecchio, "Using RIPE atlas for geolocating IP infrastructure," *IEEE Access*, vol. 7, pp. 48 816–48 829, 2019.

[8] N. Mohan, L. Corneo, A. Zavodovski, S. Bayhan, W. Wong, and J. Kangasharju, "Pruning edge research with latency shears," in *Proc. ACM Workshop Hot Topics Netw. (HotNets)*, 2020, pp. 182–189.

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

[10] T. Koch, W. Jiang, T. Luo, P. Gigis, Y. Zhang, K. Vermeulen *et al.*, "Towards a traffic map of the internet: Connecting the dots between popular services and users," in *Proc. ACM Workshop Hot Topics Netw. (HotNets)*, 2021, p. 23–30.

[11] RIPE NCC, "RIPE Atlas." [Online]. Available: https://atlas.ripe.net

[12] R. Ingabire, A. Bazco-Nogueras, V. Mancuso, L. M. Contreras, and J. Folgueira, "source code," 2026. [Online]. Available: https://github.com/rita-imdea/cloud_latency_xai

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

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

[15] PlanetLab, 2024, https://planetlab.cs.princeton.edu/.

[16] O. Tomanek, P. Mulinka, and L. Kencl, "Multidimensional cloud latency monitoring and evaluation," *Comput. Netw.*, vol. 107, pp. 104–120, 2016.

[17] R. K. P. Mok, H. Zou, R. Yang, T. Koch, E. Katz-Bassett, and K. C. Claffy, "Measuring the network performance of google cloud platform," *Proc. ACM Internet Meas. Conf. (IMC)*, 2021.

[18] P. Gill, C. Diot, L. Y. Ohlsen, M. Mathis, and S. Soltesz, "M-lab: User initiated internet data for the research community," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 52, no. 1, pp. 34–37, 2022.

[19] S. Sundaresan, S. Burnett, N. Feamster, and W. De Donato, "BISmark: A testbed for deploying measurements and applications in broadband access networks," in *Proc. USENIX Annu. Tech. Conf. (ATC)*, 2014.

[20] CAIDA, "Archipelago (Ark) measurement infrastructure," 2024, accessed on Nov. 2025. [Online]. Available: www.caida.org/projects/ark/

[21] L. Corneo *et al.*, "(How much) can edge computing change network latency?" in *Proc. IFIP Netw. Conf.* IEEE, 2021.

[22] R. Fontugne, A. Shah, and K. Cho, "Persistent last-mile congestion: not so uncommon," in *Proc. ACM Internet Meas. Conf. (IMC)*, 2020.

[23] N. Martin and F. Dogar, "Divided at the edge - measuring performance and the digital divide of cloud edge data centers," *Proc. ACM Netw.*, vol. 1, no. CoNEXT3, Nov. 2023.

[24] A. Bhat, V. Ganatra, A. Shaha, B. Chandrasekaran, and V. Naik, "On the constancy of latency at the internet's edge," in *Proc. Netw. Traffic Meas. & Anal. Conf. (TMA)*. IEEE, 2025.

[25] O. Victor Babasanmi and J. Chavula, "Measuring cloud latency in Africa," in *Proc. IEEE Int. Conf. Cloud Netw. (CloudNet)*, 2022.

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

[27] P. Sermpezis, L. Prehn, S. Kostoglou, M. Flores, A. Vakali, and E. Aben, "Bias in internet measurement platforms," in *Proc. Netw. Traffic Meas. and Analysis Conf. (TMA)*, 2023.

[28] J. Koumar, T. Smoleň, K. Jeřábek, and T. Čejka, "Comparative analysis of deep learning models for real-world ISP network traffic forecasting," *IEEE Trans. Netw. & Service Manag.*, vol. 23, pp. 715–728, 2026.

[29] S. Moghadas, C. Fiandrino, A. Collet, G. Attanasio, M. Fiore, and J. Widmer, "Spotting deep neural network vulnerabilities in mobile traffic forecasting with an explainable ai lens," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*. IEEE, 2023.

[30] Y. Gebreyesus, D. Dalton, D. De Chiara, M. Chinnici, and A. Chinnici, "Ai for automating data center operations: model explainability in the data centre context using shapley additive explanations (shap)," *Electronics*, vol. 13, no. 9, p. 1628, 2024.

[31] V. Viradia, A. Jain, S. S. Ogety, and A. Donvir, "Explainable ai (xai) in predictive cloud optimization: Cost, workload and performance," *SoutheastCon 2025*, pp. 1424–1429, 2025.

[32] T. Sim, S. Choi, Y. Kim, S. H. Youn, D.-J. Jang, S. Lee *et al.*, "eXplainable AI (XAI)-based input variable selection methodology for forecasting energy consumption," *Electronics*, vol. 11, no. 18, 2022.

[33] NLNOG Ring, accessed Nov. 2025. [Online]. Available: ring.nlnog.net/

[34] O. Alay *et al.*, "Measuring and assessing mobile broadband networks with MONROE," in *Proc. IEEE Int. Symp. World Wireless, Mobile & Multim. Netw. (WoWMoM)*, 2016.

[35] RIPE NCC, accessed Apr. 2025. [Online]. Available: www.ripe.net/

[36] RIPE Atlas, "Coverage and statistics," 2024, accessed May 2025. [Online]. Available: https://atlas.ripe.net/coverage/

[37] P. Gigis, V. Kotronis, and E. Aben, "RIPE Atlas coverage in eyeball networks," 2017, accessed May 2025. [Online]. Available: https://labs.ripe.net/author/petros_gigis/ripe-atlas-coverage-in-eyeball-networks/

[38] R. Kistel, "RIPE Atlas architecture how we manage our probes," 2023, accessed May 2025. [Online]. Available: https://labs.ripe.net/author/kistel/ripe-atlas-architecture-how-we-manage-our-probes/

[39] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *J. Am. Stat. Assoc.*, vol. 74, no. 366, pp. 427–431, 1979.

[40] D. Kwiatkowski, P. C. Phillips, P. Schmidt, and Y. Shin, "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?" *J. Econometrics*, vol. 54, no. 1, pp. 159–178, 1992.

[41] J. Lee and M. C. Strazicich, "Minimum lagrange multiplier unit root test with two structural breaks," *Review Econ. & Stat.*, vol. 85, no. 4, pp. 1082–1089, 11 2003.

[42] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *Annals Math. Stat.*, pp. 50–60, 1947.

[43] M. B. Brown and A. B. Forsythe, "Robust tests for the equality of variances," *J. Am. Stat. Assoc.*, vol. 69, no. 346, pp. 364–367, 1974.

[44] 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 Meas. Conf. (IMC)*, 2023.

[45] S. J. Taylor and B. Letham, "Forecasting at scale," *The Am. Stat.*, vol. 72, no. 1, pp. 37–45, 2018.

[46] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2019, p. 1907–1913.

[47] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong, "Interpretable machine learning: Fundamental principles and 10 grand challenges," *Statistics Surveys*, vol. 16, pp. 1–85, 2022.

[48] C. Molnar, *Interpretable Machine Learning*, 3rd ed. Lulu.com, 2025. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[49] 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. Knowl. Discov. & Data Mining*, 2016, pp. 1135–1144.

[50] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.

[51] L. S. Shapley, "A value for n-person games," *Contribution to the Theory of Games*, vol. 2, 1953.

[52] A. M. Salih *et al.*, "A perspective on explainable artificial intelligence methods: SHAP and LIME," *Adv. Intell. Syst.*, vol. 7, no. 1, 2025.

[53] L. M. Paes, D. Wei, and F. P. Calmon, "Selective explanations," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 37, 2024.

[54] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair *et al.*, "From local explanations to global understanding with explainable AI for trees," *Nature Machine Intell.*, vol. 2, no. 1, pp. 56–67, Jan. 2020.

[55] S. Lundberg, E. Dillon, J. LaRiviere, J. Roth, and V. Syrgkanis, "Be careful when interpreting predictive models in search of causal insights," 2021, [Online] Available: https://towardsdatascience.com.

[56] S. Uddin and H. Lu, "Confirming the statistically significant superiority of tree-based machine learning algorithms over their counterparts for tabular data," *PLOS ONE*, vol. 19, no. 4, Apr. 2024.

[57] M. R. Abedi, M. Fasanghari, M. Akbari, N. Mokari, and H. Yanikomeroglu, "Dynamic pricing in multi-tenant mano with resource sharing: A stackelberg game approach," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 7002–7021, 2024.

[58] M. R. Abedi, N. Mokari, M. R. Javan, H. Saeedi, E. A. Jorswieck, and N. Zorba, "Low complexity and mobility-aware robust radio, storage, computing, and cost management for cellular vehicular networks," *IEEE Trans. Vehic. Technol.*, vol. 74, no. 2, pp. 3327–3344, 2024.

[59] M. Akbari, M. R. Abedi, R. Joda, M. Pourghasemian, N. Mokari, and M. Erol-Kantarci, "Age of information aware vnf scheduling in industrial iot using deep reinforcement learning," *IEEE J. Selected Areas Commun.*, vol. 39, no. 8, pp. 2487–2500, 2021.

[60] C. Elsen, "RIPE Atlas probes in AWS," accessed May 2025. [Online]. Available: https://github.com/chriselsen/RIPE-Atlas-in-AWS