Horizon: Understanding and Predicting Global Starlink Performance

CRISTIAN BENGHE*, TU Delft, The Netherlands
VLAD GRAURE*†, TU Delft, The Netherlands
TANYA SHREEDHAR, TU Delft, The Netherlands
NITINDER MOHAN, TU Delft, The Netherlands

Starlink has deployed over 7,800 satellites serving millions of subscribers, yet predicting its performance remains an open challenge. Rapid orbital dynamics, frequent handovers, and weather-induced signal attenuation create variability that existing models, built on a handful of instrumented terminals in limited regions, cannot capture at global scale. We present `Horizon`, the first global-scale machine learning system for predicting LEO satellite Internet performance. Our key insight is that crowdsourced measurement platforms, while noisier than controlled experiments, provide the geographic diversity necessary to build globally generalizable models. `Horizon` integrates 11 months of measurements from M-Lab and Cloudflare spanning 90+ countries with meteorological data and satellite orbital propagation features. On a fully held-out one-week temporal window, `Horizon` achieves mean absolute errors of 17.76 ms for latency and 25.63 Mbps for throughput; on a standard 80/20 split it outperforms all baselines, including adaptations of state-of-the-art architectures. Feature importance analysis reveals that geographic position dominates prediction, with latitude alone contributing 42–46%, while weather features account for 14–15%, quantifying the impact of atmospheric conditions on Ku/Ka-band links. Leave-one-location-out experiments confirm that `Horizon` generalizes to regions absent from training, enabling performance estimation where measurement infrastructure does not yet exist. Our dataset and pipeline are publicly available, providing a foundation for global LEO network performance visibility.

CCS Concepts: • **Networks** → **Network performance modeling**; **Network performance analysis**.

Additional Key Words and Phrases: LEO Internet, Crowdsourced Telemetry, Performance Prediction, Starlink, Machine Learning

ACM Reference Format:
Cristian Benghe, Vlad Graure, Tanya Shreedhar, and Nitinder Mohan. 2026. Horizon: Understanding and Predicting Global Starlink Performance . *Proc. ACM Meas. Anal. Comput. Syst.* 10, 2, Article 41 (June 2026), 30 pages. https://doi.org/10.1145/3805639

1 Introduction

The global demand for reliable Internet connectivity continues to grow, yet a large population of users in rural, remote, and underserved regions remain poorly connected because deploying terrestrial infrastructure is often economically infeasible at the "last mile" and across sparsely populated areas [24, 32]. Satellite Internet has long been viewed as a coverage solution, but traditional geostationary (GEO) systems incur inherently high round-trip delay due to the orbital altitude, yielding

*Both authors contributed equally to this research.
†Now employed at Google.

Authors' Contact Information: Cristian Benghe, C.Benghe@student.tudelft.nl, TU Delft, The Netherlands; Vlad Graure, vladgraure@yahoo.ca, TU Delft, The Netherlands; Tanya Shreedhar, t.shreedhar@tudelft.nl, TU Delft, The Netherlands; Nitinder Mohan, N.Mohan@tudelft.nl, TU Delft, The Netherlands.

ACM 2476-1249/2026/6-ART41
https://doi.org/10.1145/3805639

baseline RTTs on the order of hundreds of milliseconds, making interactive applications (e.g., video conferencing, gaming, and real-time applications) noticeably degraded. Low-Earth-Orbit (LEO) satellite constellations have emerged as a promising alternative, operating at altitudes between 160 and 2,000 km to deliver substantially lower latency while maintaining a wide-area coverage. Among LEO broadband providers, Starlink [47] has achieved the largest operational deployment, with over 7,800 active satellites in the orbital constellation serving millions of subscribers across more than 140 countries. Other providers, including Amazon LEO [2] and Eutelsat OneWeb [11], are rapidly expanding their constellations, signaling a fundamental shift in how Internet connectivity will be delivered to regions where terrestrial build-out is slow or cost-prohibitive.

Despite the promise of LEO satellite Internet, *the key operational challenge is not coverage, but predictability*, as the end-to-end performance is highly variable and difficult to forecast. Satellites move at ≈ 7.6 km/s, forcing user terminals to execute handovers every 15 seconds as satellites enter and exit view [16]. Simultaneously, ground infrastructure evolves continuously as new ground stations and points of presence alter routing paths, shifting which servers users reach and through what egress points [6]. Weather compounds the variability further. The majority of the LEO operators use high-frequency bands between user-terminal-satellite-ground station for high-throughput low-latency communication (e.g. Starlink uses Ku/Ka-band). These frequencies are highly susceptible to atmospheric attenuation, with studies documenting up to 50% throughput reduction during heavy precipitation [25]. Together, these dynamics create spatio-temporal heterogeneity that users cannot anticipate and operators struggle to plan around. End users cannot reliably estimate whether their connection will support a video call or gaming session at a given time and application developers cannot adapt their systems to expected network conditions [30]. The gap between Starlink's coverage and its predictability limits its utility precisely where LEO connectivity could have the highest societal impact.

Prior research has made important strides in characterizing LEO satellite performance. Measurement studies have documented the impact of weather on Starlink connectivity, with Kassem et al. [17] reporting noticeable increases in page load times during rain events and Ma et al. [25] observing a throughput reduction of up to 50% during heavy precipitation. Subsequent work has explored predictive models for download latency [20, 50] and throughput [20, 23, 50], demonstrating that machine learning approaches can capture some of the complex relationships between environmental factors and network performance. However, these efforts share a fundamental limitation: they rely on measurements from a handful of professionally instrumented terminals deployed in limited geographic regions, predominantly concentrated in Europe and North America. The cost and logistical complexity of deploying dedicated measurement infrastructure across diverse climatic zones and geographic conditions has constrained the generalizability of existing models. As a result, current prediction approaches offer limited insight into Starlink performance in underrepresented regions, including much of Africa, South America, and Asia, where LEO connectivity may have the highest societal impact.

In this paper, we present `Horizon`, a global-scale machine learning framework for understanding and predicting Starlink network performance. Our key insight is that crowdsourced measurement platforms, while noisier than controlled experiments, provide the geographic diversity and longitudinal coverage necessary to build globally generalizable prediction models. `Horizon` integrates heterogeneous performance measurements from M-Lab's `NDT7` [27] and Cloudflare `AIM` [51], spanning over 11 months of data from users across six continents (90+ countries). We enrich these measurements with high-resolution weather data, satellite density estimates derived from orbital propagation of Two-Line Element (`TLE`) data, and carefully engineered spatio-temporal features. To address the inherent noise in crowdsourced data, we develop location-aware anomaly filtering strategies and server selection heuristics that mitigate routing biases while preserving the natural

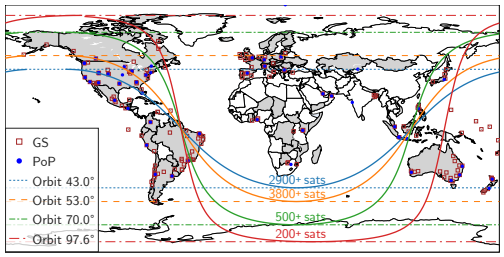


Fig. 1. Starlink infrastructure: GSs, PoPs and four orbital shells together with their respective satellite count [56], as of January 2026. Countries with Starlink availability are shown in grey.

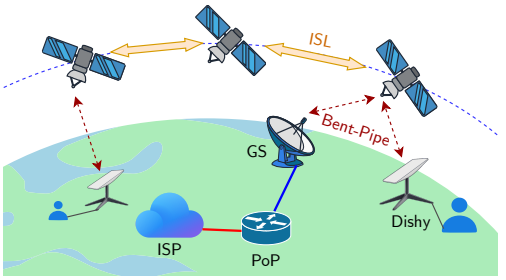


Fig. 2. Bent-pipe architecture of Starlink. One or more satellites, connected via inter-satellite links (ISLs), establish the connection between the user terminal and the ground station.

variability in Starlink performance. We train an interpretable ensemble model combining Random Forest and Gradient Boosting regressors, enabling us to quantify feature importance and understand the factors driving performance predictions. Specifically, we make the following contributions:

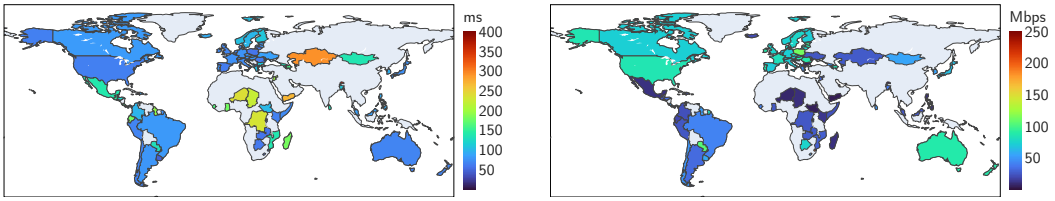
(1) We design and implement a unified data collection system that integrates crowdsourced measurements from `NDT7` and Cloudflare `AIM`. The pipeline applies location-aware server filtering to remove measurements affected by routing anomalies and validates cross-dataset compatibility using Jensen-Shannon divergence analysis, processing over 11 months of Starlink measurements spanning diverse geographic regions and climatic conditions (§3.1).
(2) We enrich the network measurements with dense meteorological features (temperature, precipitation, wind speed, and cloud cover) from the `OpenMeteo` API, satellite density estimates computed via `SGP4` orbital propagation and spatio-temporal encodings. A novel Weather Index combines these meteorological features using Partial Least Squares regression that reduces dimensionality while preserving predictive power (§3.3).
(3) An ensemble model combining Random Forest and Gradient Boosting regressors predicts Starlink download latency and throughput at fine spatial granularity. We systematically compare three anomaly filtering strategies: percentile-based, directional median absolute deviation (MAD), and Isolation Forest. We also analyze the impact of training data duration on model performance and demonstrate generalization to unseen geographic locations through leave-one-location-out experiments (§3.5).
(4) `Horizon` can be deployed as a continuously updated system that generates hourly performance forecasts for over 300 `H3` hexagons covering regions where Starlink operates. The system incorporates real-time weather forecasts and adapts to evolving constellation dynamics through periodic model retraining, enabling network planners and end users to anticipate service quality under varying conditions (§4).

Together, these contributions establish the *first global-scale, machine learning-driven telemetry and prediction system for LEO satellite Internet.* `Horizon` bridges the gap between localized measurement studies, which offer precision but limited coverage, and the practical need for worldwide performance visibility. Our dataset [3] and the `Horizon` pipeline [4] are publicly available to facilitate reproducibility and enable future research on LEO satellite network performance.

2 Background and Related Work

2.1 Background

Starlink, operated by SpaceX, is the largest operational LEO satellite Internet constellation, with ≈7,800 active satellites as of early 2026 [46]. The constellation spans four orbital shells at inclinations



a) Median download latency (ms) per country. b) Median download throughput (Mbps) per country.

Fig. 3. Global variations in Starlink network performance for September 2025, combining M-Lab NDT7 and Cloudflare AIM measurements. North America and Europe exhibit lower latency and higher throughput compared to central Africa and parts of Asia.

of 43°, 53° (including 53.2°), 70°, and 97.6°, with the majority of satellites operating in the 53° and 43° shells [35, 48, 56]. Figure 1 illustrates these shells along with the active Points of Presence (PoPs) and ground stations (GSs), and the set of 140 countries where Starlink service is available as of January 2026. To provide Internet connectivity, Starlink employs a "bent-pipe" architecture, as shown in fig. 2. Clients connect to a satellite using a user terminal (commonly known as "Dishy"), which may relay data through additional satellites via inter-satellite links (ISLs) before reaching a ground station. The satellite selection and handover process is managed by a global scheduler that considers satellite visibility, orbital dynamics, and network load [30, 49]. The high velocity of LEO satellites (≈ 7.6 km/s) results in rapid satellite motion and frequent handovers, with user-satellite associations updated every 15 seconds [16]. Each satellite has a geometric visibility window of roughly 12 minutes, though effective connectivity periods for user terminals are typically shorter. Ground stations connect to the broader Internet via PoPs distributed globally, which are continuously evolving as new infrastructure is deployed.

Starlink performance varies significantly across geographic regions, driven by differences in satellite density, ground infrastructure availability, and environmental conditions. Figure 3a and Figure 3b illustrate this variability, showing the median download latency and throughput reported by Starlink customers globally for September 2025.[1] The maps reveal substantial differences in connectivity quality across countries: North America exhibits higher throughput and lower latency compared to central African nations. Significant performance disparities are also visible within continents, as southern African countries consistently outperform their central counterparts. Multiple factors contribute to this geographic heterogeneity, including overhead satellite density [30], routing path complexity [37], and infrastructure maturity. For example, the deployment of several Starlink PoPs in Africa starting in January 2025 altered routing paths and improved performance for multiple countries in the region [6].

Weather conditions have emerged as a key driver of performance degradation in LEO satellite networks. Starlink operates in the Ku-band (12–18 GHz) and Ka-band (26.5–40 GHz) frequencies, which are susceptible to rain fade and atmospheric attenuation [53]. Rain, cloud cover, and atmospheric disturbances can significantly attenuate satellite signals, creating location-dependent and time-dependent variations in throughput and latency. These environmental factors motivate the incorporation of weather context into performance analysis and prediction models.

2.2 Related Work

Prior work on LEO satellite performance can be broadly categorized into measurement-based characterization studies and machine learning prediction models. Early measurement research on

[1]These maps combine measurements from Cloudflare AIM and M-Lab NDT7 datasets. Refer to §3 for our methodology.

Starlink focused on benchmarking throughput and latency against terrestrial broadband, demonstrating its potential to deliver low-latency connectivity in underserved regions. Michel et al. [28] provided one of the first comprehensive characterizations of Starlink performance, measuring latency, throughput, and packet loss across multiple vantage points. Mohan et al. [30] conducted a multifaceted analysis of Starlink's network architecture and performance, revealing the complexity of its connectivity patterns and routing behavior. Pan et al. [37] measured LEO satellite network performance from multiple geographic locations, highlighting the substantial variability in user experience across regions.

Several studies have quantified the impact of weather on Starlink performance. Kassem et al. [17] conducted a browser-based measurement campaign and reported noticeable increases in page load times during rain events. Ma et al. [25] observed up to 50% throughput reduction during heavy precipitation, particularly for download traffic. Laniewski et al. [21] performed controlled experiments correlating weather conditions with Starlink performance metrics, confirming the significant impact of precipitation on connectivity quality. Predictive models for Starlink performance have also been explored. `T3P` [50] proposed a throughput prediction framework that leverages historical measurements and satellite orbital information. Lanfer et al. [20] developed models incorporating weather data to forecast download latency and throughput, achieving improved prediction accuracy compared to weather-agnostic baselines. In [23], the authors analyzed the relationship between network conditions and user-perceived quality, informing prediction model design.

A shared limitation across much of this literature is coverage and representativeness. Many efforts rely on a small number of professionally instrumented terminals typically deployed in controlled environments and predominantly located in Europe and North America which are regions with mature ground infrastructure. This constrains climatic and infrastructural diversity and limits generalizability. Our work addresses these limitations by leveraging crowdsourced measurement platforms that provide global coverage. Rather than relying on controlled terminals, we integrate heterogeneous measurements from M-Lab's `NDT7` and Cloudflare `AIM`, which collectively provide dense performance coverage spanning Starlink customers across six continents. We combine these measurements with dense weather features, satellite density estimates, and spatio-temporal encodings to build prediction models that generalize across diverse geographic and climatic conditions.

3 `Horizon` Pipeline

This section presents the `Horizon`, an end-to-end system for collecting, processing, and predicting Starlink[2] network performance at a global scale. Figure 4 summarizes the five-phase architecture of the pipeline. In ❶, we collect heterogeneous data from crowdsourced measurement platforms, weather APIs, and satellite orbital databases. ❷ applies pre-processing steps including data cleaning, server filtering to mitigate systematic routing biases, and distribution comparison to validate cross-dataset compatibility. ❸ enriches the measurements with spatio-temporal features, a composite Weather Index, and satellite density estimates derived from orbital propagation. ❹ applies location-aware anomaly detection to remove outliers while preserving the natural variability in Starlink performance. Finally, ❺ trains an interpretable ensemble model combining Random Forest and Gradient Boosting regressors to predict download latency and throughput.

[2]In this work, we primarily focus on Starlink due to its extensive global deployment. However, `Horizon` can be readily adapted to support other satellite operators as their networks mature, such as Amazon Leo or Eutelsat OneWeb, provided enough measurements are available in NDT7 and Cloudflare AIM datasets

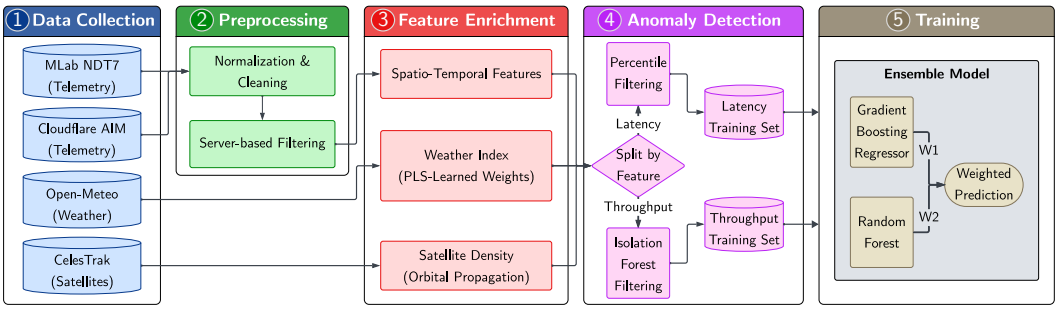


Fig. 4. Overview of the Horizon pipeline architecture. 1 collects crowdsourced measurements, weather data, and satellite positions (§3.1). 2 preprocesses and validates data compatibility (§3.2). 3 enriches features with Weather Index and satellite density (§3.3). 4 applies location-aware anomaly filtering (§3.4). 5 trains the ensemble prediction model (§3.5), yielding latency and throughput predictors.

3.1 1 Data Collection

We collect crowdsourced Starlink (identified by AS14593) measurements globally for the period from 1 January 2025 to 30 November 2025 (11 months). We also integrate high-resolution weather data and satellite position information to capture environmental and infrastructural factors affecting connectivity.

3.1.1 Global Network Performance Measurements. We collect crowdsourced performance measurements from M-Lab's NDT7 [27] and Cloudflare AIM [51]. Both datasets provide global coverage and are publicly accessible via BigQuery, making them suitable for large-scale analysis.

M-Lab. M-Lab [27] provides end-to-end throughput and latency speed tests from client devices to a globally distributed platform of measurement servers (500+ servers across 60+ metropolitan areas). The test utilizes the Network Diagnostic Tool (NDT7), which assesses application-level upload and download performance using WebSockets over TLS. Specifically, the test establishes a secure WebSocket connection over a single TCP connection configured with the BBR congestion control algorithm. During a fixed 10-second interval, the client uploads or downloads as much data as possible, thereby measuring the achievable goodput under current network conditions. The test is user-initiated and typically connects to the geographically closest available server, which is either operated by M-Lab or hosted on Google Cloud infrastructure [26]. Each test includes multiple snapshots, capturing performance metrics at different moments throughout the session, reported by both the client and the server. We analyze ≈ 15.6 M NDT7 speedtests from Starlink subscribers globally for our measurement period.

Cloudflare AIM. Cloudflare Aggregated Internet Measurements (AIM) provides end-users with a comprehensive understanding of their Internet performance through AIM scores, which evaluate connection quality for various tasks such as gaming, streaming, and video conferencing. Each category receives a label based on the results of Cloudflare Radar's speed test: "Great", "Good", "Average", "Poor", "Bad", or "Unknown". The speed test evaluates several key metrics, including download and upload throughput, loaded and unloaded latency and jitter, and packet loss. Throughput tests are conducted over HTTPS using HTTP/1.1 over a single TCP BBR connection, with progressively larger files downloaded or uploaded to measure achievable throughput. Latency and jitter are assessed using HTTP-based methods, while packet loss is measured via UDP using WebRTC to simulate real-time communication such as video calling. The test is automatically initiated when the user visits the test webpage, and the server chosen is typically geographically closest to the user within Cloudflare's CDN network. Note that AIM complements M-Lab NDT7 measurements as together they encompass Starlink performance to global cloud and CDN infrastructure, which

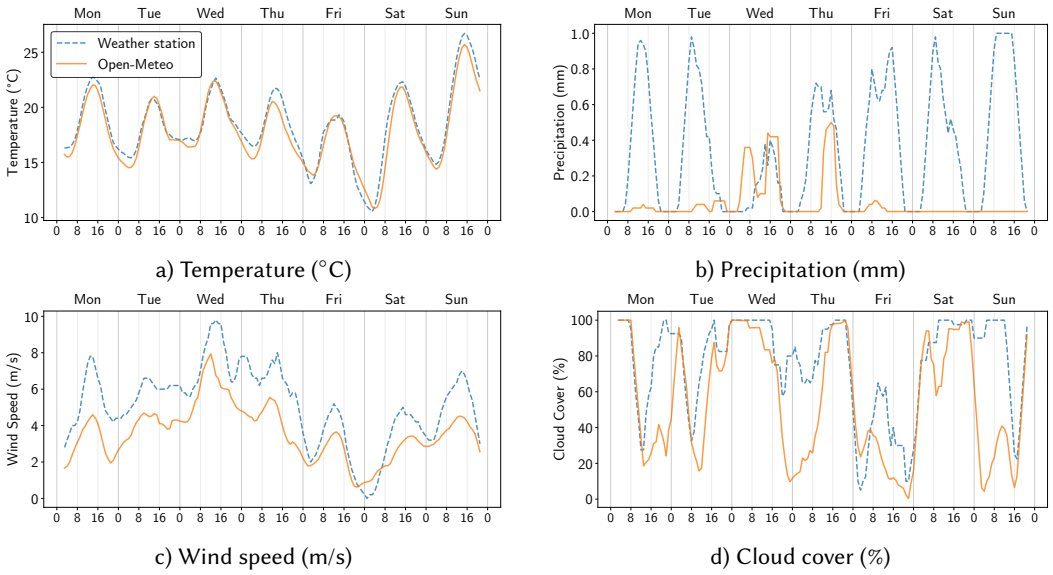


Fig. 5. Validation of OpenMeteo against Rotterdam (NL) weather station observations for 1^{st} September (Monday) to 7^{th} September (Sunday) 2025 (x-axis shows time in UTC marked at 00:00, 08:00, and 16:00). Strong temporal agreement is observed across three variables. Precipitation is less accurately captured, although errors remain marginal. See Appendix B for validation across 20 stations.

covers majority of Internet traffic behavior [8]. We analyze ≈ 157 K AIM measurements from Starlink subscribers globally for our measurement period.

As AIM and NDT7 expose different field structures and server-location representations, we normalize them into a common schema (see Appendix A) before downstream ❷ processing (§3.2).

3.1.2 Weather Data. To account for environmental factors that influence satellite connectivity, we collect hourly weather data for each client location using the OpenMeteo API [59]. We incorporate the following meteorological features: (1) temperature at 2 meters (m) above ground level (°C); (2) precipitation (mm); (3) wind speed measured at 10 m above ground (m/s); and (4) cloud cover (% of sky coverage). Recent research has shown that cloud cover and precipitation cause Ku/Ka band signal attenuation and degrade Starlink performance [21]. Similarly, strong wind can affect the movement of rain particles, which can exacerbate rain fading, and has been incorporated into models of LEO-Earth satellite connectivity [29]. Wind can also induce vibrations or misalignment in ground antennas, a phenomenon documented for large antennas in the Deep Space Network [39]. Temperature does not directly impact satellite connectivity, but it is correlated with broader meteorological factors that can affect performance and provides a proxy for diurnal cycles that may coincide with demand-driven performance variation. We query the weather data using the client lat./long. (latitude and longitude) coordinates. For each measurement timestamp, we assign weather values by linearly interpolating the two nearest hourly observations using the minute offset as the interpolation weight. We select the OpenMeteo API because it provides free hourly weather data for any location on Earth with a spatial resolution of ≈ 9 km. To validate the reliability of OpenMeteo, we compare it against ground-truth observations from national meteorological institutes. Figure 5 compares temperature, precipitation, wind speed, and cloud cover data from OpenMeteo against a representative Dutch weather station for the period from 1 to 7 September 2025 (UTC) (see Appendix B for detailed validation across 20 weather stations). We find strong temporal agreement between the two sources and observe that OpenMeteo effectively captures

weather dynamics relevant to our analysis. Temperature (fig. 5a) closely tracks the ground-truth, also clearly exhibiting the expected diurnal cycle. Although OpenMeteo and the reference station exhibit some differences in precipitation (fig. 5b), the maximum deviations remain below 1 mm/h, corresponding to light rain. Wind speed (fig. 5c) follows consistent trends across the two sources, including rapid changes. Cloud cover (fig. 5d) shows a match in relative variation despite small offsets in absolute values, which are expected given discretization and differing estimation methods in station observations.

3.1.3 Satellite Positions. We obtain satellite position data in the form of Two-Line Element (TLE) sets for all operational Starlink satellites from CelesTrak [18]. TLEs provide a standardized representation of satellite orbital elements and are regularly updated by CelesTrak. To capture changes in the constellation over time, we collect TLE data daily throughout the measurement period. We describe the use of TLE data for computing satellite density in Phase 3 of Horizon pipeline (§3.3).

3.2 ❷ Pre-processing

We apply dataset-specific processing steps to clean and validate the collected measurements.

3.2.1 Unified Network Measurements Processing. The NDT7 dataset records multiple metrics throughout the duration of each test. For consistency and completeness, we extract only the final measurement which serves as a summary of the test and includes all relevant intermediate values. To ensure data quality, we exclude incomplete or invalid entries: measurements where essential location metadata is missing (i.e., the client or server country code is null or empty), where both download and upload measurements are null, or where throughput or latency is recorded as zero.

In the Cloudflare AIM dataset, packet loss rate and jitter are reported as single values per test, whereas latency and throughput are measured multiple times during each test. Since these values are collected from the same client within a short time interval, they are typically similar. To avoid over-representing individual tests in the final analysis, we aggregate these multiple measurements into a single representative value using the median, which is resilient to outliers and better captures the central tendency in noisy measurements. The procedure for validating this aggregation choice is detailed in §3.2.3. As with NDT7, we apply data quality filters: discarding entries with missing or empty client or server location, and entries where latency or throughput is recorded as zero. Appendix A details the unified schema combining NDT7 and AIM network measurements.

3.2.2 Server Filtering. When a user performs an Internet speed test, the server is selected based on factors such as load balancing, latency, geographic proximity, server health, and ISP routing policies. Under congestion or suboptimal routing, clients may be connected to distant servers, inflating latency and introducing network location bias. Since latency is highly sensitive to propagation distance, such measurements do not reflect the user's typical experience. To mitigate this routing variability, we apply additional filtering based on client and server geolocation.

For each client location (city, country), we identify the set of "best" servers on a per-month basis. A server is classified as best if it produced at least one upload or download latency below the 1st percentile of the latency distribution for the corresponding access type during that month. This yields monthly best-server sets that adapt to infrastructure changes and evolving routing patterns. A measurement is retained only if its server belongs to the monthly best-server set for the client's location and access type. This filtering removes measurements affected by routing anomalies, congestion-driven detours, or distance-induced latency inflation, while preserving those reflecting realistic client-server proximity. The filtering removed 9.15% of measurements from the Cloudflare AIM dataset and 5.82% from NDT7. For Cloudflare AIM, the filter was generally moderate across countries, with no nation experiencing an excessive reduction in data. The procedure effectively removed server-client pairs with unusually large geographic distances; for example, in Brazil and

Botswana, excluded servers were on average more than 7,000 km from clients, reducing the average client-server distance by over 300 km in Brazil and 80 km in Botswana. In the United States, the average distance decreased from 103 km to 54 km. In the NDT7 dataset, filtering was minimal for most countries (less than 3-4% of records removed), but substantially higher in African nations with few measurements and high variability in server assignment, such as Sudan, Chad, and Benin, where over 70% of records were excluded. The procedure removed servers located more than 3,000 km from clients in Brazil, Australia, and Canada, and decreased the average client-server distance in the United States from 187 km to 71 km. Overall, the filtering effectively eliminates outlier servers while preserving most of the measurements.

3.2.3 Distribution Comparison. To evaluate the feasibility of merging heterogeneous telemetry datasets, we develop a distribution comparison procedure between NDT7 and AIM. This approach assesses how similar their performance measurements are when appropriately preprocessed, particularly with respect to different aggregation techniques applied to Cloudflare AIM data.

Our analysis focuses on September 2025,[3] which provides a sufficiently large volume of measurements while capturing typical variability in user behavior and network conditions. In particular, we compare the distributions of download and upload latency and throughput from both datasets before applying server-based filtering. We consider three aggregation techniques for summarizing the measurements: mean, median, and the 90th percentile. Each method has distinct statistical properties: the mean captures average performance but is sensitive to outliers; the median provides a more robust central tendency measure; and the 90th percentile offers insight into near-worst-case performance. To compare distributions, we select a subset of countries across continents with high measurement volumes over both datasets (USA, Brazil, UK and Japan) and compute the Jensen-Shannon Divergence (JSD), a symmetric and smoothed measure of similarity between probability distributions [33]. The JSD is defined as:

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2}\,\mathrm{KL}(P \parallel M) + \frac{1}{2}\,\mathrm{KL}(Q \parallel M), \tag{1}$$

where $M = \frac{1}{2}(P + Q)$ is the point-wise average of P and Q, and $\mathrm{KL}(\cdot \parallel \cdot)$ is the Kullback-Leibler divergence:

$$\mathrm{KL}(X \parallel Y) = \sum_i X(i) \log_2 \frac{X(i)}{Y(i)}. \tag{2}$$

The JSD is bounded between 0 and 1, where 0 corresponds to identical distributions and 1 indicates maximal divergence. We use the implementation provided by SciPy [43] and consider JSD values below 0.2 as indicative of sufficient similarity to justify merging distributions.

Table 1 reports the JSD between NDT7 and Cloudflare AIM distributions for download and upload latency and throughput across countries with high measurement volumes. Among the three aggregation methods, median aggregation achieves the lowest JSD in four out of 16 metric-country combinations and performs within 0.005 of the best method in four additional cases. The 90^{th} percentile aggregation yields the lowest JSD in eight cases but exceeds the 0.2 divergence threshold in three instances, indicating poor agreement for near-worst-case performance estimates. Mean aggregation exhibits the highest variability, with JSD values exceeding 0.34 for download throughput in some countries, reflecting its sensitivity to outliers in skewed distributions. Based on these results, *we adopt median aggregation for Cloudflare* AIM *data, as it consistently remains below the divergence threshold while achieving near-optimal alignment with* NDT7 *across all metrics.*

[3]Note that this period is only for comparing NDT7 and AIM distributions. Our overall analysis is from Jan - Nov 2025.

Country	Metric	Mean	Median	P90
USA	Download Latency	0.0163	0.0426	0.0122
	Download Throughput	0.3475	0.1609	0.0646
	Upload Latency	0.0267	0.0232	0.0330
	Upload Throughput	0.0586	0.0358	0.1218
Brazil	Download Latency	0.0726	0.1000	0.0167
	Download Throughput	0.3716	0.1469	0.2011
	Upload Latency	0.0690	0.0768	0.0408
	Upload Throughput	0.1439	0.1462	0.2157
UK	Download Latency	0.0113	0.0258	0.0252
	Download Throughput	0.4088	0.1794	0.0335
	Upload Latency	0.0123	0.0128	0.0198
	Upload Throughput	0.0214	0.0802	0.0641
Japan	Download Latency	0.0977	0.0932	0.0729
	Download Throughput	0.3897	0.1879	0.1605
	Upload Latency	0.0299	0.0324	0.0275
	Upload Throughput	0.1768	0.1351	0.2158

Table 1. Jensen-Shannon Divergence (JSD) between NDT7 and Cloudflare datasets across selected countries after server-based filtering, for September 2025. Blue cells indicate the lowest JSD per metric (best agreement), while red cells indicate high divergence (JSD ≥ 0.2).

3.3 ❸ Feature Enrichment

This phase enriches the preprocessed measurements with spatio-temporal features, a composite Weather Index derived from meteorological variables, and satellite density estimates computed via orbital propagation.

3.3.1 Spatio-Temporal Features. Each measurement record contains a UTC timestamp, client city/country, and server city/country. We represent client locations using lat./long. coordinates obtained from the `Simplemaps` dataset [44], which provides geographic information for over 40,000 cities worldwide. For cities absent from this dataset, we query the `OpenStreetMap` Nominatim API [34] to retrieve coordinates. We note that client city attribution may be inaccurate due to Starlink's GeoIP misalignment behavior, which can associate terminals with incorrect geographic locations [36]. This will lead to enriching the measurement with incorrect features. However, if the latency or throughput highly deviate from the hourly distribution of that location, they will likely be detected and removed by the anomaly filtration methods (§3.4). Zhao et al. [58] identify which cities exhibit reliable versus unreliable geolocation for Starlink users. Using these coordinates, we compute the geodesic distance between client and server locations with the `GeoPy` Python library [9], which accounts for Earth's curvature and returns distances in kms. This client-server distance serves as a proxy for the propagation delay incurred by the traffic. Note that we do not consider the Starlink PoP assignments from different regions as they are susceptible to change over time as Starlink expands its ground infrastructure globally [6]. However, past research has shown that the majority of Starlink traffic is routed through nearby PoP [6], and majority of latency and throughput performance is dominated by the satellite link [5, 30]. As such, our approximated client-server distance captures the main propagation delay component for end-to-end performance, however, future work could explore if incorporating PoP assignments can improve the accuracy of `Horizon` (see §5). We extract temporal features from the UTC timestamp to capture diurnal and weekly usage patterns. Each timestamp is decomposed into two components: (1) day of week,

encoded as an integer from 0 (Monday) to 6 (Sunday), and (2) hour with fractional minutes, computed as hour + $\frac{\text{minute}}{60}$ to produce a continuous value between 0 and 24.

3.3.2 Weather Index. To reduce feature dimensionality while preserving predictive information, we construct a composite *Weather Index* (WI) from four meteorological variables: (1) temperature, (2) precipitation, (3) wind speed, and (4) cloud cover. Due to high collinearity among these variables, rather than including all weather features independently (which would increase model complexity and risk overfitting), we project them onto a single latent dimension that maximally covaries with network performance. We use Partial Least Squares (PLS) regression to learn separate WI for latency and throughput prediction. PLS identifies linear combinations of input features that explain maximal covariance with the target variable, making it well-suited for constructing composite indices from correlated predictors. Each feature is first standardized to zero mean and unit variance to ensure equal contribution regardless of original scale. The PLS model then learns feature weights that combine these standardized values into a single WI. Cloud cover dominates both indices, contributing 46.2% for latency and 61.9% for throughput, consistent with its known impact on Ku/Ka-band signal attenuation. Precipitation ranks second, with higher weight for throughput (28.9%) than latency (23.2%), reflecting the asymmetric sensitivity of bandwidth to rain fade. Wind speed contributes substantially to the latency index (22.9%) but minimally to throughput (8.7%), potentially capturing dish pointing instability or atmospheric turbulence effects. Temperature has a negligible influence on both metrics, suggesting that thermal conditions within our measurement range do not significantly affect Starlink performance.

3.3.3 Satellite Density. We estimate satellite availability for each measurement location by computing the number of Starlink satellites within geometric visibility range at the measurement timestamp. This satellite density feature captures the instantaneous constellation configuration overhead, which influences handover frequency, routing options, and congestion levels. Our geometric model follows orbital parameters established for Starlink's LEO constellation [7]. Assuming an orbital altitude of 550–580 km and a minimum elevation angle of 40°, the effective ground coverage radius is ≈ 580 km. For each measurement, we count satellites whose sub-satellite points fall within this radius of the client location, approximating the number of simultaneously reachable satellites. The computation proceeds in four steps:

(1) *Timestamp alignment:* For each measurement, we select the `TLE` file corresponding to the measurement's UTC date to ensure accurate orbital elements.
(2) *Orbit propagation:* Satellite positions are propagated to the exact measurement timestamp using the `SGP4` algorithm [41, 54].
(3) *Geodetic transformation:* Propagated positions in the True Equator Mean Equinox (TEME) reference frame are converted to geodetic coordinates (lat./long.) using the `Skyfield` library [40], which provides high-precision transformations based on ITRF/ITRS standards [31, 54].
(4) *Spatial counting:* For each measurement, we compute the geodesic distance between the client location and every satellite's sub-satellite point, counting those within the 580 km coverage radius.

We find that satellite density directly influences user connectivity and network performance. Higher satellite counts increase the probability that at least one satellite maintains a strong line-of-sight connection, enable smoother handovers as satellites traverse the sky at ≈ 7.6 km/s, and allow traffic distribution across multiple links to reduce congestion. These effects are particularly pronounced at higher latitudes, where orbital geometry concentrates satellite coverage, and during peak usage periods when load balancing across satellites becomes critical. We emphasize that our satellite density model represents an idealized geometric upper bound on satellite availability, rather than a direct measure of usable capacity. In practice, the effective number of reachable satellites is

constrained by factors such as terminal scheduling, network load, antenna pointing limitations, and physical obstructions including terrain, buildings, and foliage [13, 55]. To mitigate the impact of measurements affected by such factors, we employ anomaly filtering described next.

3.4 ❹ Anomaly Detection

Crowdsourced network measurements are inherently noisy due to the uncontrolled nature of client devices, network conditions, and measurement timing. Even after mitigating server-selection biases (§3.2) and incorporating client-server distance to account for terrestrial propagation delay, anomalous observations may persist. Such anomalies can arise from transient link failures, short-lived congestion events, routing instabilities, satellite handover disruptions, or hardware issues at either endpoint. These factors are not directly observable through the available features, yet their presence can substantially degrade model performance if left unaddressed.

To construct a robust training dataset while respecting the pronounced geographic heterogeneity of Starlink performance, we evaluate three location-aware outlier filtering strategies. All filtering methods are applied independently for each unique lat./long. pair, acknowledging that performance distributions vary substantially across regions. For example, mature deployments with dense ground infrastructure such as in North America, exhibit different baseline performance than regions in central Africa with sparse deployment (see fig. 3). Since our predictive models target download latency and download throughput separately, each filtering strategy is applied independently to these two metrics, yielding distinct filtered datasets for each prediction task.

The two statistical methods (percentile-based and MAD) employ an adaptive temporal windowing strategy to ensure sufficient sample sizes for reliable outlier detection. Measurements are initially grouped into one-hour windows centered at each timestamp. If less than eight measurements fall within this window, the window is symmetrically extended in 30-minute increments up to a maximum span of three hours. If the extended window still contains fewer than eight measurements, all observations in that window are marked as outliers due to insufficient statistical support.[4] This adaptive approach balances temporal locality, which captures time-varying performance characteristics, with statistical reliability, which requires adequate sample sizes for robust outlier detection.

3.4.1 Percentile-Based Filtering. Percentile-based filtering retains a fixed fraction of measurements within each adaptive window, removing extreme values that likely represent anomalous conditions. We introduce a parameter k (the keep ratio) that determines the fraction of measurements retained as normal observations. For latency, where lower values indicate better performance, we retain the lowest k fraction of measurements within each window. For throughput, where higher values indicate better performance, we retain the highest k fraction. Measurements exceeding these thresholds are flagged as outliers. As mentioned earlier, in windows with fewer than eight observations, all observations are marked as outliers. A measurement may appear in multiple overlapping windows, since windows are centered at each timestamp. To avoid removing measurements flagged as outliers in only a small minority of windows, we apply a voting mechanism. A measurement is removed from the dataset only if it is flagged as an outlier in at least half of the windows in which it appears. This approach follows established practices in crowdsourced network measurement analysis [15], effectively suppressing extreme outliers while preserving natural performance variability. We evaluate $k \in \{0.70, 0.75, 0.80\}$ to explore different filtering aggressiveness levels.

3.4.2 Directional MAD Filtering. The median absolute deviation provides a robust measure of statistical dispersion that is less sensitive to outliers than the standard deviation. We adopt a directional MAD strategy that identifies abnormally high latency and low throughput values,

[4]We experimented with several filtering approaches and found this method to produce most statistically significant outcome.

targeting the performance degradation direction relevant to each metric. The MAD is defined as:

$$\text{MAD} = \text{median}\big(\,|x_i - \text{median}(x)|\,\big),$$

and is computed independently for each metric within each adaptive window.

Outliers are flagged using directional criteria that differ by metric:

(1) **Latency:** $x \geq \text{median}(x) + k \cdot \text{MAD}$ *(unusually high latency).*
(2) **Throughput:** $x \leq \text{median}(x) - k \cdot \text{MAD}$ *(unusually low throughput).*

Similar to percentile-based filtering, we mark measurements in windows with fewer than eight observations as outliers, and a measurement is removed only if flagged in at least half of its containing windows. The parameter k controls filter strictness, with larger values permitting greater deviation from the median before flagging. Prior work has employed values around $k = 2.5$ for anomaly detection in network measurements [57]. We evaluate $k \in \{2, 2.5, 3\}$, corresponding to increasingly permissive thresholds.

3.4.3 Isolation Forest Filtering. Isolation Forest (IF) provides a model-based alternative to statistical filtering that does not assume an underlying data distribution. The algorithm exploits the fact that anomalous points, being rare and different, are easier to isolate through random recursive partitioning. Points requiring fewer splits to isolate exhibit shorter average path lengths across an ensemble of isolation trees and are more likely to be anomalous.

For each lat./long. pair, we train a separate IF model using the target metric (latency or throughput) augmented with hour-of-day and day-of-week features to capture temporal patterns. Including temporal features allows the model to distinguish between measurements that are anomalous given the time of day versus those that reflect normal diurnal variation. The `contamination` parameter specifies the expected fraction of anomalies; to maintain consistency with percentile-based filtering, we set *contamination* $= 1 - k$ with $k \in \{0.70, 0.75, 0.80\}$. Unlike the window-based methods, IF operates on the full set of measurements for each location rather than within temporal windows. For consistency, we exclude locations with fewer than eight total measurements. This criterion is less restrictive than the window-based approaches, which require sufficient observations within localized time windows. Consequently, IF-based filtering retains more locations overall, particularly in sparsely measured regions where temporal windowing would discard most data.

3.5 ❺ Model Training

The final phase trains prediction models for download latency and throughput using the filtered and feature-enriched dataset. We prioritize interpretable models for two reasons. *First*, they enable quantification of feature contributions, allowing us to identify which factors most strongly influence Starlink performance. *Second*, they facilitate evaluation of whether the model captures meaningful patterns rather than spurious correlations. Prior research has consistently demonstrated that decision-tree-based models outperform neural networks on tabular datasets [14], making them well-suited for the structured, heterogeneous data in our pipeline.

We employ an ensemble model combining two complementary tree-based regressors implemented in `scikit-learn` [38]. The *Random Forest (RF)* comprises 100 decorrelated decision trees trained on bootstrap samples with random feature subsets, reducing variance and mitigating overfitting through averaging. The *Gradient Boosting Regressor (GBR)* sequentially constructs 100 shallow trees, each correcting the residual errors of its predecessors, with a learning rate of 0.1 to control the contribution of each tree. This combination leverages the variance reduction of bagging (RF) with the bias reduction of boosting (GBR), yielding robust predictions across diverse conditions.

Prior to training, all input features are standardized using a `RobustScaler`, which centers data using the median and scales using the interquartile range. Unlike standard scaling based on mean and variance, `RobustScaler` is less sensitive to outliers, an important property given the noise

Filtration Info		Latency (ms)			Throughput (Mbps)		
Name	Param	% Kept	MAE	RMSE	% Kept	MAE	RMSE
Percentile	0.7	68.16	21.90	30.07	68.01	53.98	68.06
	0.75	73.22	25.05	35.73	73.13	55.00	69.56
	0.8	77.83	28.34	39.28	77.77	55.77	70.71
Directional MAD	2	77.65	27.96	38.96	96.80	57.57	73.75
	2.5	80.97	30.71	42.56	97.20	57.53	73.71
	3	83.62	33.26	46.28	97.34	57.63	73.84
Isolation Forest	0.7	69.99	32.28	43.86	69.99	38.76	52.25
	0.75	74.99	34.31	46.82	74.99	40.39	53.95
	0.8	79.99	36.56	50.56	79.99	42.38	56.12

Table 2. For each filtration technique and parameter, the table shows first the percentage of records retained after filtering ("% Kept") for November 2025 data, followed by the MAE and RMSE of the download latency and throughput models trained on the filtered data.

characteristics of crowdsourced measurements [22]. The scaler is fit exclusively on the training split and subsequently applied to the test data to prevent data leakage.

We evaluate model performance using three complementary metrics. (1) *Mean Absolute Error (MAE)* measures the average magnitude of prediction errors, providing an interpretable measure of typical error in the same units as the target variable (milliseconds for latency, Mbps for throughput). (2) *Root Mean Squared Error (RMSE)* computes the square root of the average squared errors, giving higher weight to larger deviations and thereby penalizing models that produce occasional large mistakes. (3) The *Coefficient of Determination* (R^2) quantifies the proportion of variance in the target variable explained by the model, with values of 1 indicating perfect prediction and 0 indicating no explanatory power beyond predicting the mean. Together, MAE and RMSE quantify absolute prediction error, with RMSE being more sensitive to outliers, while R^2 provides a normalized measure of model fit that facilitates comparison across datasets with different variance characteristics.

3.5.1 Optimizing Ensemble Weights. We partition the data using an 80-20 train-test split, stratified to preserve the geographic distribution of measurements across both sets. The final prediction is computed as a weighted average of the RF and GBR outputs: $\hat{y} = w_{\text{RF}} \cdot \hat{y}_{\text{RF}} + w_{\text{GBR}} \cdot \hat{y}_{\text{GBR}}$, where $w_{\text{RF}} + w_{\text{GBR}} = 1$. We optimize the ensemble weights via grid search over the test set, evaluating weights in increments of 0.05 from 0 to 1. RMSE serves as the optimization objective, chosen for consistency with the squared-error loss functions inherent to both constituent models. This weight optimization allows the ensemble to adaptively balance the contributions of each model based on their relative performance for each prediction task.

3.5.2 Selecting the Anomaly Filtering Strategy. Table 2 reports the fraction of measurements retained by each filtering method, as well as the corresponding model performance on the test set. The results reveal distinct optimal strategies for latency and throughput prediction.

For *latency*, percentile-based filtering with $k = 0.7$ achieves the lowest MAE (21.90 ms) and RMSE (30.07 ms), outperforming both MAD and Isolation Forest approaches. This suggests that latency outliers are effectively captured by simple distributional cutoffs, and that aggressive filtering improves prediction by removing transient spikes caused by handovers or congestion. For *throughput*, Isolation Forest with $k = 0.7$ yields substantially better performance (MAE 38.76 Mbps, RMSE 52.25 Mbps) than percentile or MAD filtering. The model-based approach likely captures more complex anomaly patterns in throughput measurements, which exhibit higher variability and may not follow simple distributional assumptions.

Comparing $k = 0.7$ and $k = 0.75$ configurations, the more aggressive filtering ($k = 0.7$) provides marginal accuracy improvements but retains $\approx 5\%$ fewer measurements. This trade-off introduces overfitting risk, particularly for locations with sparse data. To balance prediction accuracy against

Months	Latency (ms)					Throughput (Mbps)				
	RF Wt.	GBR Wt.	MAE	RMSE	R^2	RF Wt.	GBR Wt.	MAE	RMSE	R^2
1	0.40	0.60	25.06	35.73	0.403	0.35	0.65	40.39	53.95	0.304
2	0.50	0.50	24.12	34.46	0.432	0.35	0.65	42.13	55.98	0.294
3	0.55	0.45	24.39	34.80	0.438	0.35	0.65	42.19	55.87	0.287
4	0.55	0.45	24.55	35.13	0.440	0.40	0.60	41.90	55.35	0.275
5	0.50	0.50	24.52	34.68	0.434	0.40	0.60	41.76	54.97	0.268
6	0.60	0.40	24.79	35.15	0.438	0.40	0.60	41.26	54.32	0.266
7	0.50	0.50	24.97	35.29	0.434	0.40	0.60	41.11	54.11	0.262
8	0.50	0.50	25.26	35.69	0.432	0.40	0.60	40.83	53.77	0.262
9	0.55	0.45	25.96	36.90	0.461	0.40	0.60	40.10	52.85	0.264
10	0.55	0.45	26.51	37.80	0.478	0.40	0.60	39.50	52.14	0.268
11	0.60	0.40	27.22	39.37	0.486	0.40	0.60	39.13	51.71	0.269

Table 3. Latency and throughput metrics for models trained with different numbers of months of data, ending in November 2025. The best latency and throughput results are highlighted in blue, together with the corresponding training window lengths.

data retention, we select percentile filtering with $k = 0.75$ for latency and Isolation Forest with $k = 0.75$ for throughput in subsequent experiments.

3.5.3 Determining Optimal Training Window. Table 3 evaluates model performance as a function of training data duration, with all periods ending in November 2025. The results reveal contrasting optimal windows for latency and throughput prediction.

For *latency*, the model achieves optimal performance when trained on a two-month window (MAE 24.12 ms, RMSE 34.46 ms). Including additional historical data leads to a slight degradation in accuracy, highlighting the strongly non-stationary nature of latency dynamics. Frequent changes in routing policies, constellation topology, and ground infrastructure make older measurements stale and less representative of current conditions, introducing outdated patterns that ultimately degrade predictive performance. For *throughput*, performance improves monotonically with additional training data, reaching optimal values with 11 months (MAE 39.13 Mbps, RMSE 51.71 Mbps). This indicates that throughput prediction benefits from exposure to diverse conditions accumulated over longer periods, including seasonal weather variations and geographic expansion of the measurement base. The higher inherent variability of throughput measurements may also require larger sample sizes to learn robust patterns.

Based on these findings, we configure the final model of `Horizon` with two months of training data for latency prediction and 11 months for throughput prediction. This differentiated approach allows `Horizon` to leverage the temporal characteristics most relevant to its target metric and achieve more accurate performance predictions (see §4.2 for comparison against state-of-the-art).

4 Evaluation

This section evaluates `Horizon`'s predictive performance for Starlink latency and throughput on a global scale, highlighting the integration of weather and satellite density features.

4.1 Methodology

We evaluate `Horizon` using MAE, RMSE, and R^2 (§3.5) under two complementary settings. The first is an 80/20 random split for baseline comparison, where all models are trained on data up to 23 November 2025 and evaluated on a common held-out test set, enabling direct comparison under identical test conditions (table 4). The second is a one-week temporal holdout (24–30 November 2025), which assesses deployment realism (§4.6) by predicting an entirely unseen future period (fig. 6). For the temporal holdout, ensemble weights are tuned on the 80/20 split and then fixed while retraining on all data up to 23 November 2025. Training and evaluation in both settings use

Model	Latency (ms)			Throughput (Mbps)		
	MAE	RMSE	R^2	MAE	RMSE	R^2
Horizon	**24.19**	**34.68**	**0.43**	**39.00**	**51.52**	**0.27**
Random Forest	25.17	36.09	0.38	39.48	53.94	0.20
GBR	24.69	36.25	0.38	39.99	52.51	0.24
XGBoost (T3P)	24.36	35.00	0.41	39.54	52.01	0.26
LSTM (T3P)	26.54	39.05	0.28	41.75	54.27	0.19
GRU Seq2Seq (StarNet)	25.12	37.81	0.32	41.81	54.23	0.19
DT	24.45	35.90	0.39	39.66	52.26	0.25
KNN	25.46	36.84	0.36	40.99	55.78	0.14
LR	29.41	43.31	0.11	44.33	56.95	0.11

Table 4. Comparison of Horizon against baseline models on the held-out test set from the 80-20 split. Horizon outperforms the state-of-the-art for both latency and throughput prediction (highlighted).

the same measurement pipeline and anomaly filtering (§3.4), a 75th-percentile filter for latency and an Isolation Forest (0.75 contamination) for throughput, ensuring that predictions reflect typical performance behavior. To assess the statistical significance of our predictions over the temporal holdout period, we construct 95% confidence intervals (CIs) for the reported error metrics. Specifically, for each city and day, we employ a non-parametric bootstrap procedure: we resample the hourly prediction errors with replacement 10^6 times. For each bootstrap sample, we compute the corresponding error metric, and then extract the 2.5^{th} and 97.5^{th} percentiles to define the lower and upper bounds of the CI. This procedure captures the empirical variability of the daily prediction errors and provides a robust measure of uncertainty.

State-of-the-art. To contextualize Horizon's performance, we compare against both standard regression models and architectures from recent LEO satellite prediction literature. Standard baselines include Linear Regression (LR), K-Nearest Neighbors (KNN, $k = 5$), and Decision Tree (DT, max depth 10), representing linear, instance-based, and non-linear approaches respectively. We also implement baselines inspired by closest state-of-the-art T3P [50] and StarNet [23], which employ deep learning on terminal-level telemetry. Since direct comparison is infeasible due to data source differences (T3P and StarNet use proprietary 1-second terminal telemetry, while Horizon uses crowdsourced data), we adapt their architectures to operate on Horizon's feature set: (1) LSTM following T3P's two-layer architecture (64 hidden units, 20% dropout); (2) XGBoost configured similarly to T3P (100 estimators, learning rate 0.1, max depth 5); and (3) GRU Seq2Seq based on StarNet's attention mechanism. All baselines use identical features (lat./long., client-server distance, hour-of-day, day-of-week, satellite density, WI), training periods, and evaluation protocols as Horizon.

Ground Truth Construction. On the holdout period we compute hourly predictions and compare them against the ground truth, which is calculated as the median of filtered measurements from the same city within a flexible temporal window (±0.5 to ±1.5 hours as in §3.4). Predictions without matches are excluded. Figure 6 reports the error metrics by country. We aggregate measurements to construct ground truth rather than comparing against individual observation points because crowdsourced data are sparse in some areas; aggregation substantially increases coverage and yields an hourly reference value for most locations. Our dataset [3] and code [4] are publicly available to facilitate reproducibility and future research.

4.2 Comparative Analysis

We compare Horizon against the state-of-the-art baselines outlined in §4.1. Table 4 presents the results for the same test set across models. Horizon outperforms all models across all metrics

New York		Santiago		Berlin	
Latency	Throughput	Latency	Throughput	Latency	Throughput
11.32 (7.7-15.3)	49.26 (35.2-61.7)	18.53 (13.5-23.4)	25.48 (16.9-33.1)	10.12 (7.7-12.3)	13.59 (8.3-18.4)
11.15 (9.2-12.9)	39.12 (29.7-47.5)	21.10 (17.6-24.3)	25.30 (19.1-30.6)	9.33 (7.8-10.8)	19.46 (14.4-24.0)
8.93 (6.9-10.7)	42.51 (27.3-56.6)	22.80 (16.2-29.2)	22.77 (16.6-28.4)	12.03 (8.4-16.4)	16.11 (11.1-20.8)
10.22 (8.6-11.7)	62.71 (43.4-80.0)	16.86 (13.0-20.5)	31.15 (23.7-38.0)	9.61 (8.0-11.3)	18.34 (12.3-23.6)
10.14 (8.0-12.1)	71.76 (53.4-87.6)	17.58 (13.2-21.7)	33.25 (22.9-43.6)	8.57 (7.3-9.7)	11.08 (8.1-13.7)
10.19 (8.3-12.0)	51.89 (30.9-69.9)	16.83 (13.9-19.6)	36.41 (25.8-46.6)	8.97 (7.4-10.4)	14.45 (10.6-17.9)
8.92 (6.1-11.5)	50.90 (30.6-68.3)	13.72 (8.0-18.5)	32.51 (15.7-45.9)	8.75 (7.2-10.2)	12.97 (10.0-15.7)

Table 5. Daily RMSE of latency and throughput predictions for selected cities, with parentheses showing the lower and upper bound of 95% confidence interval computed from hourly predictions. Rows correspond to different days from 24 November (Monday) to 30 November (Sunday) 2025. MAE, reported in Appendix C, shows similar patterns.

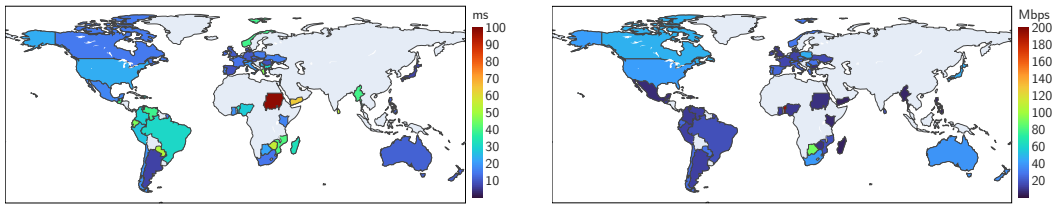
and prediction targets. For latency, `Horizon` achieves 24.19 ms MAE compared to the next-best `XGBoost` at 24.36 ms, with substantially better R^2 (0.43 vs 0.41). For throughput, `Horizon` achieves 39.00 Mbps MAE and 0.27 R^2, outperforming `XGBoost` (39.54 Mbps MAE, 0.26 R^2) and all other baselines.

The superior performance of `Horizon` over individual Random Forest and Gradient Boosting models demonstrates the value of ensemble combination. `LSTM` underperforms despite its success in `T3P`'s original setting, reinforcing findings that decision-tree-based models often outperform neural networks on tabular data [14]. `GRU Seq2Seq` underperforms both `LSTM` and tree-based models despite its sophisticated attention mechanism, suggesting that architectural complexity does not compensate for the lack of sequential temporal structure in our aggregated hourly data. `LR` performs poorly (R^2 of 0.11 for both targets), indicating that the relationship between features and Starlink performance is fundamentally non-linear. `KNN` struggles particularly with throughput (R^2 of 0.14), suggesting that simple instance-based approaches cannot capture the complex spatio-temporal patterns in the data.

4.3 Model Performance

Figure 6 visualizes RMSE geographically for 24 November 2025, the first day of the temporal holdout period. Latency predictions achieve MAE of 17.76 ms, RMSE of 26.21 ms, and R^2 of 0.574. Throughput predictions yield MAE of 25.63 Mbps, RMSE of 38.41 Mbps, and R^2 of 0.369. These results indicate that the model captures latency patterns more accurately than throughput, which exhibits higher intrinsic variability across locations and time. The strong R^2 for latency suggests good generalization to unseen country-level data, whereas the lower R^2 for throughput highlights opportunities for model refinement or inclusion of additional features.

Regional Analysis. Both models perform particularly well in North America, Europe, and Australia, where dense ground infrastructure and consistent measurement flows enable reliable predictions. Throughput predictions are also accurate in Africa, Asia, and South America, partly because throughput values in these regions are generally lower and more stable (fig. 3b). In Asia, predictions are reliable in Japan and the Philippines, where measurement density is high and Starlink infrastructure is well-established. Predictions are less accurate in Yemen and Myanmar due to sparse measurement coverage and high routing variability from limited regional PoPs. In Africa, performance is strong in countries with established Starlink infrastructure or proximity to PoPs, such as Nigeria, South Africa, and Mozambique. Conversely, Sudan exhibits the poorest latency predictions (RMSE close to 100 ms), attributable to its distance from the nearest PoPs in Nigeria and Kenya [6]. This geographic isolation induces high routing variability: after anomaly filtering, Sudan's latency retains a standard



a) Latency RMSE (ms) b) Throughput RMSE (Mbps)

Fig. 6. Country-level prediction RMSE for download latency (left) and throughput (right) on 24 November 2025.[5] MAE, reported in Appendix C, shows similar patterns. Darker colors indicate higher error. North America, Europe, and Australia exhibit lowest errors; Sudan and Yemen show highest latency prediction error.

deviation of 116 ms (reduced from 377 ms pre-filtering), with a median of 234 ms and 47.6% of records excluded. Latency predictions in northern South America show elevated error relative to other regions, though RMSE remains below 40 ms for most countries. The higher error reflects the interaction of two factors: variable routing paths to PoPs in the southern United States and Brazil, and limited measurement density in countries such as Venezuela and Colombia. These regional variations underscore the importance of measurement coverage and infrastructure proximity for prediction accuracy.

Statistical Significance. Table 5 reports the RMSE for latency and throughput predictions over the 7-day temporal holdout period (24–30 November), along with 95% confidence intervals (CIs) for New York (USA), Santiago (Chile), Berlin (Germany). These cities were selected for geographic diversity and to contrast well-connected regions (USA, Germany) with locations distant from Starlink ground infrastructure (Chile). As expected, cities with higher data volumes exhibit lower errors and narrower CIs, whereas Santiago shows larger errors and wider intervals, particularly for latency. In New York, errors for throughput are larger because there is bigger variability due to some tests having very high speeds (greater than 200 Mbps). Interestingly, weekends (29–30 November) generally have lower errors for both latency and throughput. Throughput predictions degrade more over time than latency, though not dramatically, underscoring the need for continuous model retraining to maintain prediction accuracy.

Coverage. Currently, `Horizon` generates meaningful predictions for more than 55 countries, representing ≈ 40% of countries with Starlink availability [45]. As Starlink expands and crowdsourced measurement volumes increase, we expect coverage and prediction accuracy to improve correspondingly.

4.4 Feature Importance

By extracting the feature importance from the trained models, we observe geographic position dominates both models, with latitude alone contributing ≈ 42% for latency and ≈ 46% for throughput. The results reflect the uneven distribution of Starlink infrastructure from both ground and space perspective. As seen in fig. 1, the northern hemisphere (covering EU and NA) hosts ≈ 90% of the Starlink satellite fleet and the region also more ground stations and PoPs to cater to the majority of Starlink customers. This geographic disparity leads to systematic performance differences (also evident in fig. 3) that are effectively captured by `Horizon`. Importantly, these geographic features primarily establish a stable baseline level of performance across regions.

In contrast, dynamic features are the main drivers of variability around this baseline. The Weather Index (WI) is the second most important feature for both models (affecting ≈ 14% for latency, 15.3%

[5]Only countries present in both latency and throughput datasets are included. Differences in data collection periods and filtering steps lead to slight variations in country coverage.

for throughput), confirming the significant role of meteorological conditions documented in prior work [17, 21, 25]. Satellite density contributes notably to the latency model (10.3%), consistent with its direct influence on routing options and handover quality, while for throughput the contribution is slightly lower, but still notable (6.4%). Client-server distance matters primarily for latency (8.3%) due to propagation delay, while its contribution to throughput is minimal (1.1%). Hour of day is moderately important (9.4% and 11.3%), enabling the model to capture diurnal patterns such as improved nighttime performance. Day of week has relatively minor impact (1.8% and 3.2%), suggesting weekly patterns are less pronounced than daily cycles.

4.5 Spatial Generalization

To evaluate spatial generalization, we conduct a leave-one-location-out experiment across three geographically diverse cities. We select Berlin (Germany), Santiago (Chile), and Yangon (Myanmar) as holdout locations, chosen to span different latitudes, hemispheres, and levels of Starlink infrastructure maturity. For each city, all local measurements are removed from training, and the model is evaluated on hourly predictions for 24–30 November 2025. Figure 7 presents the results.

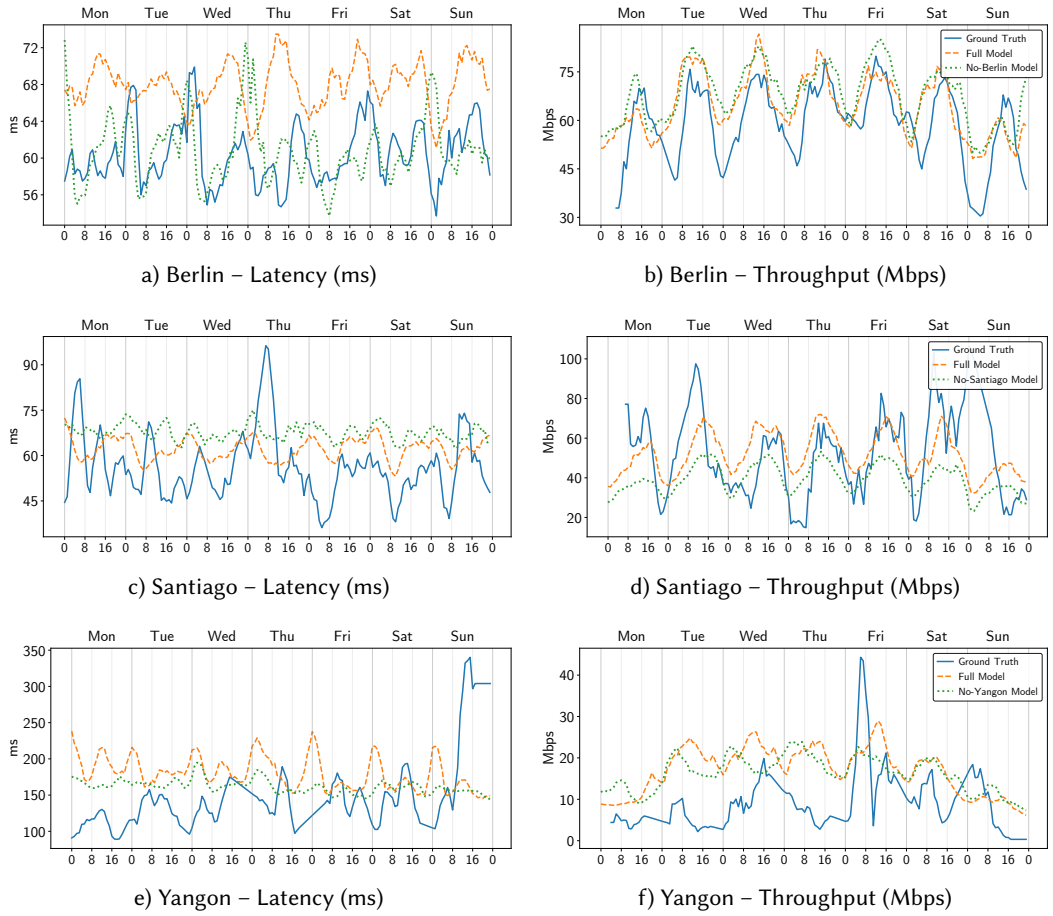
These cities represent increasing levels of difficulty for spatial generalization. Berlin lies within Starlink's densest coverage band (fig. 1), with nearby training cities Prague (281 km) and Copenhagen (356 km) providing rich spatial context. Santiago is more isolated: the nearest training location is Buenos Aires (1,137 km) across the Andes, with sparser alternatives in Montevideo (1,340 km) and Lima (2,464 km). Yangon presents the hardest case, combining limited Starlink adoption with sparse measurement coverage; the nearest training cities are Dhaka (973 km) and Kuala Lumpur (1,637 km), both with limited data.

For Berlin (DE), the model trained without local data achieves comparable latency predictions (MAE 6.68 ms vs 9.10 ms, RMSE 9.15 ms vs 10.62 ms) to the full model, with the slight improvement likely attributable to reduced variance from having fewer location-specific parameters. This result confirms that the model learns transferable spatial patterns from nearby cities such as Prague and Copenhagen rather than memorizing location-specific behavior. Throughput predictions show a modest increase in error without Berlin data (MAE 14.32 Mbps vs 13.04 Mbps, RMSE 17.48 Mbps vs 16.12 Mbps), consistent with the higher intrinsic variability of throughput measurements observed in our global evaluation (§4.3).

For Santiago (CL), removing local data leads to a moderate increase in latency error (MAE 17.97 ms vs 15.57 ms), reflecting the geographic isolation of the city with no nearby training locations. Throughput predictions remain nearly identical with and without Santiago data (MAE 27.61 Mbps vs 27.55 Mbps), indicating that throughput patterns in this region are already well captured by measurements from other locations. The higher absolute throughput errors compared to Berlin are consistent with the greater variability observed in Southern Hemisphere measurements, where ground station infrastructure is less dense.

For Yangon (MM), the no-city model produces lower latency error than the full model (MAE 51.21 ms vs 65.74 ms), a pattern also observed for Berlin. This counterintuitive result likely reflects the fact that Yangon's high absolute latency (ground-truth mean of 146 ms, well above the 55–60 ms typical of Berlin and Santiago) represents an outlier distribution that the full model slightly overfits to at the expense of generalized patterns. Throughput predictions are comparable between the two models (MAE 12.29 Mbps vs 13.18 Mbps), and the lower absolute throughput in the region (ground-truth mean of 9.8 Mbps) reflects Yangon's limited Starlink infrastructure and lower measurement density. Despite being the most isolated holdout city with only Dhaka and Kuala Lumpur as distant training neighbors, the model still tracks the temporal structure of Yangon's performance patterns.

Across all three cities, both latency and throughput errors remain within acceptable bounds for practical use. Berlin and Santiago achieve sub-20 ms latency MAE and Yangon stays within 52–66 ms



a) Berlin – Latency (ms)

b) Berlin – Throughput (Mbps)

c) Santiago – Latency (ms)

d) Santiago – Throughput (Mbps)

e) Yangon – Latency (ms)

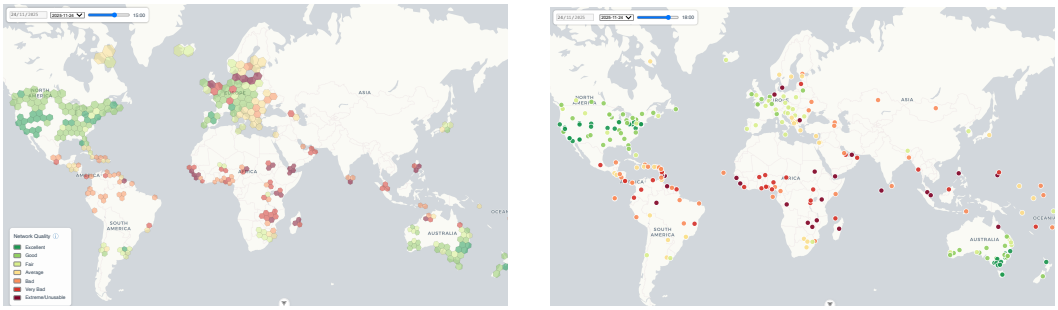
f) Yangon – Throughput (Mbps)

Fig. 7. Spatial generalization experiment: City predictions without local training data. Ground truth (blue) represents hourly median measurements; full model (orange) was trained on all locations including the specific city; no-city model (green) excludes all city data from training. The x-axis spans 24 November (Monday) to 30 November (Sunday) 2025, with time in UTC marked at 00:00, 08:00, and 16:00.

against its higher baseline. These findings demonstrate that `Horizon` captures meaningful spatial correlations that enable reliable prediction for locations without direct training data. This ability to generalize from neighboring measurements is particularly valuable for expanding coverage to newly served regions where measurement density is initially sparse. Importantly, the results across three continents confirm that this generalization extends beyond well-connected European cities to isolated locations in the Southern Hemisphere and Southeast Asia.

4.6 `Horizon` Web Deployment

We deploy `Horizon` as a continuously operating web-based system that provides global Starlink performance forecasts. The system targets two audiences: end users seeking to anticipate connectivity quality for latency-sensitive applications (e.g., video conferencing, online gaming), and researchers studying LEO network behavior at scale. Figure 8 shows the browser interface, which displays hourly predictions over a global view based on the `H3` hierarchical hexagonal index [52] at



a) Global predictions using H3 level-2 hexagons. b) City-level predictions for 270 major cities.

Fig. 8. Horizon browser interface showing Starlink performance predictions. Color scale maps network quality from excellent (green) to severely degraded (dark red); full mapping in Appendix D.2.

resolution 2 (average edge length ≈150 km). This resolution aligns with Starlink's internal service cells [45], with each hexagon centroid serving as the prediction point for its cell.

Predictions are restricted to hexagons whose centroids lie within 300 km of at least one training measurement, consistent with our spatial generalization results (§4.5). Hexagons over oceans or in regions without Starlink availability are excluded, yielding ≈300 valid cells (Appendix fig. 10). Predicted latency and throughput are mapped to color-coded QoS categories, enabling users to quickly assess whether conditions are suitable for time-sensitive activities without interpreting raw metrics. The system follows a weekly retraining cycle, leveraging its demonstrated ability to generalize to unseen 7-day windows (§4.1). Crowdsourced measurements, weather observations, and satellite orbit data (TLE) are ingested daily from `BigQuery`, `OpenMeteo`, and `CelesTrak` respectively, accumulating a full week of inputs before each retraining pass. After retraining, the system generates hourly forecasts for the following week, incorporating weather predictions to account for anticipated atmospheric effects.

5 Discussion and Conclusion

This paper presented `Horizon`, the first global-scale machine learning system for predicting LEO satellite network performance. By integrating over 11 months of crowdsourced measurements from M-Lab NDT7 and Cloudflare AIM across more than 90 countries, `Horizon` overcomes the geographic limitations of prior studies that relied on small numbers of professionally instrumented terminals in controlled environments. Our ensemble model achieves a global MAE of 17.76 ms for latency and 25.63 Mbps for throughput, demonstrating meaningful predictive capability across diverse geographic and climatic conditions. The feature importance analysis reveals that geographic position dominates prediction accuracy, with latitude alone contributing 42% for latency and 46% for throughput. This reflects the physical constraints of LEO satellite networks: latitude determines satellite visibility windows, orbital shell coverage, and proximity to ground infrastructure. Weather features contribute 14.0% for latency and 15.3% for throughput, validating prior observations about atmospheric attenuation in Ku- and Ka-band frequencies while demonstrating that weather-aware prediction offers tangible improvements over weather-agnostic baselines. The Berlin leave-one-out experiment provides further evidence that `Horizon` captures transferable spatial patterns rather than memorizing location-specific behavior. The model trained without Berlin data achieved comparable or better performance than the full model, indicating that spatial interpolation from neighboring regions can yield reliable predictions for locations lacking historical measurements. This capability is particularly valuable for expanding coverage to newly served regions where measurement density may initially be sparse.

A notable design decision is the omission of explicit infrastructure proximity features, such as distance to the nearest Starlink PoP or ground station. We made this choice for three reasons. *First*, PoP assignments are dynamic: Starlink continuously deploys new ground infrastructure, and routing policies evolve accordingly, meaning that historical PoP proximity would not reliably predict future routing [6]. *Second*, prior research demonstrates that the satellite link dominates end-to-end latency and throughput, with terrestrial backhaul contributing a smaller fraction of total delay [5, 30]. *Third*, our client-server distance feature already captures terrestrial propagation delay for the measurement endpoints. That said, server selection filtering disproportionately affects countries with sparse ground infrastructure, where routing assignments exhibit high variance. For instance, clients in Sudan and Yemen are occasionally routed to PoPs in Europe (Frankfurt, Milan) rather than geographically closer alternatives (Nairobi, Doha, Muscat), inflating latency well beyond what geographic features alone can capture. Incorporating Starlink's evolving ground topology and routing policies into the prediction pipeline is a promising direction for future work, particularly for latency, where propagation distance is the dominant factor.

Several other limitations warrant acknowledgment. For instance, intentional test spoofing, particularly in geopolitically sensitive regions, could bias measurements. While MLab and Cloudflare tests are protected against server-side attacks, extremely large-scale client-side attacks could meaningfully affect results; we consider this a low-probability risk. Moreover, the crowdsourced measurement approach introduces inherent noise and potential biases, as users who voluntarily run speed tests may not be representative of the broader Starlink customer base. Self-selection bias, where users initiate tests during periods of degraded performance, is partially mitigated through the use of two independent data sources and a robust filtering pipeline. These limitations are not unique to our work and would similarly affect any system leveraging these platforms. Second, our feature set does not include terminal-level telemetry, such as satellite assignment, dish orientation, or obstruction metrics, which could likely improve prediction accuracy if available. Third, the hourly temporal granularity and median-based aggregation smooth over short-term fluctuations caused by satellite handovers and transient weather phenomena, limiting applicability for real-time performance estimation. Finally, geographic coverage is constrained by measurement availability, with meaningful predictions currently limited to approximately 55 countries, representing roughly 40% of regions with Starlink service.

Several directions offer promising avenues for future research. Network congestion indicators derived from time-of-day patterns or regional usage estimates could capture demand-driven performance degradation. Terrain characteristics, including elevation and obstruction profiles, may explain performance variations in mountainous or heavily forested regions where line-of-sight to satellites is compromised. While this work focuses exclusively on Starlink, the methodology is applicable to other LEO constellations such as OneWeb, Amazon LEO, etc., as they become operational and measurement data accumulates. Extending the model to predict uplink performance would provide more comprehensive characterization for applications with symmetric bandwidth requirements. Finally, reducing prediction granularity to sub-hourly intervals and developing user-facing applications for optimal task scheduling represent natural extensions of this work.

Ethics. `Horizon` utilizes anonymized crowdsourced measurements in compliance with M-Lab and Cloudflare privacy policies. No personally identifiable information is collected or processed.

Acknowledgements. This research was supported by the National Growth Fund through the Dutch 6G flagship project "Future Network Services" and the Internet Society Pulse Research Fellowship.

References

[1] Administrația Națională de Meteorologie. Romanian National Meteorological Administration. https://

www.meteoromania.ro, 2025. Accessed: 2025-12-28.
[2] Amazon. Amazon Leo. https://www.aboutamazon.com/what-we-do/devices-services/amazon-leo, 2025. Accessed: 2025-11-15.
[3] Cristian Benghe, Vlad Graure, Tanya Shreedhar, and Nitinder Mohan. Horizon: Understanding and Predicting Global Starlink Performance – Dataset. https://doi.org/10.4121/0bf59468-e5cb-433f-aeb2-e04cf694b65c, 2026. Accessed: 2026-03-27.
[4] Cristian Benghe, Vlad Graure, Tanya Shreedhar, and Nitinder Mohan. Horizon: Understanding and Predicting Global Starlink Performance – Source Code. https://github.com/spear-lab/Horizon-Predicting-Starlink-Performance, 2026. Accessed: 2026-03-27.
[5] Rohan Bose, Saeed Fadaei, Nitinder Mohan, Mohamed Kassem, Nishanth Sastry, and Jörg Ott. It's a bird? It's a plane? It's CDN!: Investigating Content Delivery Networks in the LEO Satellite Networks Era. In *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks*, HotNets '24, page 1–9, New York, NY, USA, 2024. Association for Computing Machinery.
[6] Rohan Bose, Jinwei Zhao, Tanya Shreedhar, Jianping Pan, and Nitinder Mohan. Investigating Web Content Delivery Performance over Starlink. In *Proceedings of the ACM Web Conference*, 2026.
[7] Shkelzen Cakaj. The Parameters Comparison of the "Starlink" LEO Satellites Constellation for Different Orbital Shells. *Frontiers in Communications and Networks*, 2(643095), 2021.
[8] Cloudflare. Cloudflare Radar - 2025 year in review. https://radar.cloudflare.com/year-in-review/2025, 2026. Accessed: 2026-01-10.
[9] GeoPy Contributors. GeoPy: Python Geocoding Toolbox. https://github.com/geopy/geopy, 2023. Version: 2.4.1, Accessed: 2025-11-15.
[10] DataHub. Airport Codes Dataset. https://datahub.io/core/airport-codes, 2025. Accessed: 2025-12-28.
[11] Eutelsat OneWeb. Eutelsat OneWeb: High-performance multi-orbit satellite communications operator. https://www.eutelsat.com, 2025. Accessed: 2025-12-28.
[12] GeoNames. GeoNames Data. https://download.geonames.org/export/dump/, 2025. Accessed: 2025-12-28.
[13] Moinak Ghoshal, Omar Basit, Imran Khan, Z. Jonny Kong, Sizhe Wang, Yufei Feng, Phuc Dinh, Y. Charlie Hu, and Dimitrios Koutsonikolas. Replication: Performance of Cellular Networks on the Wheels. In *Proceedings of the 2025 ACM Internet Measurement Conference*, IMC '25, page 381–396, New York, NY, USA, 2025. Association for Computing Machinery.
[14] Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 507–520. Curran Associates, Inc., 2022.
[15] Matthias Hirth, Tobias Hoßfeld, Marco Mellia, Christian Schwartz, and Frank Lehrieder. Crowdsourced Network Measurements: Benefits and Best Practices. *Computer Networks*, 90:85–98, 2015.
[16] Geoff Huston. The ISP Column: Using LEOs and GEOs. https://www.potaroo.net/ispcol/2022-04/leogeo.html, 2022.
[17] Mohamed M. Kassem, Aravindh Raman, Diego Perino, and Nishanth Sastry. A Browser-side View of Starlink Connectivity. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC '22)*, pages 151–158. Association for Computing Machinery, 2022.
[18] T.S. Kelso. Frequently Asked Questions: Two-Line Element Set Format. https://celestrak.org/columns/v04n03/, 2024. Accessed: 2025-12-28.
[19] Koninklijk Nederlands Meteorologisch Instituut (KNMI). Royal Netherlands Meteorological Institute. https://www.knmi.nl, 2025. Accessed: 2025-12-28.
[20] Eric Lanfer, Dominic Laniewski, Daniel Otten, and Nils Aschenbruck. Weather-Based Link Prediction for LEO-Satellite Networks using the WetLinks Dataset. In *Proceedings of the 23rd IFIP/IEEE Networking Conference (IFIP Networking 2024)*, pages 586–588. IEEE, 2024.
[21] Dominic Laniewski, Eric Lanfer, Bernd Meijerink, Roland van Rijswijk-Deij, and Nils Aschenbruck. WetLinks: A Large-Scale Longitudinal Starlink Dataset with Contiguous Weather Data. In *Proceedings of the 8th Network Traffic Measurement and Analysis Conference (TMA 2024)*, pages 1–9. IEEE, 2024.
[22] Zhijing Li, Ana Nika, Xinyi Zhang, Yanzi Zhu, Yuanshun Yao, Ben Y. Zhao, and Haitao Zheng. Identifying Value in Crowdsourced Wireless Signal Measurements. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 607–616, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.
[23] Zikun Liu, Fan-Xue Gabriella Reidys, Sarah Tanveer, and Deepak Vasisht. Vivisecting Starlink Throughput: Measurement and Prediction. *Proc. ACM Netw.*, 3(CoNEXT4), November 2025.
[24] Lyntia. Telecommunications Technologies Helping Close the Digital Divide. https://www.lyntia.com/en/news/telecommunications-technologies-to-close-the-digital-divide/, 2024. Accessed: 2025-12-28.
[25] Sami Ma, Yi Ching Chou, Haoyuan Zhao, Long Chen, Xiaoqiang Ma, and Jiangchuan Liu. Network Characteristics of LEO Satellite Constellations: A Starlink-Based Measurement from End Users. In *Proceedings of the IEEE International*

Conference on Computer Communications (INFOCOM 2023), pages 1–10. IEEE, 2023.

[26] Kyle MacMillan, Tarun Mangla, James Saxon, Nicole P. Marwell, and Nick Feamster. A Comparative Analysis of Ookla Speedtest and Measurement Labs Network Diagnostic Test (NDT7). *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 7(1):1–26, 2023.

[27] Measurement Lab. NDT (Network Diagnostic Tool). https://www.measurementlab.net/tests/ndt/, 2025. Accessed: 2025-12-28.

[28] François Michel, Martino Trevisan, Danilo Giordano, and Olivier Bonaventure. A first look at starlink performance. In *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, page 130–136, New York, NY, USA, 2022. Association for Computing Machinery.

[29] David G. Michelson and Weiwen Liu. Simulation of rain fading and scintillation on Ka-band Earth-LEO satellite links. In *2009 Canadian Conference on Electrical and Computer Engineering*, pages 635–640, 2009.

[30] Nitinder Mohan, Andrew E. Ferguson, Hendrik Cech, Rohan Bose, Prakita Rayyan Renatin, Mahesh K. Marina, and Jörg Ott. A Multifaceted Look at Starlink Performance. In *Proceedings of the ACM Web Conference 2024 (WWW '24)*, pages 2723–2734. Association for Computing Machinery, 2024.

[31] Oliver Montenbruck and Eberhard Gill. *Satellite Orbits: Models, Methods and Applications*. Springer, 2000.

[32] Ilana Newman. For Rural Communities, Broadband Expansion Is No Single Thing. https://www.governing.com/infrastructure/for-rural-communities-broadband-expansion-is-no-single-thing, 2023. Accessed: 2025-12-28.

[33] Fangyan Nie and Jianqi Li. Image Threshold Segmentation with Jensen-Shannon Divergence and Its Application. *IAENG International Journal of Computer Science*, 49(1):200–206, 2022. Accessed: 2025-12-28.

[34] OpenStreetMap contributors. Nominatim: OpenStreetMap Search API. https://nominatim.openstreetmap.org/ui/search.html, 2025. Accessed: 2025-11-15.

[35] Zhuoliang Ou, Jiahao Zhong, Yongqiang Hao, Ruoxi Li, Xin Wan, Kang Wang, Jiawen Chen, Hao Han, Xingyan Song, Wenyu Du, and Yanyan Tang. Near-Real-Time Global Thermospheric Density Variations Unveiled by Starlink Ephemeris. *Remote Sensing*, 17(9), 2025.

[36] Jianping Pan and Jinwei Zhao. GeoFeed in the wild: A case study on StarlinkISP.net. https://www.ietf.org/slides/slides-ipgeows-paper-geofeed-in-the-wild-a-case-study-on-starlinkispnet-00.pdf, 2025. IAB Workshop on IP Address Geolocation (ipgeows). Accessed: 2026-01-12.

[37] Jianping Pan, Jinwei Zhao, and Lin Cai. Measuring a Low-Earth-Orbit Satellite Network. *arXiv preprint arXiv:2307.06863*, 2023.

[38] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[39] Timothy Pham. Assessment of Weather Effects on DSN Antenna Tracking at Ka-band. In *Proc. 4th International Conference on Advances in Satellite and Space Communications (SPACOMM)*, pages 12–15. IARIA, 2012.

[40] Brandon Rhodes. Skyfield: High precision astronomy for Python. https://rhodesmill.org/skyfield/, 2019. Accessed: 2025-12-28.

[41] Brandon Rhodes and David A. Vallado. sgp4: Satellite Orbit Propagation in Python. https://pypi.org/project/sgp4/, 2025. Accessed: 2025-12-28.

[42] Royal Netherlands Meteorological Institute (KNMI). Hourly, validated and automated in-situ ground-based meteorological observations in the Netherlands, Dataset version 1.0. https://dataplatform.knmi.nl/dataset/hourly-in-situ-meteorological-observations-validated-1-0, 2025. Accessed: 2025-12-28.

[43] SciPy. SciPy Jensen-Shannon Divergence. https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.jensenshannon.html, 2025. Accessed: 2025-12-28.

[44] SimpleMaps. World Cities Database. https://simplemaps.com/data/world-cities, 2025. Accessed: 2025-12-28.

[45] SpaceX. Starlink Availability Map. https://www.starlink.com/map, 2025. Accessed: 2026-01-11.

[46] SpaceX. Starlink Network Update. https://starlink.com/updates/network-update, 2025. Accessed: 2026-01-10.

[47] SpaceX. Starlink: Reliable High-Speed Internet from Space. https://www.starlink.com, 2025. Accessed: 2025-12-28.

[48] Hailong Su, Jinshu Su, Yusheng Xia, and Haibin Li. The Small-World Beneath LEO Satellite Coverage: Ground Hubs in Multi-Shell Constellations, 2025.

[49] Hammas Bin Tanveer, Mike Puchol, Rachee Singh, Antonio Bianchi, and Rishab Nithyanand. Making Sense of Constellations: Methodologies for Understanding Starlink's Scheduling Algorithms. In *Companion of the 19th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '23)*, pages 37–43. Association for Computing Machinery, 2023.

[50] Shubham Tiwari, Saksham Bhushan, Aryan Taneja, Mohamed Kassem, Cheng Luo, Cong Zhou, Zhiyuan He, Aravindh Raman, Nishanth Sastry, Lili Qiu, and Debopam Bhattacherjee. T3P: Demystifying Low-Earth Orbit Satellite Broadband, 2023.

[51] David Tuber. Measuring network quality to better understand the end-user experience. https://blog.cloudflare.com/aim-database-for-internet-quality/, 2023. Accessed: 2025-12-28.
[52] Uber Technologies, Inc. H3: Hexagonal Hierarchical Spatial Index. https://h3geo.org/, 2025. Accessed: 2025-12-28.
[53] Muhammad Asad Ullah, Antti Heikkinen, Mikko Uitto, Antti Anttonen, and Konstantin Mikhaylov. Impact of Weather on Satellite Communication: Evaluating Starlink Resilience. *arXiv preprint arXiv:2505.04772*, 2025.
[54] David A. Vallado, Paul Crawford, Richard Hujsak, and T. S. Kelso. Revisiting Spacetrack Report #3. *AIAA/AAS Astrodynamics Specialist Conference*, 2006.
[55] Sizhe Wang, Moinak Ghoshal, Yufei Feng, Imran Khan, Phuc Dinh, Omar Basit, Zhekun Yu, Y. Charlie Hu, and Dimitrios Koutsonikolas. Exploring the 5G Digital Divide in the Non-Contiguous US: LEO Satellites to the Rescue? *Proc. ACM Meas. Anal. Comput. Syst.*, 9(3), December 2025.
[56] Wikipedia contributors. Starlink. https://en.wikipedia.org/wiki/Starlink, 2026. Accessed: 2026-01-10.
[57] Joss Wright, Alexander Darer, and Oliver Farnan. On Identifying Anomalies in Tor Usage with Applications in Detecting Internet Censorship. In *Proceedings of the 10th ACM Conference on Web Science*, WebSci '18, page 87–96. ACM, May 2018.
[58] Jinwei Zhao. Starlink GeoIP Data: geoip/geoip-pops-ptr-latest.csv. https://github.com/clarkzjw/starlink-geoip-data/blob/master/geoip/geoip-pops-ptr-latest.csv, 2025. Accessed: 2025-11-23.
[59] Patrick Zippenfenig. Open-Meteo.com Weather API. https://open-meteo.com/, 2023. CC-BY-4.0; Creative Commons Attribution 4.0 International; Accessed: 2025-12-28.

A Network Measurements Schema Design

This appendix describes the unified schema design for integrating Cloudflare `AIM` and M-Lab `NDT7` measurements into a common data format suitable for analysis and model training.

A.1 Field Selection

We first surveyed prior research on Starlink performance measurement to identify the key metrics a telemetry system should capture. Pan et al. [37] used latency and ping measurements to evaluate Starlink's performance and explore network structure. Tanveer et al. [49] employed latency and packet loss to investigate Starlink's internal scheduling behavior. Mohan et al. [30] utilized upload and download throughput, latency, jitter, and packet loss to assess performance across applications such as gaming, streaming, and conferencing. Based on this survey, we identified throughput, latency, jitter, and packet loss as the core metrics for our schema.

A.1.1 Cloudflare `AIM` *Fields.* The Cloudflare `AIM` dataset provides both aggregated scores and raw measurement data; we use exclusively the raw measurements for finer granularity. The dataset schema is not flat: some metrics (throughput, latency) are stored as arrays with one value per test, while others are scalar values representing single measurements or aggregates across tests.

From the *upload* and *download* records, we extract arrays of transferred file sizes (bytes) and throughput values (bits per second). To capture network responsiveness under load, we use the *loadedLatencyMs* field, which contains arrays of round-trip latencies measured during active data transfer for both upload and download directions. The *loadedJitterMs* field captures latency variability, while *packetLoss.lossRatio* provides the proportion of packets lost during transmission.

For contextual metadata, we extract the autonomous system number (*clientASN*), geographic location (*clientCity*, *clientRegion*, *clientCountry*), and measurement timestamp (*measurementTime*, UTC). Server location is represented by the *serverPoP* field, an IATA airport code indicating the approximate server location.

A.1.2 `NDT7` *Fields.* In the `NDT7` dataset, upload and download tests are conducted independently, with only one direction populated per test record. During each test, the client and server periodically exchange status updates captured in the *raw* record, which contains *ServerMeasurements* and *ClientMeasurements* sub-records. To maintain alignment with Cloudflare's methodology, we focus on server-side measurements. The *TCPInfo* record contains kernel-level TCP metrics directly comparable to Cloudflare fields. We interpret *RTT* (round-trip time, microseconds) as loaded

Field Name	Data Type
uuid	varchar(255)
test_time	UTC timestamp
client_city	varchar(255)
client_region	varchar(255)
client_country_code	character(2)
server_city	varchar(255)
server_country_code	character(2)
asn	integer
data_source	varchar(255)
packet_loss_rate	numeric(10,5)
download_throughput_mbps	numeric(10,5)
download_latency_ms	integer
download_jitter_ms	numeric(10,5)
upload_throughput_mbps	numeric(10,5)
upload_latency_ms	integer
upload_jitter_ms	numeric(10,5)

Table 6. Unified schema for integrating M-Lab NDT7 and Cloudflare AIM measurement data into a common format. Latency and jitter are measured in milliseconds (ms), throughput in megabits per second (Mbps), and packet loss as a decimal ratio.

latency and *RTTVar* (RTT variance) as jitter. Throughput and packet loss are stored in the *a* record under *MeanThroughputMbps* and *LossRate* respectively. For metadata, we extract the UTC timestamp from the *a* record, client location from *Client.Geo* (city, region, ISO 3166-1 alpha-2 country code), client network from *Client.Network* (ASN), and server location from *Server.Geo* (city, country).

A.2 Schema Definition

Table 6 presents the unified schema. To ensure consistency, we standardize units across datasets: latency and jitter in milliseconds (ms), throughput in megabits per second (Mbps). Latency values are stored as integers; throughput, packet loss, and jitter as floating-point numbers rounded to five decimal places. Timestamps are standardized to UTC with second-level precision. Geographic data follows ISO 3166-1 alpha-2 for country codes. City and region names are normalized using the GeoNames *cities15000* dataset [12], which includes cities with populations exceeding 15,000 alongside standardized metadata. Regional names are sourced from the *admin1CodesASCII* dataset. We use ASCII versions exclusively to avoid issues with diacritics or non-Latin scripts. To align Cloudflare's *serverPoP* IATA codes with NDT7's city-based server metadata, we use the airport dataset from Datahub.io [10], which maps IATA codes to municipalities and countries. Although this does not provide exact server coordinates, it serves as a reasonable proxy for estimating client-server proximity.

A.3 Cross-Source Considerations

While the unified schema enables joint analysis of both datasets, important methodological differences remain. Cloudflare and NDT7 differ in measurement methodology, server infrastructure, test duration, and execution environment. All metrics are collected under different network conditions, making direct comparisons non-trivial.

To maintain analytical integrity, each record includes a `data_source` field identifying whether it originates from Cloudflare `AIM` or `NDT7`. This enables source-aware analyses and prevents inappropriate cross-dataset aggregation. The preprocessing and normalization procedures described in §3.2 address remaining differences in routing behavior and server selection.

B Weather API Validation

This appendix validates the accuracy of weather data obtained from the `OpenMeteo` API by comparing it against ground-truth observations from national meteorological institutes. We assess four variables relevant to Starlink performance prediction: temperature, precipitation, wind speed, and cloud cover.

B.1 Data Sources

We obtained observational data from the Royal Netherlands Meteorological Institute (KNMI) [19] and the Romanian National Meteorological Administration (ANM) [1].

For the Netherlands, we used the *Hourly, validated and automated in-situ ground-based meteorological observations* dataset [42], which provides expert-validated hourly measurements with missing values back-filled and known measurement errors corrected. Cloud cover in this dataset is reported as discrete values between 0 and 8 oktas, corresponding to increments of 12.5% per step. We accessed data via the KNMI API for 18 weather stations distributed across the Netherlands, covering the period 1–7 September 2025 (UTC).

For Romania, observational data is available on request for a limited number of stations and periods. We obtained data from two stations: Băneasa (Bucharest) and Piteşti, for 1 September 2025 (local time). These stations report nebulosity rather than cloud cover percentage; nebulosity is measured on a discrete scale from 0 to 10, corresponding to 10% increments.

B.2 Validation Results

Table 7 presents the MAE and RMSE between `OpenMeteo` and weather station observations for each meteorological variable.

Temperature exhibits strong agreement across all stations, with MAE values ranging from 0.68°C to 1.17°C for Dutch stations and 0.90°C to 1.10°C for Romanian stations. These errors are well within acceptable bounds for network performance modeling, where temperature variations of several degrees have minimal impact on prediction accuracy.

Precipitation shows similarly high accuracy, with MAE values below 0.5 mm for all stations. The low RMSE values (0.16–1.07 mm) indicate that `OpenMeteo` reliably captures both the occurrence and intensity of precipitation events, which is critical given the known sensitivity of Ku/Ka-band signals to rain fade.

Wind speed measurements demonstrate good agreement, with MAE values typically below 1 m/s for inland stations. Coastal stations such as Vlissingen exhibit higher errors (MAE 2.76 m/s), likely due to localized wind effects that are difficult to capture at the resolution of reanalysis models. For our purposes, these errors are acceptable since wind speed contributes a relatively small fraction to the Weather Index (§3.3.2).

Cloud cover exhibits the largest discrepancies, with MAE values ranging from 18.95% to 34.79%. These errors arise primarily from resolution differences: `OpenMeteo` provides cloud cover at 1% resolution, whereas KNMI and ANM report at 12.5% and 10% increments respectively. Additionally, cloud cover is inherently more spatially variable than temperature or precipitation, making point comparisons less informative. Despite these limitations, `OpenMeteo` captures the temporal dynamics of cloud cover sufficiently well for our prediction task, as evidenced by the feature importance analysis in §3.3.2.

Station	Coordinates		Temp. (°C)		Prec. (mm)		Wind (m/s)		Clouds (%)	
	Lat.	Lon.	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
NL Voorschoten	52.14	4.44	0.80	1.05	0.34	0.52	1.27	1.56	25.42	39.17
NL De Kooy Apt.	52.93	4.78	0.68	0.90	0.41	0.73	1.02	1.32	24.37	37.48
NL Schiphol Apt.	52.32	4.79	0.73	0.90	0.33	0.52	1.90	2.22	24.99	39.66
NL De Bilt	52.10	5.18	0.94	1.20	0.38	0.58	0.79	1.04	23.04	37.01
NL Stavoren	52.90	5.38	1.01	1.38	0.41	0.73	1.02	1.41	24.33	37.11
NL Lelystad Apt.	52.45	5.51	0.86	1.10	0.34	0.54	0.91	1.13	18.95	31.69
NL Leeuwarden Apt.	53.22	5.75	0.81	1.08	0.37	0.68	0.84	1.06	24.69	38.61
NL Deelen Apt.	52.05	5.87	1.04	1.30	0.37	0.77	0.99	1.31	21.89	35.01
NL Hoogeveen	52.75	6.57	1.03	1.32	0.41	0.71	0.91	1.08	21.02	33.40
NL Groningen Eelde	53.12	6.58	0.97	1.22	0.38	0.61	0.87	1.07	22.61	37.56
NL Twenthe Apt.	52.27	6.89	0.81	1.03	0.42	0.73	0.92	1.13	28.30	43.57
NL Vlissingen	51.44	3.60	1.17	1.59	0.45	1.07	2.76	3.06	23.74	37.48
NL Rotterdam Apt.	51.96	4.45	0.86	1.06	0.36	0.56	1.63	1.96	26.39	40.94
NL Gilze-Rijen Apt.	51.57	4.94	0.81	1.06	0.35	0.57	1.02	1.26	26.38	40.00
NL Eindhoven Apt.	51.45	5.38	0.90	1.17	0.33	0.52	0.81	1.02	21.84	33.84
NL Volkel Apt.	51.66	5.71	0.86	1.04	0.34	0.53	0.84	1.06	20.51	32.96
NL Ell	51.20	5.76	0.87	1.16	0.34	0.56	0.83	1.04	19.19	30.93
NL Maastricht Apt.	50.91	5.76	0.73	0.91	0.34	0.54	1.02	1.33	25.03	38.62
RO Băneasa	44.51	26.08	1.10	1.25	0.00	0.00	0.55	0.69	20.47	25.04
RO Piteşti	44.85	24.87	0.90	1.17	0.03	0.16	0.95	1.13	34.79	44.42

Table 7. Validation of `OpenMeteo` API against weather station observations. MAE and RMSE are computed from hourly measurements for temperature (°C), precipitation (mm), wind speed (m/s), and cloud cover (%).[6]

New York		Santiago		Berlin	
Latency	Throughput	Latency	Throughput	Latency	Throughput
8.86 (6.3-11.9)	41.73 (28.3-55.6)	15.11 (10.7-19.8)	20.17 (12.6-28.3)	8.36 (6.1-10.7)	10.72 (6.8-15.2)
9.87 (7.8-11.9)	32.84 (24.4-41.7)	19.25 (15.4-23.0)	20.42 (14.1-26.9)	8.60 (7.2-10.1)	16.38 (11.6-21.3)
7.35 (5.3-9.4)	32.49 (22.4-44.3)	18.73 (13.0-25.0)	18.54 (13.1-24.4)	9.99 (7.7-13.0)	13.13 (9.0-17.6)
9.19 (7.4-11.0)	49.03 (34.3-65.4)	14.94 (11.6-18.6)	26.34 (19.6-33.5)	8.83 (7.4-10.4)	14.22 (9.3-19.6)
8.69 (6.6-10.8)	56.10 (38.6-74.3)	15.04 (11.3-19.1)	26.32 (18.8-35.0)	7.85 (6.4-9.2)	8.99 (6.2-11.9)
8.92 (6.9-10.9)	35.11 (21.0-51.4)	15.23 (12.1-18.3)	30.28 (22.4-39.3)	7.97 (6.3-9.6)	11.82 (8.3-15.5)
7.14 (5.2-9.4)	36.45 (22.4-52.6)	9.79 (5.9-14.3)	23.18 (12.8-35.7)	7.79 (6.2-9.4)	11.00 (7.7-14.2)

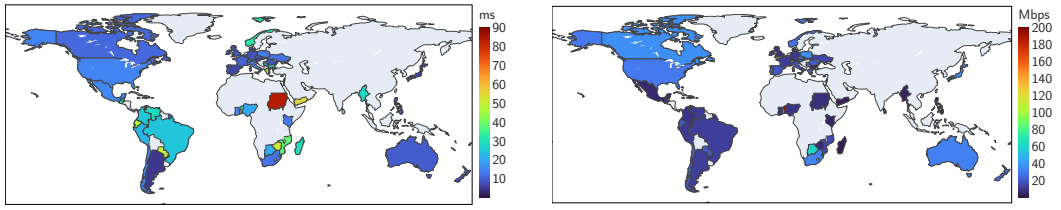
Table 8. Daily MAE of latency and throughput predictions for selected cities (rows correspond to 24–30 November 2025), with 95% confidence intervals computed from hourly predictions.

C Regional Mean Absolute Error Analysis

This appendix supplements the evaluation in §4.3 by reporting MAE. Figure 9 provides a global snapshot of country-level prediction accuracy for 24 November 2025, while table 8 details the daily MAE and 95% confidence intervals for selected cities throughout the full holdout week (24–30 November 2025).

Figure 9 presents country-level MAE for hourly predictions produced by `Horizon` on 24 November 2025. The spatial patterns closely mirror those observed in the RMSE evaluation. Both throughput and latency models achieve the lowest errors in North America, Europe, and Australia, where dense ground infrastructure and consistent measurement availability support reliable prediction. Throughput predictions are also accurate in Africa, Asia, and South America, partly because throughput levels in these regions tend to be lower and less variable (fig. 3b). Within Asia, Japan

[6]Netherlands: 1–7 September 2025 (UTC). Romania: 1 September 2025 local time (31 Aug 21:00 – 01 Sep 20:00 UTC).



a) Latency MAE (ms) b) Throughput MAE (Mbps)

Fig. 9. Country-level MAE for download latency (left) and throughput (right) on 24 November 2025.[7] Darker colors indicate higher error. North America, Europe, and Australia exhibit the lowest errors; Sudan and Yemen show the highest latency prediction error.

and the Philippines exhibit high accuracy owing to substantial measurement density and mature Starlink deployment. In contrast, Yemen and Myanmar experience higher errors, reflecting sparse measurement coverage and routing variability associated with limited regional PoPs. In Africa, performance is strongest in countries with established Starlink presence or proximity to PoPs, including Nigeria, South Africa, and Mozambique. Conversely, Sudan shows the highest latency MAE (exceeding 80 ms).

Table 8 presents the daily latency and throughput prediction MAE over the temporal holdout period (24–30 November), with 95% confidence intervals (CIs). Three geographically diverse cities are presented: New York (USA), Santiago (Chile), Berlin (Germany). The patterns mirror those observed for RMSE: errors are smaller and intervals are narrower for cities with large volumes of measurements and wider for Chile, where errors are higher. Throughput errors degrade more quickly than latency, and weekends generally show lower errors due to changing usage patterns.

D Prediction Coverage and Visualization

This appendix describes the spatial coverage of `Horizon` predictions and the visualization scheme used to communicate network quality to end users.

D.1 Spatial Coverage

`Horizon` generates predictions for level-2 `H3` hexagons [52], a hierarchical geospatial indexing system that partitions the Earth's surface into hexagonal cells. Level-2 hexagons have an average edge length of approximately 150 km, providing a balance between spatial resolution and computational tractability. Not all hexagons receive predictions; we apply filtering criteria to exclude regions where predictions would be unreliable or irrelevant.

A hexagon is included in the prediction set if it satisfies three conditions: (1) its centroid lies within 300 km of at least one training measurement point, ensuring sufficient nearby observations for reliable interpolation; (2) its area is not entirely over ocean, where Starlink service is unavailable to residential users; and (3) it falls within a country where Starlink operates [45]. Applying these criteria yields approximately 300 hexagons covering regions across six continents, as illustrated in fig. 10. Additionally, `Horizon` generates predictions for 270 major cities worldwide, selected to be representative of their respective countries and regions.

D.2 Quality of Service Visualization

The web interface displays predictions using a color-coded Quality of Service (QoS) scheme that maps predicted latency and throughput to discrete quality categories. This visualization enables

[7]Only countries present in both latency and throughput datasets are included. Differences in data collection periods and filtering steps lead to slight variations in country coverage.

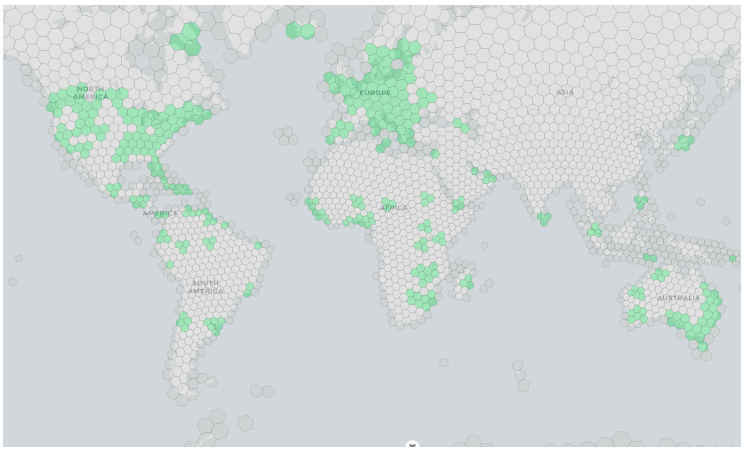


Fig. 10. Hexagons included in the Horizon prediction coverage (shown in green). Only hexagons satisfying the proximity, land coverage, and service availability criteria receive predictions.

Color	Latency (ms)	Throughput (Mbps)	Quality
	≤60	≥80	Excellent
	≤75	≥60	Good
	≤90	≥45	Fair
	≤110	≥30	Average
	≤140	≥20	Bad
	≤180	≥12	Very Bad
	>180	<12	Extreme/Unusable

Table 9. QoS color thresholds for predicted latency and throughput. Each hexagon is assigned the color corresponding to its lowest-quality metric (i.e., both latency and throughput must meet the threshold for a given quality level).

users to quickly assess expected network performance without requiring detailed numerical interpretation.

Table 9 defines the color thresholds for latency and throughput. Each hexagon is assigned the color corresponding to its lowest-quality metric, ensuring that predictions reflect the limiting factor for real-time communication (RTC) applications. For example, a location with excellent throughput (150 Mbps) but marginal latency (95 ms) would be colored yellow rather than green, alerting users to potential latency-sensitive application degradation. The thresholds are calibrated based on typical requirements for video conferencing, online gaming, and other interactive applications where both latency and bandwidth contribute to user experience.

Received January 2026; revised March 2026; accepted April 2026