
CESNET-TIME SERIES26: A MULTI-YEAR NETWORK TRAFFIC TIME SERIES DATASET FOR ANNUAL SEASONALITY, CONCEPT DRIFT, AND LONG-TERM ANOMALY DETECTION

Josef Koumar^{1,*}, Karel Hynek¹, Kamil Jeřábek², and Pavel Šiška²

¹Czech Technical University in Prague, Thákurova 9, 160 00 Prague 6, Czech Republic

²CESNET a.i.e., Generála Píky 430/26, 160 00 Prague 6, Czech Republic

*corresponding author: josef.koumar@fit.cvut.cz

ABSTRACT

Forecast-based models are fundamental to network traffic management and anomaly detection. However, it is essential to evaluate these forecasting models over extended periods of time in order to assess their adaptability to unexpected phenomena and shifts in underlying patterns. This kind of long-term evaluation requires multi-year data to account for yearly seasonality and behavioral drift. Existing public datasets are too short to support such long-horizon forecasting tasks. We introduce CESNET-TimeSeries26, comprising 995 days of continuous traffic time series from the CESNET3 network, together with a single subnet spanning approximately 4.5 years. The dataset captures 278,466 IP addresses, 285 institutions, and 566 institutional subnets at three aggregation intervals. The dataset provides a definitive longitudinal benchmark for evaluating forecasting models in both operational management and security monitoring. We demonstrate that annual seasonality is the dominant periodic component. Moreover, we show that a forecasting model can exploit the annual component only when its training window covers at least one complete annual cycle, which no sub-year dataset provides.

Background & Summary

Traffic monitoring is a cornerstone of network management and security operations [1]. As traffic encryption became ubiquitous, visibility into network communication decreased, and the research community increasingly turned to machine-learning-based classification of encrypted traffic [2, 3, 4, 5]. However, recent findings demonstrate substantial limits of supervised machine learning approaches used in the networking domain [6]. Unsupervised anomaly detection [7, 8, 9, 10, 11] can reveal malicious activities through behavioral changes in network entities without directly learning malicious patterns. A large family of unsupervised detectors is built on traffic forecasting, in which an alert is raised when the observed value deviates from the forecast by more than a specified threshold [9, 10, 11]. Therefore, the quality of such detectors is bound by the quality of the underlying forecasting models. Moreover, it is also directly bound by the data available to train and evaluate them.

The scarcity of long-term, real-world network time series has been repeatedly identified as the crucial obstacle in this area [12, 13]. Publicly available datasets are predominantly synthetic, which leads to overestimated performance and an “illusion of progress” [13], while real-world traces such as the MAWI/WIDE captures [14, 15] are limited to 15-minute daily snapshots. We addressed part of this gap with the CESNET-TimeSeries24 dataset [16], which provided 40 weeks of traffic time series for 275 thousand IP addresses of the CESNET3 network and has since been adopted by the community for benchmarking forecasting and anomaly detection methods. The CESNET-TimeSeries24 became immediately popular in the community with tens of reuses [17, 18, 19, 20, 21, 22, 23, 24].

However, a 40-week collection has a structural limitation: it does not contain a single complete annual cycle. Therefore, the annual seasonal component cannot be fitted, the recurrence of holiday and vacation patterns cannot be distinguished from one-off anomalies, and the multi-year drift of network behavior cannot be quantified at all. In an academic ISP

network, the annual rhythm of terms, exam periods, holidays, and vacations is the strongest low-frequency signal in the data. Our analyses confirm that annual seasonality is, in fact, the *dominant* periodic component for the vast majority of entities. Therefore, sub-year datasets are structurally blind to it.

This manuscript introduces CESNET-TimeSeries26, created by continuous long-term monitoring of the CESNET3 network over 995 days—from 9 October 2023 to 30 June 2026. The collection uses the identical monitoring infrastructure, capturing pipeline, metrics, anonymization, and processing pipeline as CESNET-TimeSeries24. The collection has been permanently closed due to a major change in the monitoring infrastructure, meaning new data points can no longer be added to the time series. Therefore, it is final and constitutes a stable, definitive reference for longitudinal studies. It was created from approximately 190.3 billion IP flows containing 13.6 trillion packets and 13.5 petabytes of transferred data, and provides time series for 278 thousand IP addresses, 285 institutions, and 566 institutional subnets at 10-minute, 1-hour, and 1-day aggregation intervals. In addition, the longitudinal part covers a single institutional subnet for which archived flow-level data reaches back to 2022, yielding approximately 4.5 years of continuous, TCP-only time series.

The scientific contribution of this descriptor is not that longer data yields incrementally better forecasts. In fact, we show that it does not in the point-forecast sense. The contribution is that a multi-year span makes an entire class of phenomena *measurable on public network data for the first time*. Multi-year public time series do exist in other domains, e.g., road traffic [25], retail sales [26], and electricity demand or pedestrian counts [27], yet the benchmarks most widely used for long-term forecasting span at most two to four years [28, 29, 30], and no comparable span has so far been publicly available for network traffic.

Concretely, our technical validation establishes four findings, each impossible in a sub-year collection: (i) annual seasonality is the dominant periodic component even stronger than the weekly component for 96.8% of institutions, 98.2% of institutional subnets, and 99.8% of dense IP series; (ii) the annual component of traffic cannot be exploited by a forecasting model trained on less than one year of data, but it becomes exploitable once the training window spans one complete annual cycle; (iii) most network entities exhibit statistically significant year-over-year distributional drift, and a slow multi-year structural shift is visible in metrics such as the average TTL; and (iv) a forecasting model deployed without retraining degrades by approximately 11.3 SMAPE points per quarter, implying a practical retraining interval of about two to three months. The Technical Validation presents the evidence for statements (i) and (iii), and its closing subsection, Relationship to CESNET-TimeSeries24, presents the evidence for statements (ii) and (iv). Table 1 summarizes which analyses each observation length unlocks and locates CESNET-TimeSeries26 relative to sub-year traces and the previous 40-week collection.

Methods

In this section, we describe the methods used to obtain the CESNET-TimeSeries26 dataset. The collection methodology is identical to the one used for CESNET-TimeSeries24 [16]. We summarize it here for self-containment and describe in detail the aspects specific to the multi-year collection.

Ethical statement

Since the dataset originates from a production network used by real users, privacy was a fundamental concern throughout our work. We used exclusively automated data processing with immediate anonymization. Therefore, we did not analyze or manually process non-anonymized data at any point, nor did we perform any procedure that would allow us to track users or reveal their identities.

The publication of the dataset has been approved by the Committee for Ethics in Research of the Scientific Council of the Czech Technical University in Prague under reference number 0000-11/26/51902/EKČVUT. The approval includes a waiver of explicit user consent for publishing the dataset. Moreover, all users of the CESNET3 network agreed to terms and conditions that define a monitoring process for the optimization and improvement of services (including related research) and allow sharing of the data with third parties after anonymization (<https://www.cesnet.cz/en/gdpr>).

Data capture

The network traffic was obtained from the ISP network CESNET3, which provides internet access to public and research institutions in the Czech Republic. The network spans the whole country, as shown in Figure 2, and serves approximately half a million users daily. Because packet-based monitoring is infeasible at the ISP scale, CESNET3 is monitored with a standard IP flow monitoring system located at the network perimeter. All transit lines to peering partners are equipped with flow monitoring probes.

	MAWI (15 min/day, 2007-2024)	SDN (sim.) (4 days)	Operator KPIs (1 month)	Telecom Italia (2 months)	CESNET-AGG23 (10 weeks)	GEANT (16 weeks)	Abilene (6 months)	TS24 (40 weeks)	BRAIN (375 days)	TS26 main (2.72 yr / 3 cyc)	TS26 subnet (4.5 yr / 5 cyc)
Weekly seasonality	~ 15 min/day	✗ <1 week	✓ permits	✓ permits	✓ permits	✓ permits	✓ permits	✓ permits	✓ permits	✓ shown here	✓ shown here
Holiday effects	~ 15 min/day	✗ <1 week	✓ permits	✓ permits	✓ permits	✓ permits	✓ permits	✓ permits	✓ permits	✓ shown here	✓ shown here
Annual seasonality	~ 15 min/day	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	~ partial 2nd yr	✓ shown here	✓ shown here
Cross-cycle reproducibility	~ 15 min/day	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ only 1 cycle	✓ shown here	✓ shown here
Year-over-year drift	~ 15 min/day	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ only 1 cycle	✓ shown here	✓ shown here
Leave-one-cycle-out anomaly separation	~ 15 min/day	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ only 1 cycle	✓ shown here	✓ shown here
Full-year-ahead forecast	~ 15 min/day	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ <1 cycle	✗ only 1 cycle	~ only 1 cycle to train	✓ shown here
	limited (partial)		demonstrated	feasible (length permits)	feasible (length permits)	feasible (length permits)	feasible (length permits)	feasible (length permits)	feasible (length permits)	feasible (length permits)	feasible (length permits)
				infeasible (reason in cell)	infeasible (reason in cell)	infeasible (reason in cell)	infeasible (reason in cell)	infeasible (reason in cell)	infeasible (reason in cell)	infeasible (reason in cell)	infeasible (reason in cell)

Figure 1: Analyses unlocked by a length of dataset. Columns are eleven network-traffic datasets ordered by span: MAWI [15]; SDN [31]; Network Operator KPIs [32]; the Telecom Italia [33]; CESNET-AGG23 [34]; GEANT [35]; Abilene [36]; the 40-week CESNET-TimeSeries24 [16] (TS24); BRAIN [37]; CESNET-TimeSeries26 [38] main part (TS26 main); and CESNET-TimeSeries26 [38] longitudinal subnet part (TS26 subnet). A MAWI archive of 15-minute daily snapshots is included separately: although it spans calendar years (2007–2024), each captured day is only 15 m inutes long, so it never accumulates a single continuous cycle. A full annual cycle is required for annual seasonality. Two cycles for cross-cycle reproducibility, year-over-year drift, and leave-one-cycle-out anomaly separation. Three cycles for a full-year-ahead forecast validated on a held-out complete cycle. Cells marked *demonstrated* are validated in this article. The *feasible* cells are supported by the data length but were not separately run. The *limited* cells denote partial capability. And finally, the *infeasible* cells state the limiting reason.

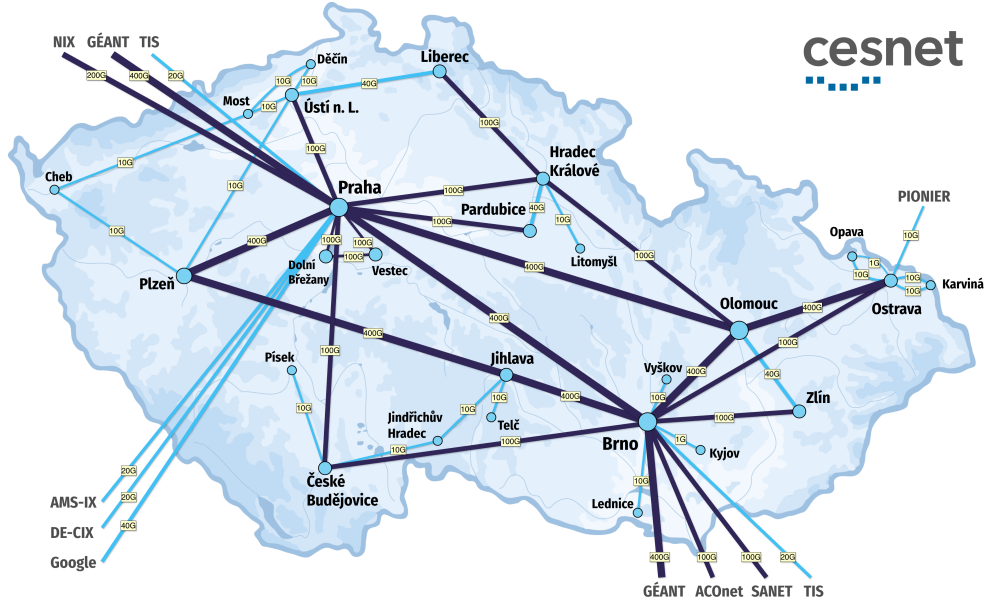


Figure 2: Topology of the CESNET3 network, which interconnects academic institutions in the Czech Republic.

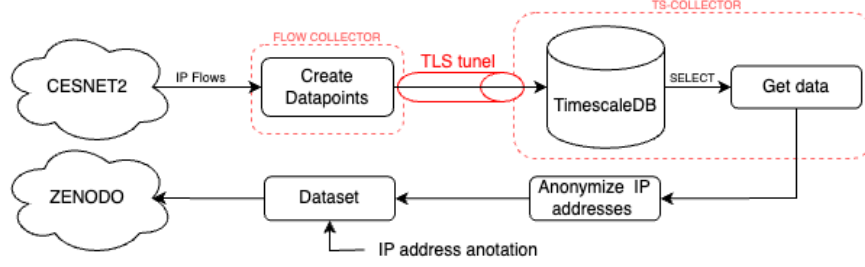


Figure 3: Architecture of dataset collection from the CESNET3 network.

Table 1: Daylight-saving transitions within the CESNET-TimeSeries26 observation window (Europe/Prague). Spring-forward events skip a local hour and are transparent on the absolute (UTC) time grid; fall-back events repeat a local hour and are identified by the -1 h offset change.

Date	Transition	Local clock change	Offset change
2022-03-27	spring forward	02:00 CET $\rightarrow$ 03:00 CEST	+1 h
2022-10-30	fall back	03:00 CEST $\rightarrow$ 02:00 CET	-1 h
2023-03-26	spring forward	02:00 CET $\rightarrow$ 03:00 CEST	+1 h
2023-10-29	fall back	03:00 CEST $\rightarrow$ 02:00 CET	-1 h
2024-03-31	spring forward	02:00 CET $\rightarrow$ 03:00 CEST	+1 h
2024-10-27	fall back	03:00 CEST $\rightarrow$ 02:00 CET	-1 h
2025-03-30	spring forward	02:00 CET $\rightarrow$ 03:00 CEST	+1 h
2025-10-26	fall back	03:00 CEST $\rightarrow$ 02:00 CET	-1 h
2026-03-29	spring forward	02:00 CET $\rightarrow$ 03:00 CEST	+1 h

IP flow monitoring systems aggregate packets into IP flow records. An IP flow record represents communication metadata associated with a single connection and is defined [39] as “a set of IP packets passing an observation point in the network during a certain time interval, such that all packets belonging to a particular flow have a set of common properties”. The flow key is defined by source and destination IP addresses, transport-layer ports, and protocol.

The monitoring infrastructure is detailed in Figure 3. Network TAPs positioned before the edge routers mirror the traffic to monitoring probes, which are servers equipped with high-speed monitoring cards capable of handling 200 Gb/s. On the servers, the ipfixprobe flow exporter (<https://github.com/CESNET/ipfixprobe>) is running with a configuration of an active timeout of 5 minutes and an inactive (idle) timeout of 65 seconds, which is standard practice for flow-based monitoring [39]. The exporter collected the following features: start time, end time, number of packets, number of bytes, and Time to Live (TTL). None of the collected information contains application-level payload, which ensures that the privacy of user communication is not compromised. The collected records were transferred over a secure (TLS) IPFIX [39] channel to a flow collector running IPFIXcol2 (<https://github.com/CESNET/ipfixcol2>).

Continuity of the measurement infrastructure

Because the dataset is intended to enable users to distinguish genuine network phenomena from measurement artifacts, we characterize the continuity of the capture pipeline over the full capture window. Before doing so, we fix the temporal reference frame, since the window spans several civil-time changes.

The Czech Republic observes Central European Time (CET, UTC+1) in winter and Central European Summer Time (CEST, UTC+2) in summer, switching on the last Sunday of March and October. Six such transitions fall within the 995-day main window, and nine within the 2022–2026 span of the longitudinal subnet; all nine are listed in Table 1. The dataset’s canonical time index is defined on a continuous absolute (UTC) grid, and the Europe/Prague local timestamps reported in this section and in Table 2 are derived from it. On this grid, the spring-forward transitions are fully transparent with no window created or lost. However, the fallback transitions repeat within a local hour and would each appear as a spurious one-hour discontinuity if diagnosed by local clock time. Therefore, we identify them by the accompanying -1 h change in UTC offset rather than by the apparent gap length, and treat them as timekeeping artifacts rather than outages.

During the multi-year capture, the monitoring infrastructure was occasionally unavailable due to software or hardware maintenance and transient faults. Table 2 lists the outages. In total, 13 outages are reported, amounting to approximately 31.5 days of missing observation. Five span more than a day, four last between half a day and a day, and four last

Table 2: Monitoring outages in CESNET-TimeSeries26. Times are Europe/Prague local. The Part column states which dataset part the outage affects. Shorter gaps and daylight-saving transition artifacts are omitted. The 2024 probe outage was a partial capacity loss (total never reached zero) and is discussed in the text rather than listed here.

Outage	Part	Last data before	Data resumed	Duration
1	subnet	2022-01-11 12:43	2022-01-12 12:43	23.8 h
2	subnet	2022-03-14 22:53	2022-03-15 13:33	14.5 h
3	subnet	2022-05-18 21:53	2022-05-19 21:43	23.7 h
4	subnet	2022-12-11 17:13	2022-12-14 10:23	2 d 17.0 h
5	subnet	2022-12-28 09:53	2023-01-02 14:33	5 d 4.5 h
6	subnet	2023-04-12 01:33	2023-04-14 05:43	2 d 4.0 h
8	main, subnet	2025-10-05 11:57	2025-10-05 13:09	1.0 h
7	main, subnet	2025-10-05 14:13	2025-10-06 05:54	15.5 h
9	main, subnet	2025-10-07 08:17	2025-10-07 10:07	1.7 h
10	main, subnet	2026-03-06 15:07	2026-03-08 18:17	2 d 3.0 h
11	main, subnet	2026-04-24 07:50	2026-04-24 09:38	1.7 h
12	main, subnet	2026-04-29 10:38	2026-05-15 07:46	15 d 21.0 h
13	main, subnet	2026-05-22 11:56	2026-05-22 13:16	1.2 h

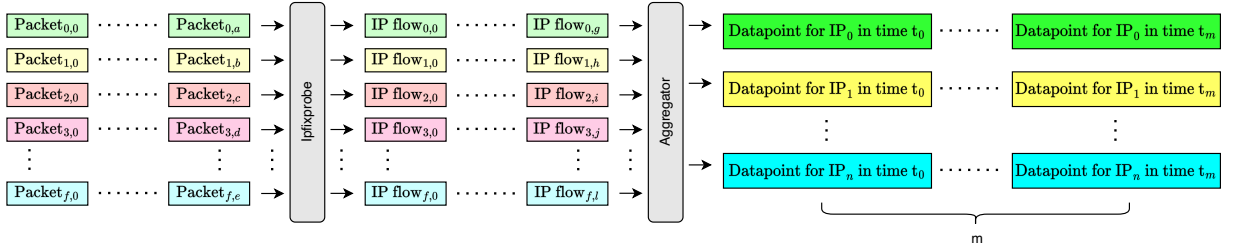


Figure 4: The aggregation process for capturing time series of length m from network traffic. For each packet $Packet_{i,j}$, there exists a $Flow_{i,k}$ to which the packet belongs. It always holds that $a \geq g$ (and similarly $b \geq h$), and in most cases a is much larger than g ; commonly, a single IP flow contains all packets of one connection ($g = 0$), e.g., for a user visiting a web page. A time series datapoint is likewise a combination of one or more IP flows.

one to two hours. Outages 1–6 fall before the start of the main collection and therefore affect only the longitudinal subnet. Outages 7–13 fall within the main window and affect both parts. Shorter gaps and daylight-saving transition artifacts are omitted as negligible. These interruptions are an unavoidable consequence of multi-year operation, where maintenance and faults occur naturally. Users requiring strict continuity can mask these intervals directly from the shared time grid. We do not recommend imputation, particularly for the 15-day and 5-day gaps.

Moreover, there was a probe outage from 2024-05-21 to 2024-06-04, which was a *partial* capacity loss rather than a complete blackout. The institution’s total fell to a median of about 43 % of its baseline (minimum 6.4 GB per 10-minute window) but never reached zero and produced no grid gap. It is consequently absent from the zero-traffic table above and is instead flagged in the known-measurement-artifacts documentation as a reduced-capacity event, so that users do not mistake the partial recovery for a genuine level shift or trend anomaly.

Time series aggregation

The flow collector runs the aggregation module, as depicted in Figure 3. The aggregation module also removes transient traffic (flows in which neither endpoint belongs to CESNET3) and all single TCP SYN packet connections, i.e., scans. Given the probe placement before routers and firewalls, scans would otherwise dominate the collected flows and introduce significant noise into the time series. Since scans are reliably detectable by simple, dedicated methods [40, 41], we exclude them. All remaining flows are passed to aggregation.

We generate evenly spaced time series for each IP address by aggregating IP flow records into data points representing the address’s behavior within 10-minute time windows. This process is shown in Figure 4. The vector of time-series metrics $v_{ip,i}$ describes the IP address ip in the i -th window, i.e., between t_i and t_{i+1} . Aggregated datapoints are stored in TimeScaleDB (<https://www.timescale.com/>), from which the time series are built.

Datapoints contain the following time-series metrics:

- **Simple volumetric metrics:** the number of IP flows, the number of packets, and the transmitted data size (number of bytes).
- **Unique volumetric metrics:** the number of unique destination IP addresses, unique destination Autonomous System Numbers (ASNs), and unique destination transport-layer ports. Aggregation of these metrics is memory-intensive since all unique values must be kept in memory; a server with 41 GB of RAM was sufficient for 10-minute aggregation at ISP scale.
- **Ratio metrics:** the ratio of UDP/TCP packets, the ratio of UDP/TCP transmitted data size, the direction ratio of packets, and the direction ratio of transmitted data size.
- **Average metrics:** the average flow duration and the average Time To Live (TTL).

Anonymization

The capture process started on 9 October 2023 and ended on 30 June 2026, i.e., after 995 days of continuous collection, at which point we extracted the time series from the database. On 30 June 2026, the capture infrastructure was decommissioned due to major changes to the monitoring infrastructure that prevented it from continuing to capture long same-origin time series. A script deployed within the database framework automatically assigns a corresponding institution and institutional subnet to each IP address. We do not extract or store IP addresses, institution names, or exact subnets. Instead, randomly assigned database identifiers are used for all three entity types. Without access to the mapping table within the database framework, this step cannot be reversed, and the published time series cannot be linked to a specific IP address, institution, or institutional network.

Longitudinal subnet collection

For one institutional subnet with 255 addresses (subnet with network mask /24, approximately 100 active), archived TCP flow-level data were available from an earlier deployment of the same monitoring infrastructure, starting in 1 January 2022. We re-aggregated these archived IP flow records into time-series data points using the same filtration and aggregation pipeline, metrics, time windows, and anonymization as for other time series in this dataset. The result spans approximately 4.5 calendar years and is published as a separate part of the dataset. Thus, it contains four complete calendar-year cycles and a partial fifth. Moreover, keyed to the academic year (starting approximately 20 September in Czechia), the span covers three complete annual cycles (2022–23, 2023–24, 2024–25) plus a partial fourth (2025–26), which is what our cross-cycle analyses use.

The deployment that produces this dataset part exports *only TCP flows*. As a consequence, all volumetric and average metrics of this part (`n_flows`, `n_packets`, `n_bytes`, `n_dest_ip`, `n_dest_asn`, `n_dest_port`, `avg_duration`, `avg_ttl`) reflect TCP traffic exclusively, and their absolute levels are not comparable with the all-protocol parts of the dataset. The two transport-ratio metrics (`tcp_udp_ratio_packets` and `tcp_udp_ratio_bytes`) are constant and equal to 1 over the whole span. They are retained solely for format compatibility with the other parts and carry no information for this part.

This longitudinal part is self-contained. It covers the 2022–2026 period, including the interval that overlaps with the main dataset. Self-containment duplicates the 2023–2026 data for these addresses (which also appear, as all-protocol series, in the main IP-address and subnet parts) but spares users the need to stitch two sources. The overlapping period is *not* identical between the two views. The main dataset part contains all metrics and is produced from all-protocol traffic. The anonymized identifiers for these addresses in the longitudinal part are identical to those in the main dataset, so users can relate the two views while remaining aware of the TCP-only versus all-protocol difference.

Data preprocessing

Preprocessing consisted of: *i*) time series filtering, *ii*) creation of multiple time aggregations, *iii*) creation of institution time series, *iv*) creation of institutional subnet time series, *v*) creation of sample IP addresses datasets, and *vi*) adding information about weekends and public holidays, school terms and exam periods, and European time changes.

Filtering CESNET-TimeSeries24 removed series with more than 99.75% empty datapoints, resulting in dropping time series with fewer than 100 datapoints. We treat this published rule as a *relative* principle rather than as the absolute constant 100, and recompute it for the 995-day window in the main dataset and the 1651-day window in the longitudinal subnet. With $995 \times 144 = 143,280$ and $1651 \times 144 = 237,744$ theoretical 10-minute windows, the threshold becomes $0.25\% \times 143,280 \approx 358$ and $0.25\% \times 237,744 \approx 594$. Thus, we remove series with at most 358 non-empty 10-minute datapoints from the main part and 594 non-empty 10-minute datapoints from the longitudinal subnet part. After filtering,

the main dataset contains time series for 278,466 IP addresses, 285 institutions, and 566 institution subnets. The longitudinal subnet part contains 135 IP address series.

Multiple time aggregation The finest aggregation level groups traffic into 10-minute windows. The window length is bounded by the flow-export process because the time series are built from IP flow records exported with an active timeout of 5 minutes. So a long-lived connection appears as a sequence of records, each carrying up to 5 minutes of traffic. Therefore, windows shorter than the active timeout would distribute the volume of such flows unevenly. Moreover, it would produce spikes and empty intervals that were not actually observed in traffic. Because export boundaries are not aligned with window boundaries, even a 5-minute window would not be enough. Therefore, we set the window to twice the active timeout, which keeps this effect small. Moreover, 10-minute intervals can be too short for longitudinal methods. Therefore, we additionally provide 1-hour and 1-day aggregations derived from the 10-minute data. Where possible, metrics were aggregated by summing values (IP flows, packets, transmitted bytes). For metrics that represent counts of unique values, which cannot be summed without information loss, we provide the sum, mean, and standard deviation, resulting in three derived metrics per original metric. Ratio and average metrics are aggregated using the mean.

Time series of institutions Many security events are visible only from the perspective of overall network traffic. Therefore, we use the institution ID exported from the database to group IP addresses by institution. We identify 285 institutions within the CESNET3 network. From which 283 are unchanged from CESNET-TimeSeries24, and 2 new institutions were added during the longer-term capture pipeline.

Time series of institutional subnets Institutions often operate multiple networks in different locations or for different purposes, and administrators commonly handle security per subnet. We therefore also group IP addresses by institutional subnets using information from CESNET. We identify 566 institutional subnets within the CESNET3 network. From which 548 are unchanged from CESNET-TimeSeries24, and 18 new institution subnets were added during the longer-term capture pipeline.

Creation of sample IP addresses datasets CESNET-TimeSeries24 provided a single uniformly random sample of IP address series, which is usable, for example, for building a method without evaluation on the whole corpus, while still comparing the method with related works. And if the method had promising results on the sample dataset, then the long computation on the whole corpus makes sense.

We retain a comparable uniform-random baseline subset (`ip_addresses_sample`), which reproduces the original sampling. Moreover, we additionally publish five subsets stratified by percentage of naturally occurring gaps in network traffic time series, so that methods can be evaluated across the full range of gap severity rather than only on the “average” series. For each series, we compute a gap ratio score on the 10-minute grid. If the gap ratio is equal to 0, it denotes a fully dense series, and values approaching 1 denote a mostly absent one. We partition $[0, 1]$ into five equidistant bands of width 0.20 (`gaps_20_00`: $[0.00, 0.20]$ near-complete to dense; `gaps_40_20`: $(0.20, 0.40]$; `gaps_60_40`: $(0.40, 0.60]$; `gaps_80_60`: $(0.60, 0.80]$; `gaps_100_80`: $(0.80, 1.00]$ very sparse). From each band, we draw $N = 1000$ series with a single numpy random generator (seed 42) consumed in a fixed order over an `id_ip`-sorted candidate pool, making the draw independent of filesystem ordering. The candidate pools all exceed N (7,852, 10,869, 12,321, 19,037, and 228,387 series across the five bands, from densest to sparsest). Thus, every subset is exactly sized with no out-of-band members. Each sampled series is materialized identically across all three aggregations to keep the subsets comparable.

Weekends and public holidays Information about non-working days is crucial for training and evaluating forecasting and anomaly detection models. We include weekend dates and Czech public holidays for the entire collection period, i.e., the years 2022–2026. Because of the multi-year span, every annual holiday (Christmas, Easter, state holidays) occurs at least twice in the main dataset and at least four times in the subnet, enabling us to distinguish recurring calendar effects from one-off anomalies.

School terms and exam periods CESNET3 interconnects universities, academic institutions, high schools, student dormitories, and other public institutions in the Czech Republic. Therefore, its traffic closely follows the academic calendar. CESNET-TimeSeries24 could only comment on this correlation, whereas the multi-year span provides the calendar explicitly and observes each phase repeatedly. For the whole collection period, we annotate every date with the prevailing Czech academic-year regime: autumn (winter-semester) teaching term, winter examination period, spring (summer-semester) teaching term, summer examination period, and vacation. Following the official academic calendars of the covered years and keyed to the Czech term start, mostly in the second half of September, however, the start of the winter term is not common for all universities in Czechia. Thus, it can deviate by a week.

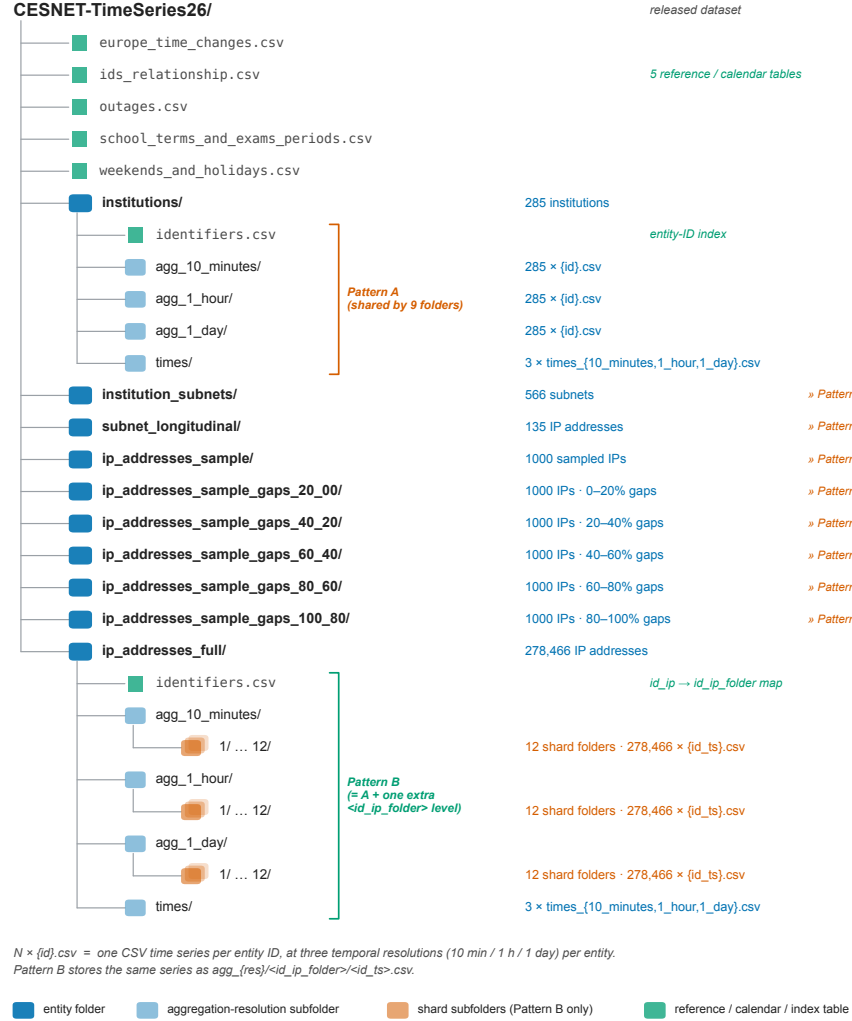


Figure 5: The file structure of the CESNET-TimeSeries26 dataset.

Each annual cycle appears at least twice in the main dataset (academic years 2023/24, 2024/25, and a partial 2025/26) and four times in the longitudinal subnet (2022/23 through 2025/26). This information lets users condition training and evaluation on the operative regime and, because each phase recurs, separate the stable academic structure from one-off events.

European time changes The Czech Republic changes its clocks twice a year. On the last Sunday of March, clocks move one hour forward (CET to CEST). On the last Sunday of October, they move one hour back (CEST to CET). The time series in the dataset are not affected by these changes directly. Each series is indexed by a sequence number of its aggregation window, and the windows follow an absolute timeline. The times translation tables map each sequence number to the corresponding wall-clock time. Therefore, no windows are added or removed during a clock change. However, the wall-clock time associated with a given window index shifts by one hour with each change. Because human activity follows local time, the daily and weekly patterns in the data also shift by one hour relative to the window index at each change. Therefore, we provide the exact instances of all clock changes in the 2022–2026 period. Users can align the windows with local time and avoid mistaking this one-hour shift for a real change in a series’ behavior.

Data Records

The dataset is delivered as compressed CSV files and is available on the Zenodo platform [38]: <https://doi.org/10.5281/zenodo.21945629>. To simplify adoption, CESNET-TimeSeries26 is also integrated into the CESNET-TSZoo

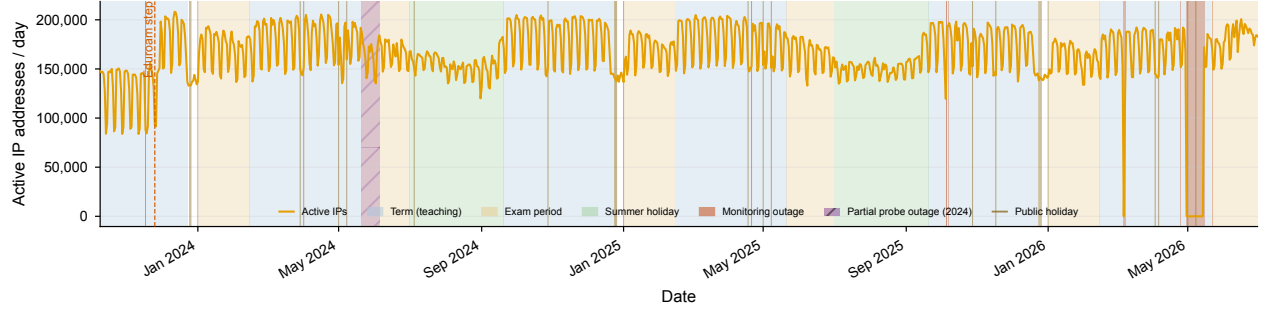


Figure 6: Daily count of active IP addresses over the 995-day collection, with known infrastructure events annotated. The November 2023 Eduroam architecture change produces a permanent level shift; the recurring Christmas and summer minima recur in every observed year.

library [42]: <https://github.com/CESNET/cesnet-tszoo>. This section describes the dataset structure and all data fields and files. The overall size of the data is 473.7 GB (141 GB compressed).

Data file hierarchy

The dataset structure is outlined in Figure 5. Each time series type (institutional subnets, institutions, individual IP addresses, samples of IP addresses, and the longitudinal subnet) is divided by aggregation window (10 minutes, 1 hour, 1 day). Time series are stored in CSV files named by the identifier of the corresponding entity. Because identifiers are not consecutive due to filtering, each time series type has its own `identifiers.csv` file listing all included entities.

The individual IP address (network entity) time series are further divided into subdirectories, each containing 25,000 files, since some file systems have difficulty handling hundreds of thousands of files in a single directory. Thus, the `identifiers.csv` also includes the subdirectory name where the corresponding series is stored.

The dataset also contains a `times` directory for each dataset part, with translation tables that map sequence numbers in the aggregation window to absolute time. Moreover, the dataset contains a relationship table linking IP address IDs, institution IDs, and institutional subnet IDs, a table listing public holidays and weekends, a table listing school terms and exam periods, a table listing European time changes, and a table listing outages of monitoring infrastructure during the capture period.

File format

The dataset files are organized as tables in CSV format. There are two file formats for the time series: series aggregated over 10 minutes contain the columns described in Table 6, and series aggregated over one hour and one day additionally contain the columns described in Table 7, which replace the unique-count metrics. The dataset further contains support files:

`ids_relationship.csv` (Table 8)

`weekends_and_holidays.csv` (Table 9)

`school_terms_and_exams_periods.csv` (Table 10)

`europe_time_changes.csv` (Table 11)

`outages.csv` (Table 12)

Technical Validation

This section verifies the properties of the published data in five steps. First, we check the overall properties of the collection. Second, we verify the presence of long-term multi-year behaviors. Third, we verify that repeated annual cycles are separable from anomalies. Fourth, we validate the 4.5-year longitudinal subnet. Fifth, we validate the stratified sample subsets. A closing subsection then states how the dataset relates to its 40-week predecessor. Unless stated otherwise, analyses use the 1-day aggregation, and we mask the outage days documented in Table 2 before any statistical testing.

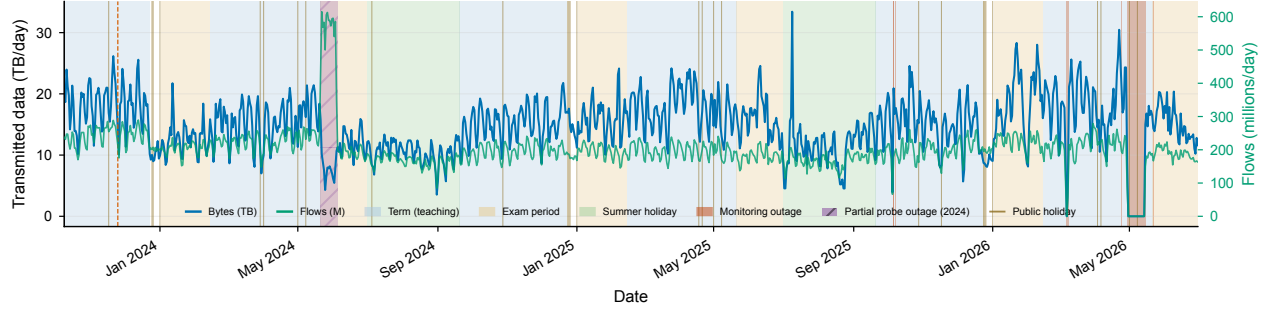


Figure 7: Daily total transmitted data over the collection. The academic calendar is directly visible, superimposed on a slow year-over-year growth trend.

Validation of overall dataset properties

Activity of IP addresses

Figure 6 shows the daily count of active IP addresses over the full window, annotated with known events. Two observations matter for users. First, the November 2023 Eduroam public-IP architecture change was already documented in CESNET-TimeSeries24. However, now it is observable over a much longer horizon because the elevated count is sustained throughout the remaining 2.5 years and never reverts. Therefore, this “a level shift of unknown persistence” identified on a 40-week window is now confirmed to be a permanent architectural change rather than a transient anomaly.

Furthermore, users of the dataset can use this labeled, expert-verified example of a genuine structural break as ground truth for evaluating change-point detectors. Second, the number of active IP addresses itself is seasonal. We can clearly see both weekly and annual seasonalities. For example, pronounced minima recur at every Christmas and every summer vacation, and the difference between teaching terms and examination periods is visible in the count alone. The practical consequence is that the population of “present” entities is not stationary. Therefore, per-entity analyses that implicitly assume a fixed device population (for example, cross-sectional feature distributions computed per day) will absorb this population seasonality unless they control for it.

Evolution of transmitted data

The dataset is built from approximately 190.3 billion IP flows carrying 13.6 trillion packets and 13.5 petabytes of transferred data. Overall, only the IP addresses part comprises 4.5 billion 10-minute datapoint rows. Figure 7 plots the daily total transmitted data (metric `n_bytes` summed over all institutional time series). The academic rhythm dominates the profile with three autumn terms, three winter examination periods, two complete summer vacations, and three Christmas minima.

Moreover, each annual seasonal condition recurs with the same direction and comparable depth in every observed year. We can also observe a slow secular growth of daily volume, quantified at +0.85 TB/day per year.

Gaps in time series

As in the CESNET-TimeSeries24 dataset, most 10-minute series are sparse, as can be seen in Figure 8. Datapoints occupy, on average, 13.7 % of the 10-minute grid, with a median of only 4.6 %, rising to 14.0 % at the 1-hour aggregation and 58.1 % at the 1-day aggregation. The distribution is not merely skewed but bimodal, and the two modes have a physical identity.

First mode, a sparse mode produced by end-user devices that are switched on during working hours, and a dense mode produced by always-on servers and infrastructure. Compared with the 40-week collection, the overall shape of the distribution is preserved, but both modes migrated. The peak of the sparse mode formerly located at 60–70 % presence now sits at 80–100 %, partially merging with the dense mode. The likely cause is that entities observable across 2.7 years are disproportionately permanently connected machines, while intermittently used devices accumulate absence. Two consequences follow. First, the appropriate aggregation is method-dependent: at 10 minutes, the median series is 95 % empty and only gap-robust methods are applicable, whereas at 1 day, the median series is majority-present and classical forecasting becomes meaningful.

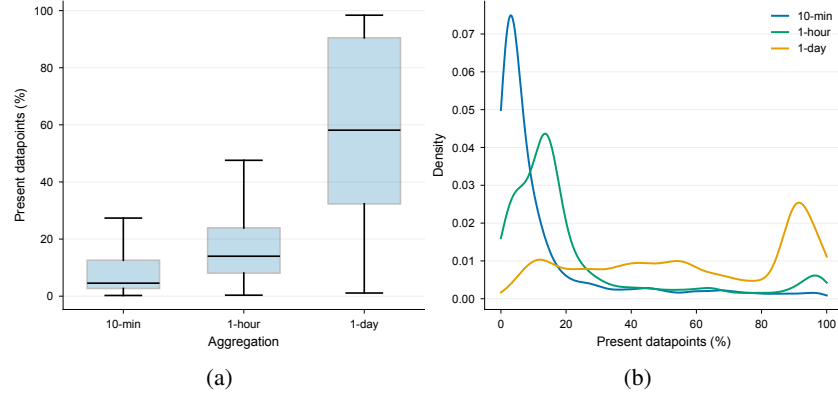


Figure 8: Distribution of the fraction of present datapoints per series over the 995 days: (a) boxplot by aggregation, (b) kernel density estimate.

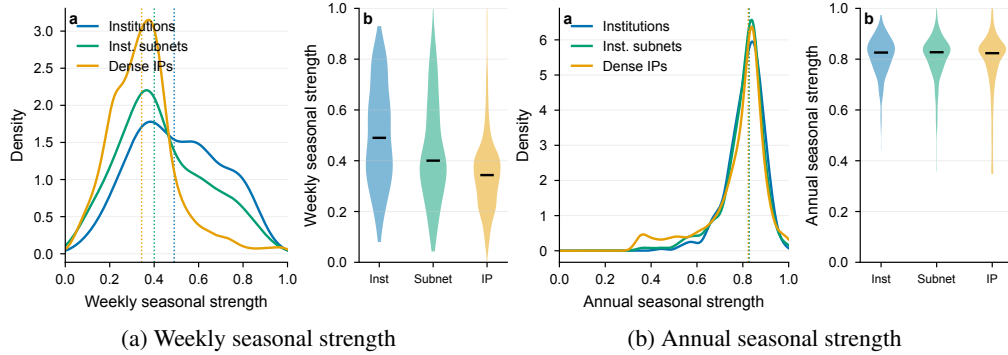


Figure 9: Distribution of Wang–Smith–Hyndman weekly and annual seasonal strengths across institutions, institutional subnets, and dense IP series (using metric `n_bytes`). Annual strength (median 0.83) exceeds weekly strength (medians 0.34–0.49) for 96.8 % of institutions, 98.2 % of institutional subnets, and 99.8 % of dense IP series.

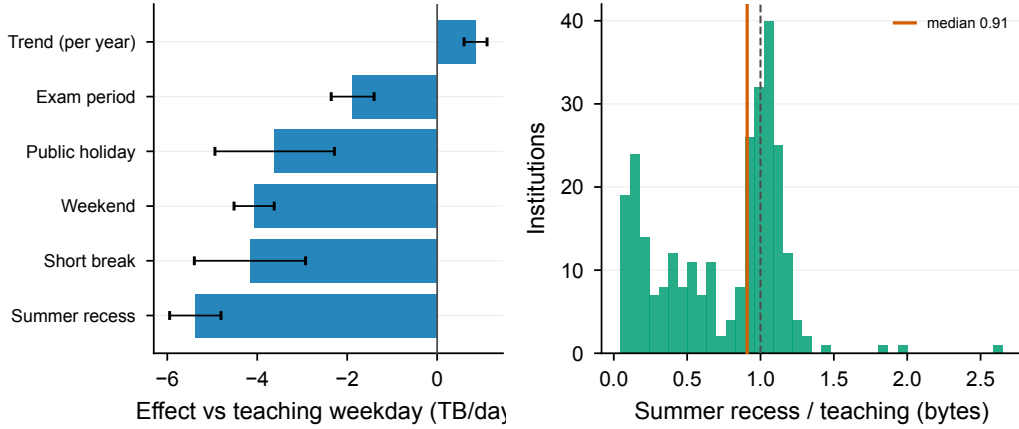
Second mode, gap severity is so heterogeneous across entities that any single random sample conflates easy and hard series. This observation motivates the stratified sample subsets validated at the end of this section.

Entity lifetime

A question that only a multi-year collection can answer is how long an individual entity (IP address) remains active in network traffic. The median lifetime of the 278,466 IP addresses is 972 days out of the 976.6 days that carry data (the 995-day window minus the outages), with an IQR of [924, 976]. Thus, 58.1 % of entities are observable for at least 99 % of the window and 72.8 % for at least 95 %.

The population decomposes into interpretable cohorts. A full-span cohort of 63.0 % entities is present from the first to the last day of the collection. A second cohort of 25.7 % entities first appears in late November 2023 during the mentioned Eduroam expansion and persists through the end. A third cohort of 7.0 % entities shows genuine arrivals and departures spread over time, and a short-lived tail of 2.4 % entities disappears within the first months of their lifetime. Two consequences follow.

First, the dataset is well-suited to long-history per-entity modeling because more than 2.5 years of history is available for most entities. Second, lifetimes must be read as *observability windows*, not as device histories, because an IP address may be reassigned to different physical devices within a single series. This can appear multiple times in a single day on heavily used Eduroam WiFi networks, which are also included in the dataset.



(a) Effect of each calendar condition on daily transmitted volume relative to a teaching weekday, with 95 % confidence intervals. The conditions order by campus occupancy rather than by whether teaching is formally in session. (b) Distribution of the summer-recess-to-teaching volume ratio computed separately for each institution. This analysis revealing two populations (universities and office-like networks) that the network aggregate blends into a single curve.

Figure 10: Calendar structure of the network load, measured directly by conditioning on the academic calendar.

Validation of long-term structure

This subsection verifies the presence of long-term multi-year behaviors, which are an annual seasonal component, a calendar structure tied to the academic year, and year-over-year drift. Each is reported as a measured property with its statistical support.

Annual seasonality is the dominant periodic component

We first ask which periodic component actually carries the seasonal variation. MSTL (Multiple Seasonal-Trend decomposition using Loess) splits a daily series into a trend, one seasonal component for each specified period, and a remainder. The Wang–Smith–Hyndman seasonal strength [43] then measures how much of the variance that remains after removing the trend is explained by a given seasonal component. It ranges from 0 (no seasonality at that period) to 1 (the component explains all remaining variance). Using an MSTL decomposition with weekly and annual periods and the Wang–Smith–Hyndman strength measure [43] on daily transmitted data (metric `n_bytes`), we compute both strengths for every entity with sufficient coverage. The distribution of Wang–Smith–Hyndman strength is shown in Figure 9.

The median annual strength is 0.83 at every entity level, whereas the median weekly strength ranges only from 0.34 (dense IP series) to 0.49 (institutions). Moreover, the annual component is stronger than the weekly one for 96.8 % of 280 institutions, 98.2 % of 544 institutional subnets, and 99.8 % of 5,000 randomly selected dense IP series (IP address with maximum 40 % of gaps). In other words, the strongest regular signal in academic-network traffic is not the familiar weekly rhythm on which sub-year benchmarks concentrate, but the annual one.

Week-to-week seasonal changes are caused by timetables changing between the halves of a term, by attendance declining as a term progresses, and by examination weeks and short breaks interrupting the weekly routine. By contrast, the annual cycle is fixed by the academic calendar, which places terms, examination periods, and vacations at nearly the same position every year. Therefore, the weekly component is less regular than the annual one, and the strength measure reflects that difference. The observed dominance of the annual component is thus consistent with how the network is used.

Load tracks campus occupancy, not the teaching calendar

To verify that the calendar events correspond to the behavior of the series, we regressed daily transmitted volume on the academic regime (teaching term, examination period, short break, summer recess), weekends, and public holidays. Figure 10a shows the shift in daily volume relative to a teaching weekday, with a fitted level of 16.74 TB/day. The trend term is a slope rather than a level shift and is reported separately: overall daily volume grows by 0.85 TB/day per year. The conditions fall into three tiers rather than a strict ordering. Summer recess removes the most traffic. Short breaks,

weekends, and public holidays form a middle group whose coefficients are mutually indistinguishable. Examination periods remove the least, and are separated from every other condition.

That grouping carries the substantive finding that the network does not respond to whether teaching is formally in session but to how many people are physically on campus. Examination periods keep buildings, laboratories, and dormitories occupied and retain roughly two-thirds of the volume that a summer recess loses, whereas a public holiday is not measurably different from an ordinary weekend. Two properties of the data follow. First, weekends and public holidays are statistically indistinguishable in their effect on volume, so the data do not support treating them as separate regimes. Second, the calendar conditions and the trend together explain $R^2 = 0.437$ of the daily variance. The remaining 56 % is the variation that the calendar does not account for, and it is the part of the signal on which anomaly detection operates.

The aggregate academic calendar effect is a property of a few large institutions, not of the network

Computing the summer-recess-to-teaching volume ratio separately for each of the 280 institutions with sufficient data yields a median of 0.909 and an IQR of [0.366, 1.052]. The distribution is distinctly bimodal, as shown in Figure 10b. The 32.9 % institutions fall below 0.5 of their teaching-period volume, 35.0 % sit at or above 1.0, and only 15.7 % occupy the intermediate range [0.5, 0.9) that the aggregate coefficient would suggest is typical. To test whether recess changes an institution's volume at all, we applied a two-sided Mann–Whitney U test to the daily volumes of recess days versus teaching days of each institution. The test asks whether the two sets of days come from the same distribution and makes no assumption about its shape. It finds the teaching-versus-recess contrast significant for 68.9 % institutions.

However, significance is not synonymous with a summer decline: 51.8 % of those institutions decrease, and 17.1 % significantly *increase* their traffic during recess. The largest confirmed increases are expected, given the institution's type. The two strongest are in the health sector, whose operations do not follow the academic calendar. The next is a national Internet infrastructure operator whose traffic is not driven by on-campus users. The remaining large increases come from the largest server-heavy address spaces in the dataset, where maintenance, migrations, and batch computation are scheduled when users are absent.

Transmitted volume is extremely concentrated because six institutions account for half of all traffic. Moreover, every one of the six largest falls in the seasonal mode, with recess ratios between 0.41 and 0.90. The two modes correspond to institution type rather than institution size, separating occupancy-driven entities such as teaching faculties and dormitories from office networks and machine-driven entities such as research clusters and hosting infrastructure, whose load is indifferent to the academic calendar. This is the clearest argument in our validation for publishing per-entity series rather than aggregates alone. An aggregate seasonal component averages two populations into a single curve that describes neither, and evaluating a seasonal model only on the network total would report a level of calendar predictability that does not transfer to the entities users actually deploy models on.

The annual component repeats across cycles

Dominance of the annual component does not by itself show that it repeats. Therefore, we tested repetition on two deliberately contrasting institutions: institution 74, a university-like network, and institution 101, an office-like network. Correlating the daily profile of the first annual cycle with that of the second while shifting the phase, the university's profile peaks sharply at a +2-day shift with Pearson $r = 0.825$, whereas the office's correlation has no seasonal peak. Fitting the calendar effects independently in each cycle, on the decimal logarithm of daily transmitted bytes, gives a vacation effect of -1.01 , -0.88 , and -0.80 log units for the university and a public-holiday effect of -1.52 , -1.53 , and -1.36 log units for the office. Therefore, both institutions carry repeatable annual structure, but of different kinds, and the repetition is a measurable property rather than a visual impression.

Year-over-year behavioral drift

Seasonality describes what repeats. Drift describes what does not. To verify that the dataset contains measurable drift, we compared the per-entity distribution of each metric in the first collection year with that in the second, using the two-sample Kolmogorov–Smirnov (KS) test over the 3,541 entities sufficiently observed in both years. The KS test asks whether two samples come from the same distribution; its statistic is the largest difference between the two empirical distribution functions. Significant drift ($p < 0.05$) is widespread across the dataset and affects 82.6 % of entities for `n_flows` (median KS statistic 0.25), 73.8 % for `tcp_udp_ratio_bytes`, 71.5 % for `n_bytes`, and 63.8 % for `avg_ttl`, as shown in Figure 11. The drift has a direction. Within this cohort, the mean daily flow count fell by roughly a quarter and the mean daily byte volume by 13 % from year one to year two. For most entities, next year's data is therefore measurably different from this year's.

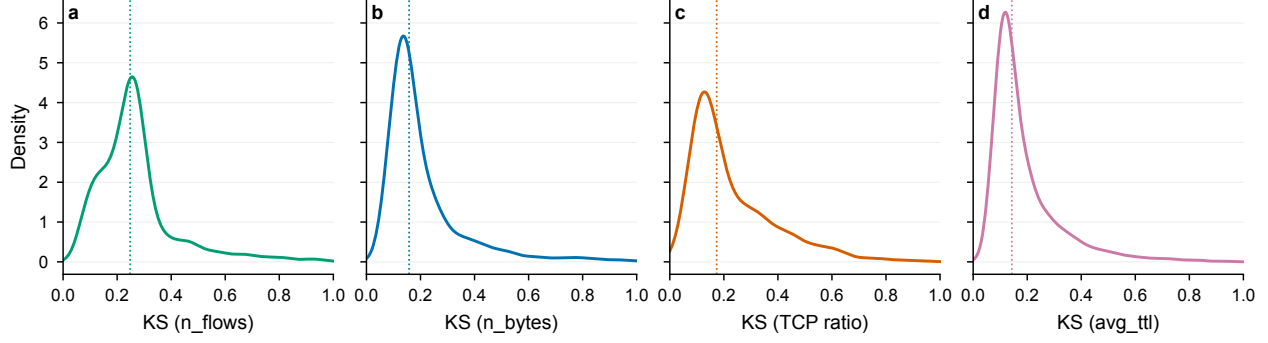


Figure 11: Distribution across entities of the year-1-versus-year-2 KS statistic for four metrics. Significant distributional drift is the norm rather than the exception (63.8–82.6 % of entities depending on the metric).

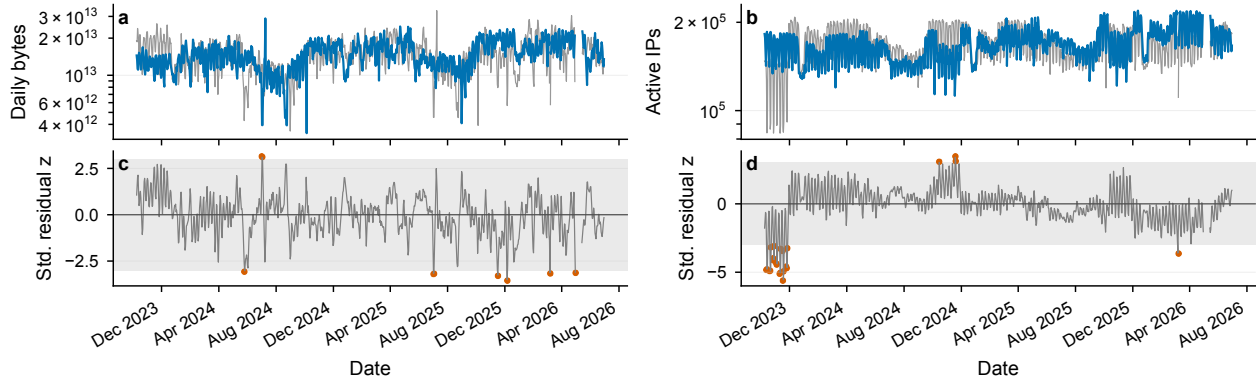


Figure 12: Leave-one-cycle-out annual baseline. Recurring calendar effects (Christmas, summer) are absorbed (median $|z| \approx 0.5\text{--}1.2$), while non-recurring events remain detectable (2024 probe outage $|z| \approx 3.0$ in byte volume. November 2023 Eduroam step $|z| \approx 5.6$ in the active-IP count).

Validation of separation of annual events and anomalies

Calendar structure poses a specific problem for anomaly detection. A Christmas traffic collapse is a large deviation from the preceding weeks. A detector trained on less than a year of data must either raise an alarm on it or be told manually to ignore it. Repeated annual cycles resolve this problem in a principled way. We formed a leave-one-cycle-out annual baseline, in which the expectation for each cycle is built only from the other cycles, and scored standardized robust residuals against it, as shown in Figure 12.

The separation works in both directions. The baseline absorbs the recurring calendar effects. The Christmas and summer periods, which would dominate the alarm list of any short-window detector, collapse to median residuals of $|z| \approx 0.5\text{--}1.2$. Non-recurring events remain prominent. The 2024 partial probe outage reaches $|z| \approx 3.0$ and the November 2023 Eduroam architecture step $|z| \approx 5.6$. The two events surface in different metrics. The outage is most visible in byte volume, whereas the Eduroam step is most pronounced in active-IP count. Therefore, the dataset supports the evaluation of seasonally aware anomaly detectors with a built-in ground-truth contrast: two labeled, non-recurring events that a detector must flag, and a set of recurring calendar effects that it must not flag.

Validation of the longitudinal subnet

The 4.5-year longitudinal subnet extends the analyses above beyond the reach of the main collection, as shown in Figure 13. One feature of the figure needs a note before the findings. Between April and July 2023, the summed flow count increases by about one order of magnitude, then returns to its previous level. This is not a measurement artifact. A single address, IP identifier 1590672, accounts for 99.5 % of the excess flows in that period, and the five most active addresses together account for 99.8 %. That address is present in the dataset only from 2023-04-09 to 2023-07-17, so it appears and disappears together with the excess. Therefore, the episode is a genuine short-lived entity, visible in the per-address series, and it is one example of why the subnet aggregate must be read together with its member series. We keep it in the data and in the figure as a labeled event.

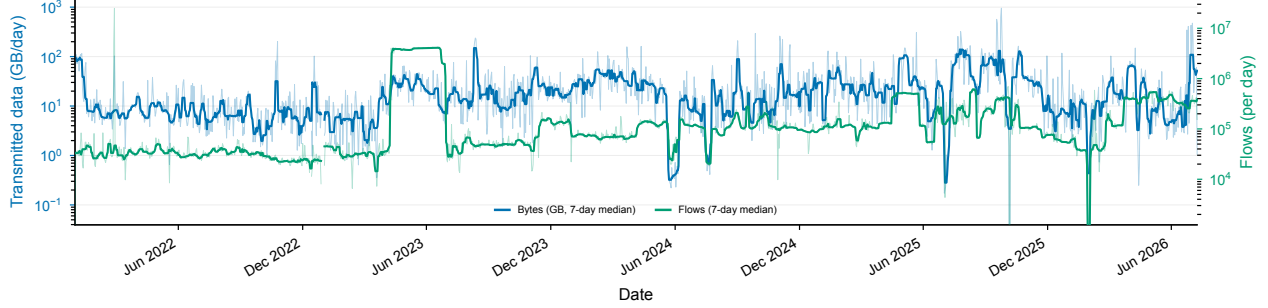


Figure 13: Daily total transmitted data and flow count of the longitudinal subnet over its 4.5-year span (7-day medians). The elevated flow count between April and July 2023 is driven by a single short-lived address (IP identifier 1590672), which appears in the dataset only from 2023-04-09 to 2023-07-17 and accounts for 99.5 % of the excess flows during that period.

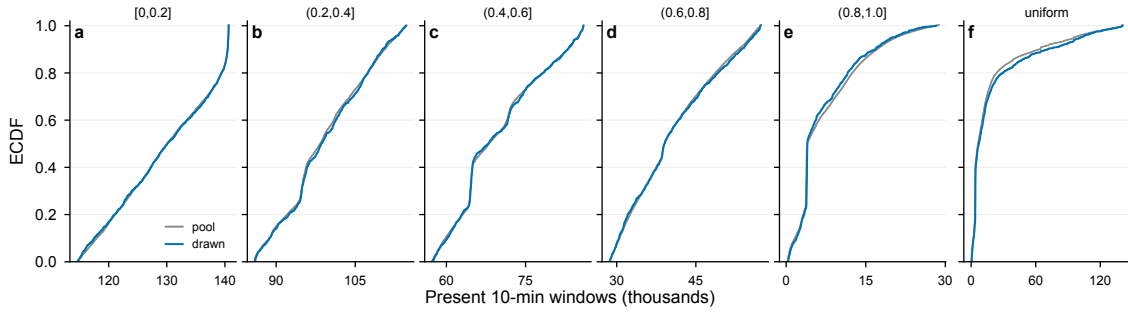


Figure 14: Per-band comparison of the n_{present} distribution of the drawn 1,000 entities against the full band candidate pool. Distributions coincide across all bands (KS $p = 0.12\text{--}0.93$), confirming that each subset is a representative draw from its band.

Three findings follow from the subnet’s length. First, its traffic grows strongly. Summed over the 135 series, byte volume rises by roughly +277 % and flow counts by +775 % over the 4.5 years. However, it is caused by two IP addresses from all of them. Without the series with ID 79658, it is +88 %, and also without the series with ID 75864, it is +26 % during the whole period.

Second, the two volumetric metrics behave differently across cycles. The summed byte profile does not reproduce annually (mean cross-cycle Pearson $r \approx -0.1$) because the byte volume of a small set of entities is heavy-tailed, and a few large transfers can dominate any given week. Flow counts track the number of connections rather than their size, and they exhibit a weak, recurring academic rhythm.

Third, the four complete cycles allow an evaluation that, to our knowledge, no other public network dataset supports: training on three full cycles and forecasting the entire held-out fourth year. On flows, a weekly-plus-annual model beats a weekly-only model over the full-year horizon (SMAPE 53.2 versus 60.7, $R^2 +0.15$ versus -0.05). On bytes, the annual component brings nothing (SMAPE 81.6 versus 81.5, $R^2 \leq 0$). At this small aggregation scale, annual structure is therefore usable for count metrics but not for volume metrics.

Validation of the stratified sample subsets

The full IP address part contains more than 278 thousand series. Evaluating a method on all of them is impractical, and a single random sample reports only one average difficulty. Therefore, we publish six fixed-size ($N = 1000$) sample subsets: a uniform-random baseline that reproduces the original CESNET-TimeSeries24 sampling, and five subsets stratified by percentage of gaps into equidistant windows. We validate the subsets for stratification integrity and representativeness, so that authors can select a subset with confidence in its contents.

Every subset is exactly sized (1,000 drawn from pools that all exceed 1,000) with zero out-of-band entities. Deciding membership in integer n_{present} space removes float ambiguity entirely. Within each band, the drawn 1,000 are statistically indistinguishable from the full band pool in terms of the number of present datapoints, as shown in Figure 14. The uniform baseline is indistinguishable from the entire 278,466-entity population. Therefore, the baseline

Table 3: Relationship of the CESNET-TimeSeries26 IP-address entity set to CESNET-TimeSeries24, under the recomputed relative filtering threshold ($> 99.75\%$ empty, i.e., ≤ 358 non-empty 10-minute datapoints).

Entity set	Institutions	Institution subnets	IP addresses
CESNET-TimeSeries24 entities	283	548	275,124
CESNET-TimeSeries26 entities	285	566	278,466
Shared (qualify in both)	283	548	253,543
Added (qualify now, absent before)	2	18	24,923
Dropped (in TS24, no longer qualify)	0	0	21,581

can stand in for the dataset as a whole. The five bands are stratified by design and must *not* be concatenated into a pseudo-representative pool, because doing so would weight sparse series at one two-hundredth of their true prevalence. The candidate pool sizes themselves are a finding about the population. The very-sparse band draws from 228,387 entities (82.0 % of the dataset), whereas the dense $[0.00, 0.20]$ band draws from only 7,852 (2.8 %), and the three densest bands each hold fewer than 13,000 candidates. Dense, always-on series, the kind on which forecasting papers prefer to demonstrate, are a small minority of real ISP-scale data.

Relationship to CESNET-TimeSeries24

CESNET-TimeSeries26 is the successor of CESNET-TimeSeries24 [16], not an extension of it. This section explains how the two datasets relate and what the changes are over the multi-year span. We use CESNET-TimeSeries24 as the controlled instance of a sub-year collection because it comes from the same network, monitoring pipeline, and metrics, and differs only in length. The conclusions apply to sub-year network traces in general.

Continuity The capture pipeline, metrics, aggregation, and anonymization are unchanged, and the anonymized identifiers denote the same entities in both datasets. Table 3 shows overlaps between entities in both datasets. Of the 275,124 IP-address series of CESNET-TimeSeries24, 253,543 remain in the new dataset under the recomputed relative completeness threshold, 21,581 no longer qualify, and 24,923 series that became active during the extension are new. Furthermore, all 283 institutions and 548 institutional subnets are retained, with 2 and 18 added. Therefore, results, labels, and identified events published for the 40-week dataset remain valid starting points for this one. Moreover, every series in CESNET-TimeSeries26 whose ID also appears in CESNET-TimeSeries24 is a strict extension of it, i.e., its first 40 weeks are identical to the corresponding series in CESNET-TimeSeries24..

The dominant periodic component is unobservable below one cycle The annual component carries more of the periodic variation than the weekly one for 96.8 % of institutions, 98.2 % of institutional subnets, and 99.8 % of dense IP series, as shown in Figure 9. A record shorter than one year cannot contain this component. A test on the university-like institution 74 places the break-even point for an annual term at exactly one complete cycle, as shown in Figure 15. Below that length, the annual term is harmful because a sub-year window observes only a fragment of the annual curve, thus, the seasonal and trend components cannot be separated from each other.

A harmonic model trained on a 40-week window, the span of CESNET-TimeSeries24, reaches a median MASE of 1.37. The same model class trained on two full cycles with an annual term reaches 0.87. Two cycles of history *without* the annual term yield 1.39, similar as for the 40-week window. Therefore, a longer history is not valuable in itself. It becomes valuable when the model can express the annual component. Across all 255 institutions with sufficient coverage, the gain from the annual term grows with the depth of the institution’s summer drop (Spearman $\rho = -0.571$, $p = 2 \cdot 10^{-23}$). This is expected observation because only a model with an annual term can forecast correctly the behavior change from the teaching period to the summer break. The majority of institutions are universities and schools for which is change of behavior at the start of summer holidays significant behavior.

Growth and seasonality separate only over multiple years The fitted trend of +0.85 TB/day per year (+5.1 %) is small against a summer-recess drop of 5.38 TB/day. Over the 40-week window of CESNET-TimeSeries24, the trend accumulates to roughly 0.65 TB/day, which is less than the uncertainty in the seasonal coefficients. Over 2.7 years, it accumulates to about 2.3 TB/day and separates cleanly. A sub-year record cannot distinguish secular growth from the rising limb of a seasonal cycle, so any capacity-planning or long-horizon forecasting study built on it extrapolates a quantity it cannot identify.

Calendar effects cannot be separated from anomalies without a second cycle The leave-one-cycle-out baseline of Figure 12 needs at least two complete cycles, because the expectation for each cycle must come from other cycles. On a sub-year record, a detector has no annual expectation to violate; it must either alarm on every Christmas and every

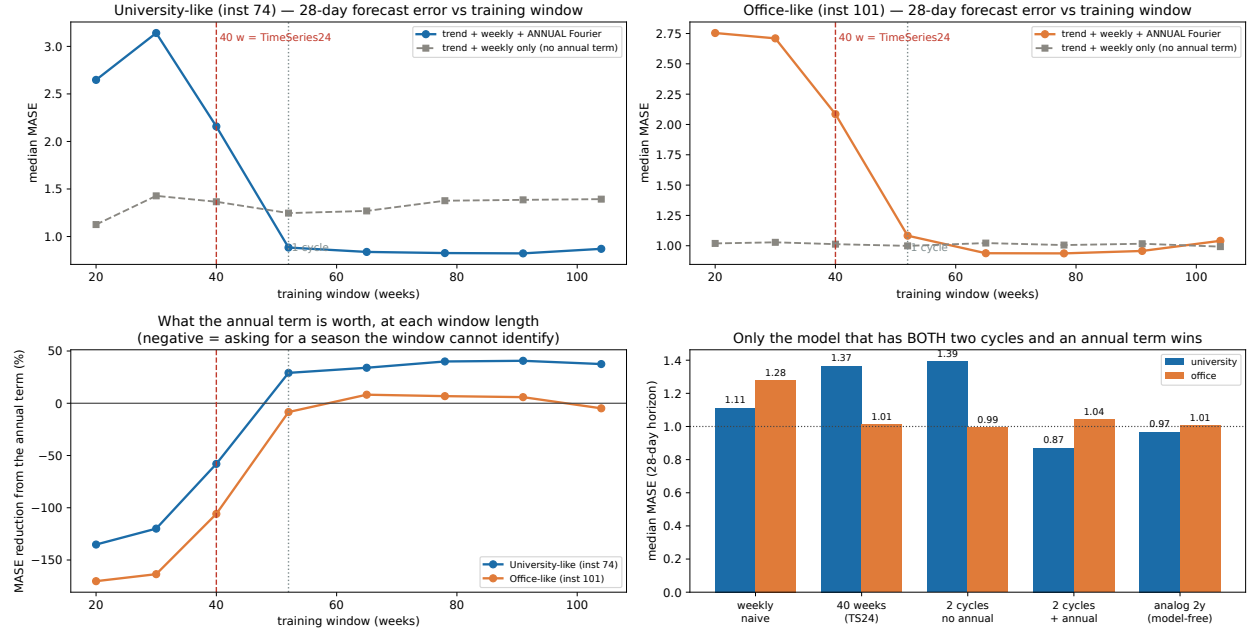


Figure 15: Exploitability of the annual component in a rolling-origin backtest with a 28-day horizon. Top: median MASE as a function of training-window length, with and without the annual term. The annual model starts to win only once the window covers one full cycle. Bottom left: the MASE reduction attributable to the annual term for each window length. Below one cycle, the contribution is negative. Bottom right: only the model that includes both historical cycles and an annual term beats all baselines for the university. The office is served equally well by any weekly model.

summer recess or be told manually to ignore them. This class of evaluation, seasonally aware anomaly detection with a built-in contrast between recurring and non-recurring events, is therefore unavailable on any sub-year dataset.

Model staleness is measurable only over multiple years The drift verified above has an operational consequence: a deployed model goes stale. To measure how fast, we trained a fixed seasonal model once on the first annual cycle and evaluated it, without retraining, on consecutive 30-day blocks over the following 20 months, on 40 dense series (20 institutions and 20 IP addresses). We compared it against the same model class retrained every 30 days on a trailing one-cycle window. The stale model’s median SMAPE grows from 25.5 % to 123.9 %, approximately 11.3 SMAPE points per quarter, whereas the retrained model stays near 29.5 % (Figure 16). After about 2.3 months the stale error exceeds $1.5\times$ the fresh baseline, which implies a practical retraining interval of roughly two to three months for this model class. Measuring this decay requires a complete training cycle followed by a long evaluation period. A sub-year record cannot even complete the training cycle, so the staleness of a deployed model is, to our knowledge, measurable on public network data for the first time.

Usage Notes

This section describes recommendations for evaluating network traffic forecasting (and forecasting-based anomaly detection) on this dataset to ensure that results from different authors remain comparable. The checklist is shown in Table 4.

Dataset selection The dataset is divided into several parts, each of which is independently usable: Full IP addresses, Sample IP addresses, Sample IP addresses with exact gap ratio, Institutions, Institutional subnets, and Longitudinal subnet. Authors must state which part(s) (1) and which aggregation interval(s) (2) they use, reporting results separately for each. The approach must indicate whether it is multivariate or univariate (3), and any unused metrics must be stated (4). All preprocessing must be described (5). Given that the Full IP address part contains more than 278 thousand series, evaluating all of them is challenging. Therefore, we also publish a stratified sample subset and a stratified sample subsets with predefined gap ratios.

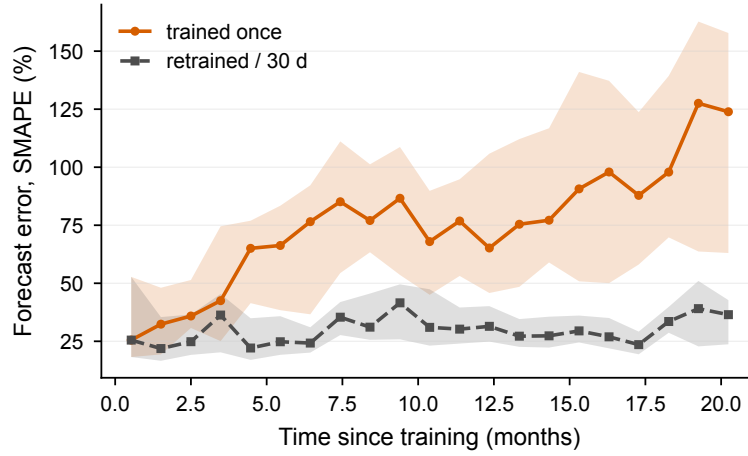


Figure 16: Forecast error versus time since training. Without retraining, median SMAPE grows by ~ 11.3 points per quarter ($25.5\% \rightarrow 123.9\%$ over 20 months); retraining every 30 days holds error near 29.5% , implying a practical retraining interval of roughly two to three months.

Table 4: Checklist for addressing recommended evaluation procedures.

No.	Recommendation
(1)	Specify which dataset part(s) are used in the analysis.
(2)	Specify the aggregation interval(s) used.
(3)	Indicate whether the approach is multivariate or univariate.
(4)	Clearly state if not all metrics are used.
(5)	Document all preprocessing steps, including filtering, normalization, and gap handling.
(6)	Ensure the training phase starts from the beginning of the relevant part’s time frame (main dataset: 2023-10-09; longitudinal subnet: its own 2022 start).
(7)	Specify the duration of the training window.
(8)	Define and describe the validation window if employed.
(9)	Clearly describe the retraining process if the model is retrained during evaluation.
(10)	Specify the forecasting horizon.
(11)	Clearly specify the evaluation metrics used.
(12)	Provide an overall comparison across series using statistical distributions and aggregate statistics.
(13)	Assess and document computational requirements and deployability.
(14)	Make source code of experiments and models publicly available.
(15)	Report results separately per evaluation year (or clearly delimited sub-periods) in addition to overall results, so the effect of behavioral drift is visible.
(16)	When claiming annual-seasonality modeling, ensure the training window covers at least one complete annual cycle and state this explicitly.
(17)	Prefer the canonical split for comparability: train 2023-10-09 to 2025-10-08 (two full years), test the remainder.

Longitudinal subnet caveats Three caveats apply to the longitudinal subnet part. It is TCP-only throughout, so its absolute volumetric levels must not be compared directly with the all-protocol parts, and it must not be used in analyses that depend on UDP traffic. Its transport-ratio metrics are constant and should be excluded from the model. Recommendation (6) applies with the part’s own 2022 start date rather than 2023-10-09.

Training and forecasting correctness The training phase must start at the beginning of the relevant part’s time frame (6), with the training-window duration stated (7); a validation window, if used, must be defined (8); retraining must be described (9); and the forecasting horizon must be specified (10).

Evaluation metrics The chosen metrics must be specified (11). We recommend the following (n observations, y_i actual, $\hat{y}_i$ predicted):

- **Root Mean Squared Error (RMSE)**, sensitive to large errors [44]:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (1)$$

For combining multiple RMSEs we recommend the weighted RMSE:

$$\text{Weighted RMSE} = \sqrt{\frac{\sum_{i=1}^n \sigma_i^2 \cdot \text{RMSE}_i^2}{\sum_{i=1}^n \sigma_i^2}} \quad (2)$$

where σ_i^2 is the variance of the true values in the i -th series.

- **Symmetric Mean Absolute Percentage Error (SMAPE)** [44], with ϵ a small constant:

$$\text{SMAPE} = \frac{100\%}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i| + \epsilon)/2} \quad (3)$$

with weighted variant

$$\text{Weighted SMAPE} = \frac{\sum_{i=1}^n \sigma_i^2 \cdot \text{SMAPE}_i}{\sum_{i=1}^n \sigma_i^2} \quad (4)$$

- **Coefficient of Determination (R^2)** [45, 44]:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \mu_y)^2} \quad (5)$$

with weighted variant

$$\text{Weighted } R^2 = \frac{\sum_{i=1}^n \sigma_i^2 \cdot R_i^2}{\sum_{i=1}^n \sigma_i^2} \quad (6)$$

Multiple time series evaluation All metrics are computed per series; an overall comparison across series must additionally be made (12) using distributions per metric (mean, standard deviation, the weighted variants, and histogram/KDE plots).

Computational requirements and deployability Evaluation should assess not only accuracy but also training time, inference speed, and resource consumption (13); excessive requirements may preclude real-time or resource-limited deployment. The source code of experiments and models should be made publicly available (14).

Per-year reporting and annual-cycle claims Because the dataset spans multiple years and the behavior of most entities drifts from year to year, an overall metric can mask a model that degrades over time. Results should therefore be reported per evaluation year or per clearly delimited sub-period (15). Any approach claiming to model annual seasonality must use a training window that covers at least one complete annual cycle and state this explicitly (16). Below that length, the annual term degrades the forecast. For cross-study comparability, we recommend the canonical split (17): train on the first two full years (2023-10-09 to 2025-10-08) and test on the remainder.

Table 5: Software used for creating the dataset.

Name	Version	Link
Ipfixprobe	4.11.0	https://github.com/CESNET/ipfixprobe
IPFIXcol2	2.2.1	https://github.com/CESNET/ipfixcol2
NEMEA Framework	0.14.0	https://github.com/CESNET/Nemea-Framework
NEMEA modules	2.20.0	https://github.com/CESNET/Nemea-Modules
NEMEA Supervisor	1.8.2	https://github.com/CESNET/Nemea-Supervisor
TimeScaleDB-14	2.15.0	https://www.timescale.com/
Python	3.9.0	https://www.python.org/downloads/release/python-390/
Create Datapoints module	–	https://github.com/koumajos/CESNET-TimeSeries24-CD

Code Availability

The dataset has been produced using open-source software: the flow exporter Ipfixprobe, the flow collector IPFIXcol2, the NEMEA processing system and modules (all on GitHub), and the TimeScaleDB database. We also provide the Create Datapoint module and deployment scripts for the NEMEA Supervisor and for building the database. Software versions and repository links are summarized in Table 5.

Integration into CESNET-TSZoo To simplify adoption, CESNET-TimeSeries26 is also integrated into the CESNET-TSZoo library [42]: <https://github.com/CESNET/cesnet-tszoo>. This library already provides access to CESNET-TimeSeries24 dataset and others network traffic time series datasets. Therefore, researchers can also download the dataset using the library, select the aggregation level and the source type (IP addresses, institutions, or institutional subnets), and create train, validation, and test splits through the same interface they already use for CESNET-TimeSeries24 dataset. Because the library handles data loading, filtering, and preprocessing in a consistent way, experiments can move between the two datasets with minimal code changes.

Acknowledgements

This research was supported by the Ministry of Education, Youth and Sports of the Czech Republic in the project “e-Infrastructure CZ” (LM2023054). Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

Author contributions statement

J.K. and K.H. proposed the time series metrics; J.K. designed and implemented the architecture for collecting time series metrics and maintained the collection pipeline over the extended multi-year period; K.H. and P.Š. designed and implemented the architecture for capturing network traffic; K.J. maintained the flow-monitoring infrastructure and prepared the archived longitudinal-subnet data for re-aggregation; J.K. and K.H. preprocessed the data into the final version of the dataset; J.K. performed the long-term technical-validation analyses (annual seasonality, reproducibility, drift, model longevity, and calendar-anomaly separation); K.H. and K.J. handled supervision; and all authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

References

- [1] Alessandro D’Alconzo, Idilio Drago, Andrea Morichetta, Marco Mellia, and Pedro Casas. A survey on big data for network traffic monitoring and analysis. *IEEE Transactions on Network and Service Management*, 16(3):800–813, 2019.
- [2] Giuseppe Aceto, Domenico Ciuonzo, Antonio Montieri, and Antonio Pescapé. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. *IEEE transactions on network and service management*, 16(2):445–458, 2019.

- [3] Josef Koumar, Karel Hynek, Jaroslav Pešek, and Tomáš Čejka. Nettisa: Extended ip flow with time-series features for universal bandwidth-constrained high-speed network traffic classification. *Computer Networks*, 240:110147, 2024.
- [4] Iman Akbari, Mohammad A Salahuddin, Leni Ven, Noura Limam, Raouf Boutaba, Bertrand Mathieu, Stephanie Moteau, and Stephane Tuffin. Traffic classification in an increasingly encrypted web. *Communications of the ACM*, 65(10):75–83, 2022.
- [5] Richard Plný, Karel Hynek, and Tomáš Čejka. Decrypto: Finding cryptocurrency miners on isp networks. In *Nordic Conference on Secure IT Systems*, pages 139–158. Springer, 2022.
- [6] Kamil Jerabek, Jan Luxemburk, Richard Plný, Josef Koumar, Jaroslav Pesek, and Karel Hynek. When simple model just works: Is network traffic classification in crisis? *arXiv preprint arXiv:2506.08655*, 2025.
- [7] Iwan Syarif, Adam Prugel-Bennett, and Gary Wills. Unsupervised clustering approach for network anomaly detection. In Rachid Benlamri, editor, *Networked Digital Technologies*, pages 135–145, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [8] Ren-Hung Hwang, Min-Chun Peng, Chien-Wei Huang, Po-Ching Lin, and Van-Linh Nguyen. An unsupervised deep learning model for early network traffic anomaly detection. *IEEE Access*, 8:30387–30399, 2020.
- [9] Asrul H Yaacob, Ian KT Tan, Su Fong Chien, and Hon Khi Tan. Arima based network anomaly detection. In *2010 Second International Conference on Communication Software and Networks*, pages 205–209. IEEE, 2010.
- [10] Tomasz Andrysiak, Łukasz Saganowski, Michał Choraś, and Rafał Kozik. Network traffic prediction and anomaly detection based on arfima model. In *International Joint Conference SOCO'14-CISIS'14-ICEUTE'14: Bilbao, Spain, June 25th-27th, 2014, Proceedings*, pages 545–554. Springer, 2014.
- [11] Kamil Jerabek, Josef Koumar, Jiří Setinský, and Jaroslav Pesek. Explainable anomaly detection in network traffic using llm. In *NOMS 2025-2025 IEEE Network Operations and Management Symposium*, pages 1–6. IEEE, 2025.
- [12] Gabriel O Ferreira, Chiara Ravazzi, Fabrizio Dabbene, Giuseppe C Calafiore, and Marco Fiore. Forecasting network traffic: A survey and tutorial with open-source comparative evaluation. *IEEE Access*, 11:6018–6044, 2023.
- [13] Renjie Wu and Eamonn J Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE transactions on knowledge and data engineering*, 35(3):2421–2429, 2021.
- [14] Kenjiro Cho, Koushirou Mitsuya, and Akira Kato. Traffic data repository at the {WIDE} project. In *2000 USENIX Annual Technical Conference (USENIX ATC 00)*, 2000.
- [15] Romain Fontugne, Pierre Borgnat, Patrice Abry, and Kensuke Fukuda. MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking. In *ACM CoNEXT '10*, Philadelphia, PA, December 2010.
- [16] Josef Koumar, Karel Hynek, Tomáš Čejka, and Pavel Šiška. Cesnet-timeseries24: Time series dataset for network traffic anomaly detection and forecasting. *Scientific Data*, 12(1):338, 2025.
- [17] Ching Chang, Jeehyun Hwang, Yidan Shi, Haixin Wang, Wei Wang, Wen-Chih Peng, and Tien-Fu Chen. Time-imm: A dataset and benchmark for irregular multimodal multivariate time series. *Advances in Neural Information Processing Systems*, 38, 2026.
- [18] Fan Liu, Behrooz Farkiani, and Patrick Crowley. Time-series foundation models for isp traffic forecasting. *arXiv preprint arXiv:2511.17529*, 2025.
- [19] Josef Koumar, Timotej Smoleň, Kamil Jeřábek, and Tomáš Čejka. Comparative analysis of deep learning models for real-world isp network traffic forecasting. *IEEE Transactions on Network and Service Management*, 23:715–728, 2025.
- [20] Wenjing Feng, Lanhao Li, Huichao Zhou, Bingshu Xie, Dawei Zhao, Yi Zhang, and Feifei Wang. Robust long-term spatial-temporal forecasting for dynamic networks: The mdetst model. *Mathematics*, 14(4):661, 2026.
- [21] John Sengendo and Fabrizio Granelli. From prediction to protection: A network digital twin framework for traffic monitoring and anomaly detection. *Computer Networks*, page 112250, 2026.
- [22] Chris Misa and Ramakrishnan Durairajan. Sustainability or survivability? eliminating the need to choose in leo satellite constellations. In *Proceedings of the 24th ACM Workshop on Hot Topics in Networks*, pages 123–130, 2025.
- [23] Zifan Zhang, Zhichao Hou, Tingxiang Ji, and Yuchen Liu. Centile: A telemetry foundation model evaluated by the decisions it drives. *arXiv preprint arXiv:2608.01725*, 2026.

- [24] Eilaf Babai, Aalaa Babai, and Koji Okamura. Which deep learner? a systematic evaluation of advanced deep forecasting models accuracy and efficiency for network traffic prediction. *arXiv preprint arXiv:2601.02694*, 2026.
- [25] Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. Largest: A benchmark dataset for large-scale traffic forecasting. *Advances in neural information processing systems*, 36:75354–75371, 2023.
- [26] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *International journal of forecasting*, 38(4):1346–1364, 2022.
- [27] Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*, 2021.
- [28] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.
- [29] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.
- [30] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.
- [31] Duc-Huy Le, Hai-Anh Tran, Sami Souihi, and Abdelhamid Mellouk. An ai-based traffic matrix prediction solution for software-defined network. In *ICC 2021 - IEEE International Conference on Communications*, pages 1–6, 2021.
- [32] Pablo Fondo-Ferreiro, Miguel Rodelgo-Lacruz, Francisco Javier González-Castaño, Felipe Gil-Castiñeira, and David Candal-Ventureira. Network operator kpis time series dataset, July 2023.
- [33] Gianni Barlacchi, Marco De Nadai, Roberto Larcher, Antonio Casella, Cristiana Chitic, Giovanni Torrisi, Fabrizio Antonelli, Alessandro Vespignani, Alex Pentland, and Bruno Lepri. A multi-source dataset of urban life in the city of milan and the province of trentino. *Scientific data*, 2(1):150055, 2015.
- [34] Tomas Benes, Jaroslav Pesek, and Tomas Cejka. Look at my network: An insight into the isp backbone traffic. In *2023 19th International Conference on Network and Service Management (CNSM)*, pages 1–7. IEEE, 2023.
- [35] Steve Uhlig, Bruno Quoitin, Jean Lepropre, and Simon Balon. Providing public intradomain traffic matrices to the research community. *SIGCOMM Comput. Commun. Rev.*, 36(1):83–86, January 2006.
- [36] Yin Zhang, Matthew Roughan, Nick Duffield, and Albert Greenberg. Fast accurate computation of large-scale ip traffic matrices from link loads. In *Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '03, page 206–217, New York, NY, USA, 2003. Association for Computing Machinery.
- [37] Sebastian Orłowski, Roland Wessäly, Michal Pióro, and Artur Tomaszewski. Sndlib 1.0—survivable network design library. *Networks: An International Journal*, 55(3):276–286, 2010.
- [38] Josef Koumar, Karel Hynek, Kamil Jeřábek, and Pavel Šiška. Cesnet-timeseries26: A multi-year network traffic time series dataset for annual seasonality, concept drift, and long-term anomaly detection, September 2026.
- [39] Paul Aitken, Benoît Claise, and Brian Trammell. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. RFC 7011, September 2013.
- [40] Stuart Staniford, James A Hoagland, and Joseph M McAlerney. Practical automated detection of stealthy portscans. *Journal of Computer Security*, 10(1-2):105–136, 2002.
- [41] Monowar H Bhuyan, Dhruba Kr Bhattacharyya, and Jugal K Kalita. Surveying port scans and their detection methodologies. *The Computer Journal*, 54(10):1565–1581, 2011.
- [42] Milan Kureš, Josef Koumar, and Karel Hynek. Cesnet ts-zoo: A library for reproducible analysis of network traffic time series. In *2025 21st International Conference on Network and Service Management (CNSM)*, pages 1–5. IEEE, 2025.
- [43] Xiaozhe Wang, Kate Smith, and Rob Hyndman. Characteristic-based clustering for time series data: Wang, smith and hyndman. *Data mining and knowledge Discovery*, 13(3):335–364, 2006.
- [44] Davide Chicco, Matthijs J Warrens, and Giuseppe Jurman. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *Peerj computer science*, 7:e623, 2021.
- [45] Sewall Wright. Correlation and causation. *Journal of agricultural research*, 20(7):557, 1921.

Table 6: Detailed descriptions of time series metrics for each IP address dataset. Note: in the longitudinal subnet part, all volumetric metrics reflect TCP traffic only, and `tcp_udp_ratio_packets/tcp_udp_ratio_bytes` are constant (= 1); see the Longitudinal subnet collection subsection.

Column name	Description
<code>id_time</code>	Unique identifier for each aggregation interval within the time series, used to segment the dataset into specific time periods for analysis.
<code>n_flows</code>	Total number of flows observed in the aggregation interval, indicating the volume of distinct sessions or connections for the IP address.
<code>n_packets</code>	Total number of packets transmitted during the aggregation interval, reflecting the packet-level traffic volume for the IP address.
<code>n_bytes</code>	Total number of bytes transmitted during the aggregation interval, representing the data volume for the IP address.
<code>n_dest_ip</code>	Number of unique destination IP addresses contacted by the IP address during the aggregation interval, showing the diversity of endpoints reached.
<code>n_dest_asn</code>	Number of unique destination Autonomous System Numbers (ASNs) contacted by the IP address during the aggregation interval, indicating the diversity of networks reached.
<code>n_dest_port</code>	Number of unique destination transport layer ports contacted by the IP address during the aggregation interval, representing the variety of services accessed.
<code>tcp_udp_ratio_packets</code>	Ratio of packets sent using TCP versus UDP by the IP address during the aggregation interval, providing insight into the transport protocol usage pattern. This metric belongs to the interval $\langle 0, 1 \rangle$ where 1 is when all packets are sent over TCP, and 0 is when all packets are sent over UDP.
<code>tcp_udp_ratio_bytes</code>	Ratio of bytes sent using TCP versus UDP by the IP address during the aggregation interval, highlighting the data volume distribution between protocols. This metric belongs to the interval $\langle 0, 1 \rangle$ with the same rule as <code>tcp_udp_ratio_packets</code> .
<code>dir_ratio_packets</code>	Ratio of packet directions (inbound versus outbound) for the IP address during the aggregation interval, indicating the balance of traffic flow directions. This metric belongs to the interval $\langle 0, 1 \rangle$, where 1 is when all packets are sent in the outgoing direction from the monitored IP address, and 0 is when all packets are sent in the incoming direction to the monitored IP address.
<code>dir_ratio_bytes</code>	Ratio of byte directions (inbound versus outbound) for the IP address during the aggregation interval, showing the data volume distribution in traffic flows. This metric belongs to the interval $\langle 0, 1 \rangle$ with the same rule as <code>dir_ratio_packets</code> .
<code>avg_duration</code>	Average duration of IP flows for the IP address during the aggregation interval, measuring the typical session length.
<code>avg_ttl</code>	Average Time To Live (TTL) of IP flows for the IP address during the aggregation interval, providing insight into the lifespan of packets.

Table 7: Time series metrics which replace metrics `n_dest_ip`, `n_dest_asn` and `n_dest_port` in the 1-hour and 1-day aggregations.

Column name	Description
<code>sum_n_dest_ip</code>	Sum of numbers of unique destination IP addresses.
<code>avg_n_dest_ip</code>	The average number of unique destination IP addresses.
<code>std_n_dest_ip</code>	Standard deviation of numbers of unique destination IP addresses.
<code>sum_n_dest_asn</code>	Sum of numbers of unique destination ASNs.
<code>avg_n_dest_asn</code>	The average number of unique destination ASNs.
<code>std_n_dest_asn</code>	Standard deviation of numbers of unique destination ASNs.
<code>sum_n_dest_port</code>	Sum of numbers of unique destination transport layer ports.
<code>avg_n_dest_port</code>	The average number of unique destination transport layer ports.
<code>std_n_dest_port</code>	Standard deviation of numbers of unique destination transport layer ports.

Table 8: Content of the `ids_relationship.csv` file.

Column name	Description
<code>id_ip</code>	ID of the IP address.
<code>id_institution</code>	ID of the institution which owns the IP address.
<code>id_institution_subnet</code>	ID of the institution subnet in which the IP address belongs.

Table 9: Content of the `weekends_and_holidays.csv` file.

Column name	Description
Date	The date of the day in format “YYYY-MM-DD”.
Type	The type of the day (Weekend or Holiday).

Table 10: Content of the `school_terms_and_exams_periods.csv` file.

Column name	Description
Start	The first day of the period in format “YYYY-MM-DD”.
End	The last day of the period (inclusive) in format “YYYY-MM-DD”.
Type	The type of the academic period (WinterTerm, WinterExams, SummerTerm or SummerExams).

Table 11: Content of the `europa_time_changes.csv` file.

Column name	Description
date	The date of the daylight-saving transition in format “YYYY-MM-DD”.
transition	The direction of the transition (spring_forward or fall_back).
measured_from	The dataset part in which the transition falls inside the collection span (main or subnet_longitudinal).
local_clock_change	The local wall-clock jump, e.g. 02:00 CET → 03:00 CEST.
utc_offset_before	The UTC offset of local time before the transition (+01:00 for CET, +02:00 for CEST).
utc_offset_after	The UTC offset of local time after the transition.
offset_change_h	The signed change of the UTC offset in hours (+1 for spring-forward, −1 for fall-back).

Table 12: Content of the `outages.csv` file.

Column name	Description
id	Sequential ID of the outage.
part	The dataset part affected by the outage (main or subnet_longitudinal).
scope	The extent of the outage (infrastructure-wide or subnet-local).
kind	How the outage manifests in the data: grid-gap for windows absent from the shared time grid, present-zero for windows present in the grid but carrying zero traffic.
last_data_before	Timestamp of the last window carrying data before the outage, with explicit UTC offset.
data_resumed	Timestamp of the first window carrying data after the outage, with explicit UTC offset.
duration	The length of the outage in human-readable form, e.g. 15 d 21.0 h.
n_windows_10min	The number of 10-minute windows the outage spans.