CESNET-TIMESERIES 24: TIME SERIES DATASET FOR NETWORK TRAFFIC ANOMALY DETECTION AND FORECASTING

Josef Koumar^{1, 2, *}, Karel Hynek¹, Tomáš Čejka¹, and Pavel Šiška¹

¹CESNET a.l.e., Generála Píky 430/26, 160 00 Prague 6, Czech Republic
²Czech Technical University in Prague, Thákurova 9, 160 00 Prague 6, Czech Republic
*corresponding author: Josef Koumar (josef.koumar@fit.cvut.cz)

ABSTRACT

Anomaly detection in network traffic is crucial for maintaining the security of computer networks and identifying malicious activities. One of the primary approaches to anomaly detection are methods based on forecasting. Nevertheless, extensive real-world network datasets for forecasting and anomaly detection techniques are missing, potentially causing performance overestimation of anomaly detection algorithms. This manuscript addresses this gap by introducing a dataset comprising time series data of network entities' behavior, collected from the CESNET3 network. The dataset was created from 40 weeks of network traffic of 275 thousand active IP addresses. The ISP origin of the presented data ensures a high level of variability among network entities, which forms a unique and authentic challenge for forecasting and anomaly detection models. It provides valuable insights into the practical deployment of forecast-based anomaly detection approaches.

Keywords network traffic \cdot big-data \cdot time-series \cdot dataset \cdot anomaly detection \cdot forecasting \cdot network traffic forecasting \cdot network traffic prediction

1 Background & Summary

Traffic monitoring plays a crucial role in network management and overall computer security [1]. Network-based intrusion detection and prevention systems can protect infrastructure against users' sloppiness, policy violations, or intentional attacks from inside. However, maintaining network security has become increasingly challenging due to the widespread adoption of traffic encryption, which significantly reduces visibility.

As a result, gaining insights into encrypted network traffic has become essential, particularly for threat detection. Recent research has focused on detecting security threats through the classification of encrypted traffic using machine learning techniques [2, 3, 4, 5, 6]. However, in the domain of network traffic monitoring, there is a substantial challenge in acquiring up-to-date threat datasets [7]. Machine Learning classification models can detect already known attacks, which are captured in the dataset or those closely resembling them (such as malware from the same family). Therefore, unsupervised anomaly detection plays a crucial role in network traffic monitoring [8] as it can identify unknown (zero-day) attacks due to behavioral changes caused by infection [9].

The unsupervised anomaly detection method assigns anomalous scores to network entity behavior based on patterns and characteristics learned from historical data. One of the most used types of unsupervised anomaly detection algorithms is based on traffic forecasting (also referred as network traffic prediction). An anomaly alert is raised when the difference between the forecasted value and observation exceeds a defined threshold. Nevertheless, traffic forecasting can also be applied in other networking use cases, such as traffic management in data driven networks, resource allocation, and service orchestration.

In recent years, there has been a rapid development in forecasting and anomaly detection methods, not limited to computer science. Wu et al. [10] attributed this development to the rise and successful use of neural networks. Nevertheless, the recent performance improvement of forecasting methods applied to network traffic monitoring is uncertain due to the absence of long-term datasets [11]. In their survey, Ferreira et al. [11] describe the lack of a

reference dataset as the crucial obstacle related to performance evaluation. Additionally, real-world datasets used in the evaluation are not publicly available due to privacy concerns. Therefore, the majority of publicly available datasets have synthetic origins.

Synthetic data does not necessarily represent real-world tasks. Wu et al. [10] show that novel anomaly detection and forecasting approaches evaluated on synthetic datasets lead to the illusion of nonexisting progress. The more preferable option is to use up-to-date real-world data, like the MAWIlab project WIDE [12], that publishes anonymized packet captures daily. However, the WIDE project provides only brief 15-minute daily packet traces [13], which is an insufficient time window for effective traffic modeling.

To address these challenges, we decided to create a dataset called CESNET-TimeSeries24 that was collected by long-term monitoring of selected statistical metrics for 40 weeks for each IP address on the ISP network CESNET3 (Czech Education and Science Network). The dataset encompasses network traffic from more than 275,000 active IP addresses, assigned to a wide variety of devices, including office computers, NATs, servers, WiFi routers, honeypots, and video-game consoles found in dormitories. Moreover, the dataset is also rich in network anomaly types since it contains all types of anomalies identified by Chandola et al. and Basdekidou et al. [14, 15], ensuring a comprehensive evaluation of anomaly detection methods. Last but not least, the CESNET-TimeSeries24 dataset provides traffic time series on institutional and IP subnet levels to cover all possible anomaly detection or forecasting scopes. Overall, the time series dataset was created from the 66 billion IP flows that contain 4 trillion packets that carry approximately 3.7 petabytes of data. The CESNET-TimeSeries24 dataset is a complex real-world dataset that will finally bring insights into the evaluation of forecasting models in real-world environments.

2 Methods

In this section, we provide detailed information about all methods used for obtaining the dataset. Since the dataset was obtained from a production network, CESNET3, and used by real users, privacy was a fundamental concern in our work, leading us to conduct our research with careful consideration. The indisputable advantages of real traffic generated by hundreds of thousands of people come with understandable privacy concerns. Thus, we used only automatic data processing with immediate data anonymization. With this, we declare that we did not analyze or manually process non-anonymized data or perform any procedures that could 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 0000-10/24/51902/EKČVUT. The approval also includes a waiver of explicit user consent for publishing the dataset. Moreover, all users of the CESNET3 network agreed with the terms and conditions that define a monitoring process for 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).

2.1 Data capture

The network traffic was obtained from the CESNET3—an ISP network that provides internet access to public and research institutions in the Czech Republic. The network spans a whole country, as shown in Figure 1, and serves approximately half a million users daily. Since ISP networks transfer huge volumes of data, packet-based monitoring systems are infeasible due to the costs of processing and storage capacity. Therefore, the ISP networks (including CESNET3) are monitored using a standard IP flow monitoring system located at the perimeter—all transit lines to the peering partners are equipped with flow monitoring probes.

IP flow monitoring systems aggregate data packets into IP flow records. An IP flow record represents communication metadata associated with a single connection and is defined [16] 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." Commonly, these properties are referred to as flow keys and consist of source and destination IP addresses, transport layer ports, and protocol.

The monitoring infrastructure for creating the dataset is detailed in Figure 2. A network TAPs are positioned before the edge routers of the CESNET3 network and mirror the traffic to the monitoring probe, which is a server equipped with a high-speed monitoring card capable of handling 200 GB/s. On the monitoring probe, the Ipfixprobe (https://github.com/CESNET/ipfixprobe) flow exporter is installed. The Ipfixprobe exporting process was set with an active timeout of 5 min and an inactive (idle) timeout of 65 s. Long connections are split when the connection duration is longer than the active timeout, and a flow record is exported even though the actual connection is not terminated yet. If no packet is observed within the inactive timeout period, the connection is considered terminated, and a flow record is exported. Using active and inactive timeouts for splitting connections is standard practice for flow-based network monitoring [16]. Additionally, the Ipfixprobe collected following features: start time, end time, number of packets,

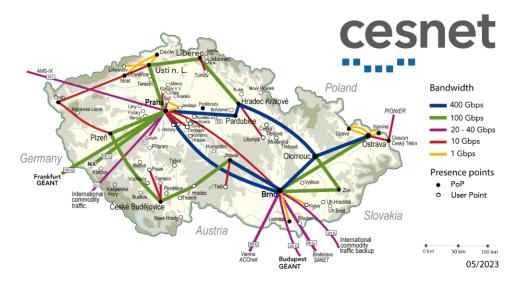


Figure 1: Topology of the CESNET3 network, which interconnects academic institutions in the Czech Republic

number of bytes, and Time to Live (TTL). None of the collected information contains application-level information, which ensures that the privacy of users' communication is not compromised. The collected data are then sent using IPFIX [16] protocol to the IP flow collector server with IPFIXcol2 (https://github.com/CESNET/ipfixcol2) flow collecting software installed.

2.2 Time series aggregation

The flow collector server contains aggregation and filtration modules, as depicted in Figure 2. The filtration module removes all transient traffic—where both source and destination addresses do not belong to CESNET3 and the packets are just passing through the network. Moreover, we also filter out all single TCP-SYN packet connections—scans. Given the ISP origin and probe placement before routers and firewalls, the scans would represent the absolute majority of the dataset. Moreover, a large number of scans would cause significant noise in the time series, which would result in possible false negatives in anomaly detection. Since the scans can be easily detected with simple methods [17, 18], we decided to remove them from the dataset. All other flows are then passed to the aggregation module.

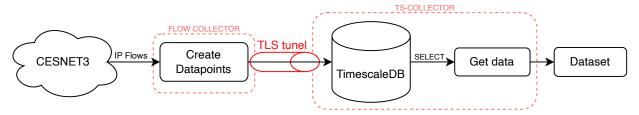


Figure 2: Architecture of dataset collection from the CESNET3 network

We generate evenly spaced time series for each IP address by aggregating IP flow records into time series datapoints. The process diagram for network traffic processing and aggregation is shown in Figure 3. The created datapoints represent the behavior of IP addresses within a defined time window of 10 minutes. The vector of time-series metrics $v_{ip,i}$ describes the IP address ip in the i-th time window. Thus, IP flows for vector $v_{ip,i}$ are captured in time windows starting at t_i and ending at t_{i+1} . The aggregated datapoints are then stored in TimeScaleDB (https://www.timescale.com/), where time series are built.

Datapoints created by the aggregation of IP flows contain the following time-series metrics:

• *Simple volumetric metrics:* the number of IP flows, the number of packets, and the transmitted data size (i.e. number of bytes)

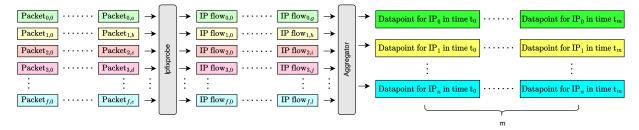


Figure 3: This diagram describes the aggregation process for capturing time series with length m from network traffic. For each packet $Packet_{i,j}$ exists $Flow_{i,k}$ where the packet belongs. Furthermore, it is always true that $a \geq g$ (for $b \geq h$ and others similarly), and in most cases, the a will be much larger than g. Moreover, it is common that only one IP flow contains all packets from one connection (g = 0). This is common, for example, for connections generated by a user visiting a web page. Similarly, a time series datapoint is a combination of one or more IP flows.

- *Unique volumetric metrics*: the number of unique destination IP addresses, the number of unique destination Autonomous System Numbers (ASNs), and the number of unique destination transport layer ports. The aggregation of *Unique volumetric metrics* is memory intensive since all unique values must be stored in an array. We used a server with 41 GB of RAM, which was enough for 10-minute aggregation on the ISP network.
- *Ratios 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)

2.3 Anonymization

The capturing process started on 9. October 2023 and ended on 14. July 2024. Thus, after 40 weeks, we extracted time series from the database. In the database framework, a script is deployed that automatically adds corresponding institution and institution subnet for each IP address. Nevertheless, we omit the extraction of exact IP addresses, institutions, and institution subnets. Instead, we used database IDs for IP addresses, institutions and institution subnets as an identifiers, which were randomly assigned during database creation. The omitting IP addresses, institutions and institution subnets in the data extraction performs effective anonymization. Nobody without access to the original database cannot revert this step and connect time series with a particular IP address, institution, or institutional network.

2.4 Data preprocesing

This subsection describes the preprocessing steps in detail.

Filtering: The obtained raw dataset from the database contains a time series for approximately 400 thousand IP addresses. However, many of them were almost empty—network entities were not active most of the dataset creation time, resulting in an empty 10-minute aggregated record. Therefore, we remove time series with a smaller number of datapoints than 100, which is approximately 0.25% of the maximum datapoints that the time series in this dataset can contain. This action results in a time series for 275,124 IP addresses.

Multiple time aggregation: The original datapoints in the dataset are aggregated by 10 minutes of network traffic. The size of the aggregation interval influences anomaly detection procedures, mainly the training speed of the detection model. However, the 10-minute intervals can be too short for longitudinal anomaly detection methods. Therefore, we added two more aggregation intervals to the datasets.

We provide additional one-hour and one-day aggregation intervals. These aggregated intervals were created from the 10-minute interval. When possible, time series metrics were aggregated using the sum of values (such as IP flows, number of packets and transimitted bytes). Nevertheless, metrics that represent a number of unique values cannot be easily summed without losing potentially important information. Therefore, in that case, we provide sum, mean, and standard deviation that results in three new time series metrics per each original metric. Finally, we use the mean for the time series metrics, which represent ratio or average values.

Time series of institutions: Many security events can be visible only from the perspective of overall network traffic. Therefore, we use the institution ID exported from the database to divide the IP addresses into groups based on institutions. We identify 283 institutions inside the CESNET3 network. These time series aggregated per each institution ID provide a view of the institution's data.

Time series of institutional subnets: Many institutions have multiple networks, which are usually in different locations or have different purposes in the organization. The administrators usually like to handle security not only overall but also per each subnet. Therefore, we divide the IP addresses into groups based on institution subnets in the same location using information from ISP CESNET. These time series aggregated per each institution's network provide a view of data that would probably be monitored by the institution's SoC team.

We identify 548 institution subnets inside the CESNET3 network. Almost 75 % of institutions have exactly one network. Moreover, the next 14 % of institutions have two networks. However, there is also an institution with 105 networks. This is because the CESNET3 network interconnects to a lot of high schools, hospitals, and museums. Therefore, 75 % of institutions' time series is in the institution subnets' time series dataset.

Weekend and holidays: In the network, traffic forecasting and anomaly detection can be crucial to add information about weekends and holidays into the model training and evaluation. Therefore, we included weekend dates and dates of public holidays in the Czech Republic for convenience.

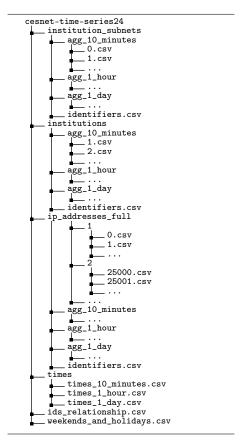


Figure 4: The file structure of the CESNET-TimeSeries24 dataset.

3 Data Records

The dataset is delivered in the form of compressed CSV files and is available at the Zenodo Platform via URL link: https://zenodo.org/records/13382427. This section provides the dataset structure and explains all data fields and files.

3.1 Data file hierarchy

The dataset structure is outlined in Figure 4. Each time series type (either institutional subnets, institutions, or raw IP addresses) is also divided by the aggregation window (10 minutes, 1 hour, and 1 day). The time series data, containing all relevant features, is stored in CSV files. Each file is named according to the identifier of the entity whose behavior the time series follows, ensuring a clear association between the time series and the corresponding entity. Since identifiers

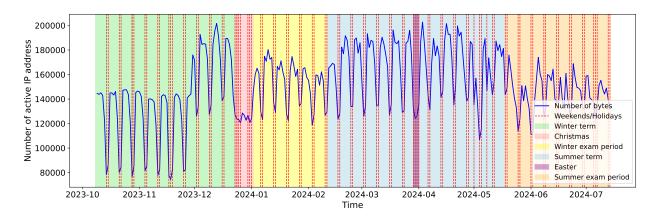


Figure 5: The evolution of the number of active IP addresses for each dataset's day

are not a row of consecutive numbers due to previous filtration and some numbers are missing, each time series type has its own identifiers.csv file, which contains lists of all identifiers of the included entities.

The time series representing the activity of individual IP addresses is additionally divided into multiple subdirectories. Since there are 275,124 IP address time series in the dataset, each located in its own file, some filesystems had difficulty handling such a large number of files in a single directory. Therefore, we organized them into subdirectories, each with 25,000 files. For individual IP addresses, the identifiers.csv file also includes the name of the subdirectory where the corresponding time series data is stored.

The dataset also contains a times directory, which contains translation tables between the sequence number of the aggregation time window and the absolute time. Such translation is often useful for plots with forecasting performance. Finally, it also contains a relationship table between IP addresses, institutions, and institution subnets, as well as a table denoting public holidays and weekends.

3.2 File format

The dataset files are organized as tables in CSV file format. There are two different file formats for the time series. The time series aggregated over 10 minutes contain columns described in Table 1. The time series aggregated over one hour and one day contains additional data features due to reaggregation. Therefore, apart of features described in Table 1 it also contains features described in Table 2. Moreover, the dataset contains two support files—ids_relationship.csv and weekends_and_holidays.csv. Table 3 describes the content of ids_relationship.csv, which provides information about the relationship between the IP addresses, institutions, and institution subnets dataset types. Furthermore, Table 4 describes the content of weekends_and_holidays.csv that provides information about which days are non-working days in the Czech Republic (weekends and national holidays).

4 Technical Validation

This section provides technical validation of the dataset and is divided into three parts: 1) Validation of overall dataset properties, 2) Validation of the existence of anomalies, and 3) Validation of usability of the dataset for forecasting approaches.

4.1 Validation of overall dataset properties

In this section, we aim to validate the overall statistical properties of the dataset across the 40 weeks.

Activity of IP addresses The dataset contains network traffic of more than 275 thousand IP addresses. However, these IP addresses typically do not communicate constantly over time. The evolution of the number of active IP addresses for each dataset's day is shown in Figure 5. We can see that the number of active IP addresses correlates with the weekends and holidays, which is highly expected behavior. Moreover, we can see a slight correlation with terms and exam periods, which is caused by the CESNET3 interconnecting many universities and dormitories.

Table 1: Detailed descriptions of time series metrics for each IP address dataset.

Column name	Description		
id_time	Unique identifier for each aggregation interval within the time series, used to segment the dataset into specific time periods for analysis.		
n_flows	Total number of flows observed in the aggregation interval, indicating the volume of distinct sessions or connections for the IP address.		
n_packets	Total number of packets transmitted during the aggregation interval, reflecting the packet-level traffic volume for the IP address.		
n_bytes	Total number of bytes transmitted during the aggregation interval, representing the data volume for the IP address.		
n_dest_ip	Number of unique destination IP addresses contacted by the IP address during the aggregation interval, showing the diversity of endpoints reached.		
n_dest_asn	Number of unique destination Autonomous System Numbers (ASNs) contacted by the IP address during the aggregation interval, indicating the diversity of networks reached.		
n_dest_port	Number of unique destination transport layer ports contacted by the IP address during the aggregation interval, representing the variety of services accessed.		
tcp_udp_ratio_packets	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 <0, 1> where 1 is when all packets are sent over TCP, and 0 is when all packets are sent over UDP.		
tcp_udp_ratio_bytes	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 <0, 1> with same rule as tcp_udp_ratio_packets		
dir_ratio_packets	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 <0, 1>, 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.		
dir_ratio_bytes	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 <0, 1> with the same rule as dir_ratio_packets.		
avg_duration	Average duration of IP flows for the IP address during the aggregation interval, measuring the typical session length.		
avg_ttl	Average Time To Live (TTL) of IP flows for the IP address during the aggregation interval, providing insight into the lifespan of packets.		

Table 2: Time series metrics which replace metrics n_{dest_ip} , n_{dest_asn} and n_{dest_port} in aggregation.

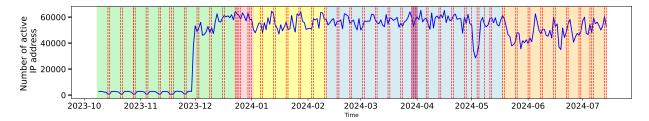
Column name	Description	
sum_n_dest_ip Sum of numbers of unique destination IP addresses.		
avg_n_dest_ip	The average number of unique destination IP addresses.	
std_n_dest_ip	Standard deviation of numbers of unique destination IP addresses.	
sum_n_dest_asn	_n_dest_asn Sum of numbers of unique destination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

Table 3: Content of the ids_relationship.csv file

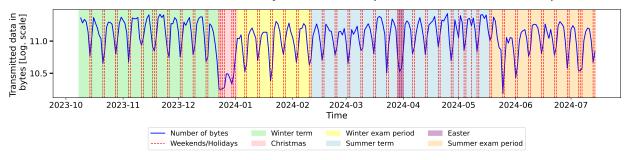
Column name	Description	
id_ip	ID of the IP address	
id_institution	ID of the institution which own the IP address	
id_institution_subnet	ID of the institution subnet in which the IP address belongs	

Table 4: Content of the weekends_and_holidays.csv file

Column name	nn name Description	
Date	The date of the day in format "YYYY-MM-HH"	
Туре	The type of the day (Weekend or Holiday)	



(a) The evolution of the number of active problematic university's IP addresses for each dataset's day



(b) The evolution of overall transmitted data by problematic university

Figure 6: The comparison of evolution of active IP addresses and transmitted data by problematic university

However, the number of active IP addresses increased by approximately 50 thousand IP addresses at the end of November. We further evaluated this anomaly and found out it was caused by a single institution, which is one of the universities connected to the CESNET3 network. The evolution of the number of active IP addresses belonging to this university is shown in Figure 6a. We can see that the number of IP addresses significantly increased at the end of November. However, the evolution of transmitted data does not correlate with the evolution of a number of active IP addresses which can be seen in Figure 6b.

We discussed our findings with experts in CESNET. We provided them with the newly occurring IDs of IP addresses, and they found out that this anomaly is caused by the change in the university's network architecture, which, from the end of November, used public IP addresses for Eduroam WiFi.

Evolution of transmitted data Overall data that were transmitted by IP addresses in the dataset is shown in Figure 7. The transmitted data in the aggregation window is stored in the time series metrics n_bytes. In the figure, it can be seen that the evolution of transmitted data size highly correlates with weekends and holidays. Moreover, CESNET3 interconnects universities and dormitories; thus, we can also see a correlation between the terms and the exam periods. This observation is in line with the findings of Beneš et al. [19].

Furthermore, it can be seen that the decrease in traffic after the end of the summer term is even larger than the decrease in traffic during Christmas. Therefore, we performed the evaluation of this anomaly, and we found out that one of the monitoring probes was broken from approximately 2024.05.21 16:30 to approximately 2024.06.04 20:00. So, the probe did not send IP flows to the IP flow collector.

Gaps in time series Real-world network traffic data often contains gaps where a device does not transmit any data. In some instances, the entire network's traffic may exhibit such gaps. These gaps, or spaces, pose a challenge that must be addressed before applying forecasting methods.

One way to manage these gaps is through the aggregation process. If the aggregation window is sufficiently large, the resulting time series might eliminate these gaps. However, for scenarios involving multiple processes, such as traffic from multiple devices or entire networks, it's unlikely that a single aggregation window will be effective across the board. Additionally, using a large aggregation window can obscure important patterns in the time series, such as anomalies. As a result, it's inevitable that some time series will contain gaps.

In our dataset, many time series contain significant gaps. In Figure 8a shows the average percentage of active datapoints in these time series, along with the standard deviation. For the 10-minute aggregation interval, active datapoints make

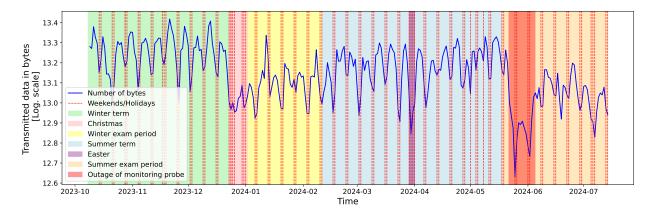
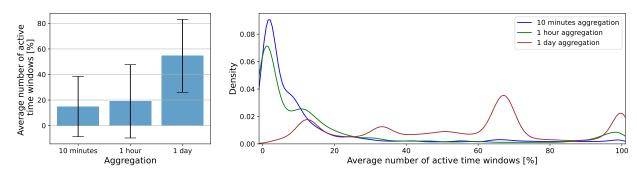


Figure 7: Overall data that were transmitted on the CESNET3 network and are captured in the dataset



- aggreagation windows
- (a) Analysis of the average number of active (b) Analysis of active time windows using Kernel Density Estimation (KDE) density function

Figure 8: Analysis of gaps in dataset

up less than 20% of the time series, meaning that gaps constitute more than 80% on average. As expected, the number of gaps decreases as the aggregation interval increases. However, even with a day-long aggregation interval, gaps still account for more than 40% of the time series on average.

Further analysis reveals an additional distribution pattern. The Kernel Density Estimation (KDE) distribution function in Figure 8b highlights this, particularly during the day aggregation, where there is a peak between 60% and 70%. This peak corresponds to the percentage of working days in the dataset, which is 67.5%. This suggests that a significant portion of the time series represents workstations, as expected.

Validation of existence of anomalies

Anomaly types in dataset There are many types of anomalies that were described in the literature. All of these anomalies are present in this dataset. Examples of the anomaly types occurring in the dataset are shown in Figure 9.

The first type of anomaly is *Point anomaly* [14], which is a single data point that significantly deviates from the rest of the datapoints in the time series. There are two types of point anomalies:

- Global A global outlier is a data point that deviates significantly from the overall pattern or distribution of the entire dataset. It is an extreme value when compared to the rest of the data.
- Contextual A data point that is an outlier within a specific context or condition but not necessarily when viewed in a different context.

The second type of anomaly is Collective Anomaly [14], which is a sequence of datapoints that is anomalous when considered together, even if individual points might not be. There are two types of collective anomalies:

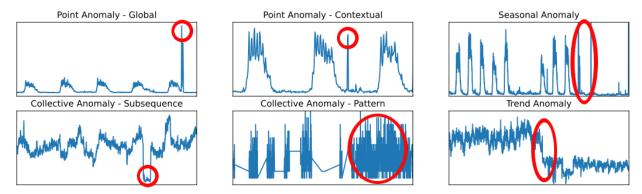


Figure 9: Examples of the anomaly types occurring in the dataset

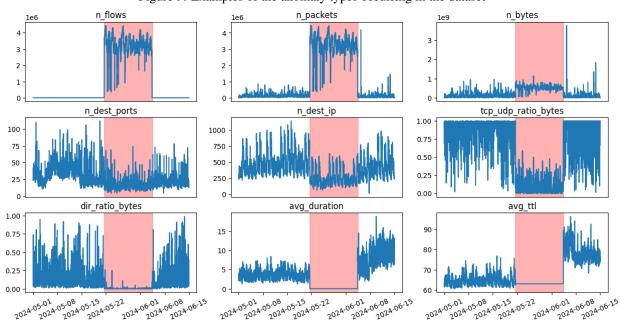


Figure 10: Analysis of time series metrics of detected anomaly on time series IP address ID 1367 identified as DoS by CESNET experts

- Subsequence A contiguous subsequence of datapoints that is anomalous compared to the rest of the time series.
- Pattern A sequence of datapoints that together form an unusual pattern, which does not conform to the known patterns of the time series.

The third type of anomaly is *Trend Anomaly* [15] which is an unexpected change in the trend of the time series data, such as a sudden shift from a positive to a negative trend. Similar data behaviors in data science are also called data or content drifts [20], so, sometimes, this anomaly type is called *Drift Anomaly*.

Security incident analysis In the analysis of the detected anomaly within the dataset, it is crucial to provide a clear explanation of the metrics that characterize the event. Specifically, for the anomaly detected on time series with IP address ID 1367, identified as a Denial of Service (DoS) attack by CESNET experts, the time series metrics were scrutinized in detail, as illustrated in Figure 10.

The analysis reveals several key observations. Firstly, both the flow and packet counts increase significantly, yet they rise in parallel, indicating a one-packet flow characteristic. Despite the increase in flows and packets, the number of bytes transmitted does not rise correspondingly, suggesting that the packets involved are small in size. Furthermore, the number of destination ports remains steady, dismissing the possibility of a scan. A significant shift in the TCP/UDP ratio is observed, with almost all the anomalous traffic being UDP, further supporting the DoS identification.

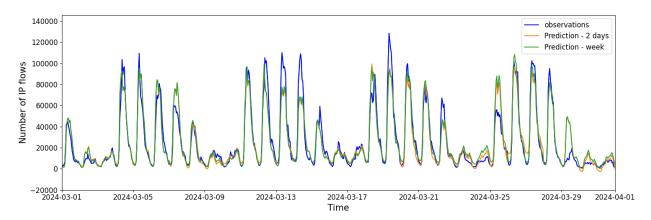


Figure 11: Example of using SARIMA model with order (1, 1, 1) and with seasonal order (1, 1, 1, 168) for forecasting of number of IP flows for time series of IP address's ID 103

Additional metrics strengthen this conclusion. The traffic direction is near zero, indicating that nearly all the anomalous traffic is directed toward the monitored IP address. The average duration of the flows is also close to zero, which is consistent with one-packet flows typically seen in DoS attacks. The Time To Live (TTL) values remain nearly constant during the anomaly, implying that the traffic likely originates from a single source. Finally, the number of destination IPs shows a slight decrease, reinforcing the idea of a singular sender and confirming the effectiveness of the DoS attack. These combined metrics clearly point to a DoS event targeting the monitored IP address.

4.3 Validation of usability of dataset

For the validation of usability, we decided to demonstrate the usage of the dataset's time series for network traffic forecasting. We select an IP address's time series with an ID 103, the number of IP flows, and a one-hour aggregation interval. To demonstrate time series forecasting applied to the dataset's data, we select the basic SARIMA (Seasonal Autoregressive Integrated Moving Average) as the forecasting model—the order was equal to (1,1,1) and seasonal order equal to (1,1,1,168). This means that each of the three components of the model has the same weight in the modeling, and the seasonality was set to one weak (168 hours).

The SARIMA were trained on the monthly data (31 datapoints). We select two different prediction interval to demonstrate the difference. The first prediction window a 7 days long, and the second prediction window is 2 days. Furthermore, the model was retrained with the sliding window equal to the prediction window. Therefore, the model predicted 36 weeks.

The result of this demonstration is shown in Figure 11. The Figure contains only one of The predictions for 2 days, and the model retraining after 2 days resulted in slightly better results, as can be seen. Moreover, we can compare the results by using evaluation metrics like Root Mean Square Error (RMSE), Symmetric Mean Absolute Percentage Error (SMAPE), and R^2 Score. In all used metrics, a lower value represents better forecasting performance. The predicted week data achieves 10951.26 RMSE, 40.66 SMAPE, and 0.77 R^2 Score. And the predicted 2 days of data achieve 10293.81 RMSE, 40.86 SMAPE, and 0.79 R^2 Score. Therefore, the predicted 2 days achieve better RMSE and R^2 Score. However, it achieves a slight decrease in the SMAPE metric.

5 Usage Notes

This section describes recommendations for using the evaluation of network traffic forecasting (and forecasting-based anomaly detection) in detail. We recommend authors follow this evaluation procedure in order to achieve one of the motivations for the creation of this dataset, which is comparability between approaches. We believe that following recommendations will help the community to process our dataset and be able to compare with different approaches that also follow our recommendations. Thus, we present a checklist for addressing all our recommendations which is shown in Table 5. Furthermore, the example of using the dataset in the form of a Jupyter Notebook is available on GitHub (https://github.com/koumajos/CESNET-TimeSeries24-Example). Moreover, the source codes of experiments that provide the evaluation example using our recommendations are also available on this GitHub repository.

Table 5: Checklist for Addressing Recommended Evaluation Procedures

No.	Recommendation		
(1)	Specify which dataset(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 handling gaps in time series.		
(6)	Ensure the training phase starts from the beginning of the dataset's time frame (2023-10-09).		
(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 the evaluation phase.		
(10)	Specify the forecasting horizon (length of time into the future for predictions).		
(11)	Clearly specify the evaluation metrics used in the article.		
(12)	Provide an overall comparison across each time series using statistical distributions and aggregate statistics.		
(13)	Assess and document the computational requirements and deployability of the model.		
(14)	Make source codes of your experiments and model publicly available for the community.		

Dataset Selection The dataset utilized in this study is divided into four distinct parts, each of which can be independently used for evaluation: the Full IP address dataset, the Sample IP address dataset, the Institutions dataset, and the Institution subnets dataset. Therefore, it is imperative for authors to clearly state which dataset type(s) they are using in their analyses (Recommendation 1). Moreover, each dataset contains three aggregation intervals; thus, the aggregation interval(s) used must also be specified (Recommendation 2). In cases where multiple dataset types and/or aggregation levels are employed, results must be reported separately for each dataset and aggregation without combining them. This ensures the clarity and reproducibility of the findings.

Furthermore, the approach must clearly indicate whether it is multivariate (multiple time series metrics are modeled simultaneously) or univariate (each time series metric is modeled individually) (Recommendation 3). If not all metrics are used, this must be explicitly stated (Recommendation 4), and comparisons with different approaches should only be made in the following cases. First, for univariate approaches, comparisons should be made individually per each metric. Second, for multivariate approaches, different methods must use the same metrics to ensure accurate comparison.

Moreover, when any preprocessing steps are applied to these dataset parts, it is crucial that these steps are thoroughly described (Recommendation 5). This includes a detailed description of any filtering, normalization, or transformation processes. Furthermore, this also handles gaps in time series, which must be addressed and described in detail. Particularly, if the preprocessing involves filtering the time series data, this may lead to results that are not directly comparable with studies using unfiltered versions of the same dataset types. Therefore, such preprocessing steps should be clearly justified, and their impact on the analysis should be discussed.

Given that the Full IP address dataset contains more than 275 thousand time series, evaluating methods for all these time series is challenging. Therefore, we encourage authors to create new, smaller datasets featuring interesting time series behaviors from the Full IP address dataset and share them with the community via platforms such as Zenodo. This practice would facilitate further research by providing accessible, focused datasets that highlight specific patterns or anomalies, fostering collaboration and innovation within the research community.

Training Correctness To ensure the integrity and validity of the training process, several key guidelines must be followed. The training phase of the time series must always commence from the very beginning of the dataset's time frame, which starts on 2023-10-09 (Recommendation 6). This causes the model to be trained on the entire range of available data, capturing all relevant trends and patterns, ensuring performance results comparability. The duration of the training window must be explicitly specified in the article (Recommendation 7). This includes detailing how much historical data is used to train the model before it begins making predictions.

Moreover, if a validation window is employed during the model development process, the duration and purpose of this validation window must be clearly defined (Recommendation 8). This helps in assessing the model's performance on unseen data before it is applied to the test set. Furthermore, if the model is retrained during the evaluation phase, this retraining process must be clearly described in the article (Recommendation 9). Authors should specify the retraining frequency, the data used for retraining, and how the retrained model is validated.

Forecasting Correctness To ensure consistency and transparency in the forecasting methodology, authors must clearly describe the following aspects of their prediction process. Authors must specify the length of time into the

future for which predictions are made (Recommendation 10). This could be a fixed period (e.g., one week ahead) or a rolling window that adjusts over time. Authors should explain whether the window is shifted by a fixed interval or if it adapts based on certain criteria.

Evaluation metrics The evaluation of the forecasting model should be done by the evaluation metrics. The chosen metrics must be clearly specified in the article (Recommendation 11). We recommend using the following metrics for evaluation (n is the number of observations, y_i are the actual observed values in the time series, $\hat{y_i}$ are the predicted values):

• Root Mean Squared Error (RMSE) calculates the square root of the average squared errors, maintaining the same units as the original data. It is sensitive to large errors, similar to MSE, which helps in detecting significant anomalies. The downside is that this sensitivity might distort the overall model assessment if large errors are not critical. The RMSE can be computed by the equation 1. [21]

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

Moreover, for the combination of multiple RMSEs, we recommend using the weighted RMSE, which can be calculated using the equation 2.

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

where σ_i^2 represents the variance of the true values in the i^{th} dataset, and RMSE_i is the corresponding RMSE.

• Symmetric Mean Absolute Percentage Error (SMAPE) addresses some issues of MAPE by symmetrizing the error calculation, providing more stability when actual values are near zero. This symmetry makes it less biased towards overestimations or underestimations. However, SMAPE can still be less intuitive than simpler metrics and may over-penalize certain error types. The SMAPE can be computed by the equation 3; the ϵ in the equation is a small constant that is added to avoid division by zero. [21]

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

Moreover, for the combination of multiple SMAPEs, we recommend using the mean and standard deviation of SMAPEs. The weighted SMAPE can also be calculated using the equation 4.

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

where σ_i^2 represents the variance of the true values in the i^{th} dataset, and SMAPE_i is the corresponding SMAPE.

• Coefficient of Determination (R^2 Score) measures the proportion of variance explained by the model, providing insight into the model's explanatory power. It is useful for comparing models, but it can be misleading for non-linear models and does not directly measure prediction accuracy. The R^2 Score can be computed by the equation 5. [22, 21]

$$R^{2} = 1 - \frac{\sum_{i=1}^{n} (y_{i} - \hat{y}_{i})^{2}}{\sum_{i=1}^{n} (y_{i} - \mu_{y})^{2}}$$
 (5)

Moreover, for combining multiple \mathbb{R}^2 scores, we recommend using the weighted \mathbb{R}^2 score, which can be calculated using the equation 6.

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

where σ_i^2 represents the variance of the true values in the i^{th} dataset, and R_i^2 is the corresponding R^2 score.

Table 6: 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

Multiple Time Series Evaluation All recommended metrics are computed individually for each time series. However, an overall comparison must be made across each time series in the dataset (Recommendation 12). Therefore, we recommend using statistical distributions per metric to compare overall performance. Initially, aggregate statistics such as the mean and standard deviation should provide a general sense of precision across the dataset, or the weighted variation of the metrics described before can be used. For a more detailed evaluation, we suggest utilizing distribution plots, such as histograms or KDE plots. KDE plots, in particular, are highly effective for detailed comparisons across multiple models, offering a nuanced view of the distribution and performance variations.

Computational Requirements and Deployability of the Model When evaluating a model's performance, it is crucial not only to consider its precision but also to assess its computational requirements and feasibility of the deployment (Recommendation 13). The computational complexity of the model should be analyzed, taking into account factors such as training time, inference speed, and resource consumption (e.g., CPU/GPU usage, memory footprint). Models that require excessive computational resources may be impractical for real-time applications or deployment in environments with limited resources.

Code Availability

The dataset has been produced using open-source software. The flow exporter Ipfixprobe, flow collector IPFIXcol2, the NEMEA processing system, and the NEMEA modules are available on GitHub. We use the TimeScaleDB database. Moreover, we provide Create Datapoint module and deployment scripts for the NEMEA Supervisor and for building the database. The versions of used software with links to corresponding repositories are summarized in Table 6.

Acknowledgements

This research was funded by the Ministry of Interior of the Czech Republic, grant No. VJ02010024: Flow-Based Encrypted Traffic Analysis and also by the Grant Agency of the CTU in Prague, grant No. SGS23/207/OHK3/3T/18 funded by the MEYS of the Czech Republic. This research was also 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. propose the time series metrics, J.K. designs and implements the architecture for collecting time series metrics, K.H. and P.Š. design and implement the architecture of capturing network traffic, J.K. and K.H. preprocess data to the final version of the dataset, T.Č. handled funding and 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] Jan Luxemburk and Tomáš Čejka. Fine-grained tls services classification with reject option. *Computer Networks*, 220:109467, 2023.
- [6] Richard Plnỳ, Karel Hynek, and Tomáš Čejka. Decrypto: Finding cryptocurrency miners on isp networks. In *Nordic Conference on Secure IT Systems*, pages 139–158. Springer, 2022.
- [7] Jorge Luis Guerra, Carlos Catania, and Eduardo Veas. Datasets are not enough: Challenges in labeling network traffic. *Computers & Security*, 120:102810, 2022.
- [8] 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.
- [9] 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.
- [10] Renjie Wu and Eamonn J Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE transactons on knowledge and data engineering*, 35(3):2421–2429, 2021.
- [11] 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.
- [12] Kenjiro Cho, Koushirou Mitsuya, and Akira Kato. Traffic data repository at the {WIDE} project. In 2000 USENIX Annual Technical Conference (USENIX ATC 00), 2000.
- [13] 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.
- [14] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys* (CSUR), 41(3):1–58, 2009.
- [15] Vasiliki A Basdekidou. The momentum & trend-reversal as temporal market anomalies. *International Journal of Economics and Finance*, 9(5):1–19, 2017.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] Indrė Žliobaitė, Mykola Pechenizkiy, and Joao Gama. An overview of concept drift applications. *Big data analysis: new algorithms for a new society*, pages 91–114, 2016.
- [21] 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.
- [22] Sewall Wright. Correlation and causation. Journal of agricultural research, 20(7):557, 1921.