Host Behavior in Computer Network: One-Year Study

Tomas Jirsik and Petr Velan

***Abstract*—An analysis of a host behavior is an essential key for modern network management and security. A robust behavior profile enables the network managers to detect anomalies with high accuracy, predict the host behavior, or group host to clusters for better management. Hence, host profiling methods attract the interest of many researchers, and novel methods for host profiling are being introduced. However, these methods are frequently developed on preprocessed and small datasets. Therefore, they do not reflect the real-world artifacts of the host profiling, such as missing observations, temporal patterns, or variability in the profile characteristics in time. To provide the needed insight into the artifacts of host profiling in real-world settings, we present a study of the host behavior in a network conducted on a one-year-long real-world network dataset. In the study, we inspect the availability of the data for host profiling, identify the temporal patterns in host behavior, introduce a method for stable labeling of the hosts, and assess the variability of the host characteristics in the course of the year using the coefficient of variance. Moreover, we make the one-year dataset containing nine characteristics used for host behavior analysis available for public use and further research, including selected use cases representing host profiling caveats. We also share the record of analysis presented in the paper.**

***Index Terms*—Network measurement, host profiling, netflow, clustering, temporal patterns.**

I. Introduction

EFFECTIVE network management requires advanced visibility and awareness of the processes taking place in the computer network. For this purpose, network administrators take advantage of various systems and tools that ease the comprehension and provide network visibility. A significant part of these tools leverage host profiling. The host profiling aims to identify the typical behavior of a host that can be utilized for identification of anomalous behavior or prediction of its expected behavior. Moreover, based on the behavior profile, the hosts can be aggregated into clusters of hosts with similar properties, such as deployed services, which aids the network management as well as security policy definition and application.

Various approaches to host profiling ranging from the entropy-based approaches to supervised and unsupervised machine learning approaches can be found in the literature [1], [2], [3], [4], [5]. The most common approach in the host profiling research is to describe features used for profile computation (e.g., source ports used by a host, entropy of destination IP addresses a host communicated to) and to provide algorithms used for computing the profile (e.g., graphlets [6]). The profiles computed from either synthetic or real-world dataset are then evaluated from several aspects, including a statistical description of the profile characteristics, consistency of the host clustering, and precision of the host profile classification. However, temporal aspects of the host profiling are mostly neglected in the current research.

The behavior of a host in a network evolves in time. Successful application of existing host profiling research into real-world deployment requires an analysis of host behavior and its changes from a long-term perspective. Nevertheless, the current state-of-the-art host profiling research rarely takes the real-world temporal aspects into account. The datasets utilized for profile creation and temporal evaluation span from 15 minutes in [7] to one month period in [8]. The length of these time spans is insufficient to reflect the long-term behavior of a host reliably. Furthermore, the authors focus primarily on the optimization of algorithms for host profiling and do not fully appreciate the variability of the hosts' behavior. However, long-term communications patterns (e.g., a host communicating only a short period of the time in a week) and behavior stability (e.g., the variability of a given feature over a day) significantly affect the computation of a host behavioral profile. If they are not taken into account, the host profiling algorithms provide biased results, which may lead to incorrect management decisions.

To address the subject of long-term host behavior, we present a comprehensive study that shows the artifacts influencing the host profiling from a long-term perspective. Using IP flow network monitoring infrastructure, we have collected data from /16 IPv4 CIDR university network over one year period. Since the university network environment is diverse and includes various kinds of hosts, the obtained dataset covers a large variety of host behaviors. We have conducted an explanatory analysis of the dataset to determine the influence of data availability, temporal pattern identification, and profile characteristics stability on host profiling.

Manuscript received May 1, 2020; revised September 22, 2020 and October 26, 2020; accepted November 2, 2020. Date of publication November 6, 2020; date of current version March 11, 2021. The methodology and result discussions were supported by the ERDF project "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822). Data collection and cleaning, data analysis, use-cases presentation, and state-of-the-art overview were supported by the Sharing and Automation for Privacy Preserving Attack Neutralization project. This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant agreement No 833418. The associate editor coordinating the review of this article and approving it for publication was T. Inoue. *(Corresponding author: Tomas Jirsik.)*

The authors are with the Institute of Computer Science, Masaryk University, 601 77 Brno, Czech Republic (e-mail: irsik@ics.muni.cz; velan@ics.muni.cz).

Digital Object Identifier 10.1109/TNSM.2020.3036528

TABLE I
AN OVERVIEW OF THE STUDY OBJECTIVES

Research Area	Research Questions	Purpose
Data Availability	How are the missing observations distributed?	Identify a reasonable time window for host profile
(Sections IV-A, V-A)	How to handle missing observations?	Show impact on the host profile computation
Temporal Stability	Are there temporal patterns in host behavior?	Validate and explore the temporal pattern at host level
(Sections IV-B, V-B)	How to label hosts based on temporal patterns?	Identify suitable labeling approach
	How stable is the temporal labeling?	Investigate fluctuation of the temporal labels of the hosts
Profile Characteristics	How to assess stability of the characteristics?	Provide a coherent methodology
Stability	What is the stability of the host characteristics?	Compare stability of different characteristics used in host profiling
(Sections IV-C, V-C)	Are there clusters of hosts with similar stability?	Identify hosts with stable behavior

The main contributions of this article are three-fold:

1) we provide an insight into long-term aspects of host behavior in a network,
2) we release a real-world dataset that describes a one-year behavior of a large variety of hosts and is open for public use and further research at [9], [10],
3) we provide suggestions for building a robust host behavior profile based on our observations from a real-world network.

Moreover, our methodology for the long-term behavioral profile evaluation can serve as an indicator of the applicability of host profiling approaches in real network management scenarios. The study also contains a description of three selected use cases that demonstrate the caveats of host profiling discussed in the paper on observed real-world host's behaviors.

The rest of the paper is organized as follows. Section II defines the objectives of the study and states research questions. The data collection methods and dataset characteristics are described in Section III. Methodology used to analyse the dataset is provided in Section IV. Results of the study are presented and discussed in Section V. Related work is reviewed in Section VI. Section VII concludes the paper.

II. Study Objectives

A dataset that spans over one year and a large network with a wide variety of hosts offer an opportunity to make observations that would be impossible to make using a dataset covering a shorter time period or smaller networks. A higher number of observations in the one-year dataset also increases the statistical significance of the executed study. For example, a validation of the week patterns in host behavior is supported by 52 observations in our dataset compared to 4 observations in the one-month dataset used in [8].

Being able to use such a dataset, we state the general objective of this study to *evaluate the characteristics of host behavior in a network from a long-term perspective*. For this study, we identify the following research areas: availability of data for host behavior profiling, temporal stability, and profile characteristics stability. These research areas cover artifacts of host behavioral profiling and are significant from a long-term perspective and profiling applicability in real-world deployments. A high-level overview of the study objectives is provided in Table I.

A. Data Availability

The availability of the data for computing of a host profile is often an ignored fact when host profiling is discussed in the literature. Missing observations of a host behavior significantly influence the profiling result, however. There are two prominent reasons for missing observations in a dataset. First, hosts in a network do not, by nature, communicate all the time they are connected to the network; a host can be turned off, hibernated, or communicate only on the local network where data are not collected. Second, data might be missing due to monitoring process downtime, e.g., because of scheduled infrastructure maintenance.

A host profile is usually an aggregation of the observed features, e.g., the volume of traffic transferred over a specific time window. The missing observations and how they are handled influence the outcome of the aggregation. For example, when automated creation of a day profile is computed as an average over hour sums of transferred bytes, it will be biased for a machine that communicates in business days only. During the weekend, this machine's profile will become zero, and the regular Monday's traffic will look like an anomaly from the perspective of the weekend profile.

Hence, our study will investigate the distribution of missing observations over the year for all hosts in the network. We expect this investigation to shed light on the availability of data in the long term and offer suggestions for what the suitable time windows and aggregation functions for host profile creations are. Moreover, we provide a discussion on how to handle missing observation when building a behavioral profile, so that the impact of the missing observations is minimized.

B. Temporal Stability

It is a commonly accepted fact that temporal patterns in network traffic can be observed on a network and its subnets [11]. The temporal patterns often follow diurnal patterns, or weekday patterns, for example. It is not surprising that these patterns can be, to some extent, also observed in the behavior of a single host in a network.

This study will validate the presence of temporal patterns in host communication habits. Having one-year data at hand, we can evaluate the presence of the temporal patterns during the year with reasonable statistical significance even when considering the weekday patterns. We will examine how many of the hosts show these temporal patterns and discuss the presence of the patterns for different host types on the observed network.

Apart from the validation of the presence of the temporal patterns in hosts' communication habits, we aim to study whether it is possible to label host profiles based on the temporal patterns observed, such as day vs. night talkers, or business vs. weekend day talkers. Temporal behavior labels for each host in a network would enable network managers to create appropriate policies; for each label type, they could, e.g., assign an individual security policy and thus improve network security. Moreover, we will apply several approaches to label assignments based on statistically significant behavior during the year. Last but not least, we will investigate the long-term stability of the assigned labels.

C. Profile Characteristics Stability

Stability of behavior profiles is an essential prerequisite for its further use for anomaly detection, network policy management, or host behavior prediction. A host profile usually consists of several characteristics such as a number of communication peers, volume of transferred traffic, or a number of used distinct ports. If these characteristics show a large variability in observed volumes in time, the behavior prediction based on these characteristics will not be accurate, or the confidence interval of the predictions will be too wide to be applied in practice.

The analysis of the profile characteristics stability can discover hosts with high variability of behavior, which would be impractical to predict, on the one hand. On the other hand, the stability analysis can identify highly-stable hosts for which it will be straightforward to predict their behavior. The stability of the volume of the characteristics can also be used for security management. Stable, and hence expected behavior, poses a lower risk to network security, while the hosts with highly volatile behavior deserve closer attention of the security officers.

Hence, we will inspect the stability of the volumes of the typical characteristics present in host behavioral profiles. We will summarize a methodology on how to assess the profile characteristics stability. Next, we will compare the stability of the different characteristics present in host profiles over one year of observations. Last, we will analyze the stability of the observed characteristics of different types of hosts present in the network and use them to identify clusters of hosts with similar stability of characteristics. We believe that our findings will help to identify general types of hosts with more stable behavior, which are suitable for anomaly detection or behavior prediction.

III. Dataset

Our study is built upon the real-world dataset collected over the year 2019 (January 1st – December 31st) [9]. All aspects of the data collection and preprocessing need to be discussed to ensure correct comprehension of the study's results. Therefore, this section describes the characteristics of the collected dataset, provides details about the network environment, where the data was collected, explains methods used for raw data retrieval, and details the extract-transform-load processes used for the dataset creation.

TABLE II
Overview of Selected Subnets

Subnet	Ranges	Description
Workstations (SUB_WORK)	/25, /24	Workstations located at the faculty used both for administration and development.
Servers (SUB_SRV)	/24, /24	Segment with the servers hosting both web services and services for network infrastructure.

A. Network Description

Raw data for the dataset was retrieved from the university network spanning a /16 IPv4 CIDR range. The network is divided into 26 administrative subnets that represent individual faculties and institutes of the university. There is no central management of the university network; the backbone of the university network and the connection to ISP are operated by the Institute of Computer Science, each faculty or institute autonomously manages its network subnet and applies its own policies. Such a distributed autonomous configuration of the network results in the fact that the behavior of the hosts can be influenced by dissimilar network management approaches. Nevertheless, the university network as a whole is, in general, an open and policy-free network compared to a business network. The behavior captured in our dataset is not bound by any strict restrictions and represents the natural behavior of the hosts in the network. The decentralization of the management of the network leads to the lack of central network asset management and, therefore, lack of effective host labeling.

As for the variety of hosts in a network, the university network represents a diverse environment. There are typical workstations used for administrative tasks, research & development workstations, and shared workstations in public PC rooms, for example. Apart from workstations, there are servers ensuring the critical functions of the network and university itself, such as DNS servers, servers hosting the information systems of the university, their databases and Web interfaces, mail servers, and servers for identity management. Apart from the critical infrastructure, there are numerous servers hosting research databases, Web presentations, or development servers. Furthermore, the university provides wireless connections for all students and several business partners as well. Last but not least, the university operates a cloud environment used for research and extensive computation tasks. However, due to the above-mentioned lack of the central network asset management, we do not have information on each host present in the network. Still, we were able to identify two sets of subnets, that include hosts with different behavior - segments with the majority of the workstations, and segments that include mainly servers, see Table II for details.

The university network is connected to the Internet Service Provider by two 40 Gbit lines. Measured at both ISP lines, the average connection rate is 6.44 k connections per second with

TABLE III
HOST-RELATED IP FLOW FEATURES

Type	Name	Aggregation
Aggregations	# of flows (FL)	src IP
	# of packets (PKT)	src IP
	# of bytes (BYT)	src IP
	Flow duration (sec) (DUR)	src IP
Distinct counts	# of peers (PEER)	src IP, dst IP
	# of ports (PORT)	src IP, dst port
	# of protocols (PROTO)	src IP, dst protocol
	# of AS numbers (AS)	src IP, dst AS number
	# of countries (CTRY)	src IP, dst country

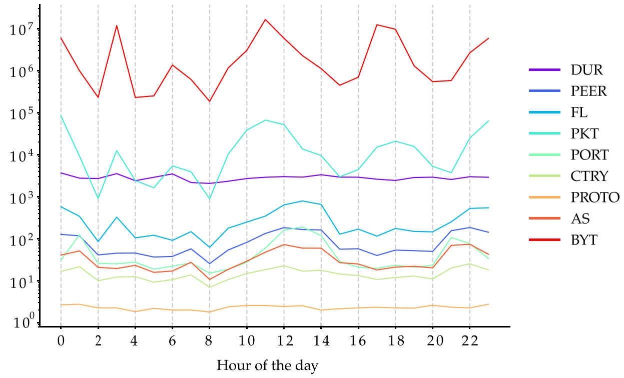


Fig. 1. Observed features for a sample host during a day time window.

a packet rate of 473.82 k packets per second, and throughput of 3.52 Gbps. For detailed characteristics of the properties of selected university network subnets, refer to [11].

B. Data Collection Process

Due to the size of the network, speed of the network, and time span of the network observation, IP flow monitoring is preferred to Deep Packet Inspection (DPI) as a method for information retrieval. IP flow monitoring was designed to monitor network traffic in large-scale high-speed networks, where DPI fails due to performance limitations. An IP flow is an abstraction of a uni- or bidirectional connection. All packets belonging to a particular IP flow have a set of common properties called flow keys [12]. The traditional 5-tuple of flow keys comprises of source and destination IP address, source and destination port, and transport protocol. Apart from the flow keys, IP flow contains several statistics about the connection (such as the number of transferred bytes and packets). Information from the application layers of network traffic has been included in IP flows recently, which further increase the visibility into the network. For a detailed description of IP flow monitoring aspects, consult [13].

The observation points for the network traffic measurement were located at the connection of the university network to its Internet service provider (ISP). As discussed above, the university network is connected by two connection points to ISP. On both connection points, we installed passive traffic access points (TAP) that transparently copy all passing-through network traffic without inducing any packet loss. Dedicated high-speed IP flow probes then processed the mirrored network traffic.

The location of the observation point at the connection points of the network to ISP implies that we observe only the ingress and egress traffic of the university. We are not able to observe intra-network traffic. There are probes that monitor the intra-network communication, e.g., communication between faculties. However, using these probes would lead to the necessity of data deduplication, as multiple probes can observe one connection between faculties. The deduplication is a labor-intensive and error-prone process. Hence, the resulting dataset could include a higher number of duplicate and noisy observations. Considering the above-stated, we chose to place the observation points at the connection points to ISP to keep the dataset clear. Moreover, the observation of the network traffic at network connection points represents a typical setting of the network monitoring infrastructure in real-world deployments.

During the IP flow metering process, no sampling was applied. The IP flows were created as single-directional using the following settings: 60 seconds for inactive timeout and 300 seconds for active timeouts. Since the storage of IP flows for the whole years would be impractical due to storage capacities, the host behavioral characteristics were computed each month, and only the preprocessed characteristics were stored.

C. Feature Description

First, we identified features that represent host behavior and are retrievable from IP flows. Moore *et al.* [14] presented a list of IP flow features that can be used for IP flow-based classification. We considered features that represent a host's behavior. The selected features are summarized in Table III. Since we are interested in the host's behavior, all features represent traffic that originates at the host.

The level of the raw data aggregation was set to one hour. Our experience shows that one-hour time window represents a sufficient aggregation that masks the natural burstiness of the network traffic while maintaining a reasonable number of observations. Hence, for each hour in a year and each host, we computed the value of the given characteristic, i.e., $Obs_{(j,k)} = (FL_{(j,k)}, PKT_{(j,k)}, \dots, CTRY_{(j,k)})$ where $j = 1, \dots, 65536$ is host identification, and $k = 1, \dots, 8760$ is an hour in a year. The aggregation features sums the given characteristic over an hour interval, e.g., $FL_{(j,k)} = \sum$ (all flows with src IP = host j in hour k of the year). The distinct count features counts unique pairs described in the Aggregation column of Table III over the given hour, e.g., $PEER_{(j,k)}$ = number of unique pairs $(\mathit{src\ IP\ of\ host\ j, dst\ IP})$ in hour k of the year. Given the combination of the number of the observed hosts (65536), the number of the observations (8760), and the number of the observed characteristics (9), the resulting datasets comprises of 5,166,858,240 observations. The observed features for a sample host during a day are illustrated in Figure 1.

D. Privacy

We are aware that the monitored data contains privacy-sensitive information, such as IP address. Hence, we declare that the monitored data used for our research were processed in accordance with the EU General Data Protection Regulation 2016/679. The monitored data were collected for specified purposes, and the appropriate technical and organizational measures were taken to safeguard the rights of the data subjects. We processed the data in a manner that ensured appropriate security of the data, including protection against unauthorized or unlawful processing, accidental loss, destruction, or damage. We implemented appropriate technical and organizational measures to ensure a level of security appropriate to the risk, including the pseudonymization and encryption of the data, assurance of confidentiality, integrity, availability, and resilience of data processing systems. The publicly available dataset is anonymized using state-of-the-art anonymization techniques so that the re-identification of the individual IP addresses is made as difficult as possible. For the anonymization of the IP addresses we used the cryptography-based prefix-preserving anonymization. Apart from the anonymized IP addresses, only the volumetric statistics are published.

IV. Methodology

This section describes a methodology used to achieve the study objectives described in Section II. We describe the methodology for each presented research area separately in the sections below.

A. Data Availability

The goal of the data availability research area is to investigate the long-term availability of the data needed for host behavioral profile computation across a large variety of hosts. The fact that there is no connection originating from the given host during the observation one hour is represented by an N/A value in the dataset. Hence, we analyze N/A values present in the dataset to explore data availability for host profile computation.

The first step of the data availability analysis is the exploration of the distribution of the N/A values over the whole dataset. We identify how many IP addresses do not communicate during the whole year; these IP addresses might represent unassigned IP addresses. Analogously, we identify observation windows during which no IP address communicated; this observation window might represent an outage of the network monitoring probe, as it is highly unlikely that no IP address from /16 network would communicate.

Having captured the number of N/A values in the dataset as a whole, we examine the distribution of the N/A values for the individual hosts. For each host, we compute the total number of N/A values and show the distribution of the host's total numbers in the whole dataset. We expect that the distribution of total N/A values for each host in the dataset follows the behavior of the hosts, e.g., host communicating only during the working hours. The peaks in the distribution should mean that a typical behavioral pattern is present at a significant number of hosts. To identify these typical behavior patterns, we compute the number of observations for typical host behavior and compare them with the host's N/A values distribution.

Next, we want to explore the impact of the level of the aggregation of the data used for the profile computation on the number of the N/A values observed. Hence, we define multiple aggregation levels common for profile computation: 1 hour (1H - the original aggregation), 2 hours (2H), 4 hours (4H), 8 hours (8H), half a day (12H), one day (1D), and one week (7D). For each of these aggregation levels, we compute the distribution of the N/A values and compare basic distribution statistics. We expect that the higher aggregation level would imply a lower number of the N/A values on average. We also investigate the impact of the aggregation on different subnets.

TABLE IV
Approaches to N/A Values Handling

Approach	Description
None	N/A values are not handled.
Replace	Replaces N/A values with a selected value, e.g., 0.
Padding	Replaces N/A values with the last previous non-NA value.
Interpolation	Replaces N/A values with a interpolation between the closest non-N/A values. The interpolation can be linear, quadratic, polynomial, for example.

TABLE V
Used Averaging Methods

Averaging Method	Description	Pros (✔) & Cons (✘)
Std-Mean	Standard implementation of mean, N/A values are replaced by 0's	✔ Commonly used ✔ Implemented in a majority of libraries ✘ Decreases mean ✘ Exaggerates standard deviation
Mean (excl N/A)	N/A values are excluded from the calculation	✔ Does not skew the mean by N/A values ✘ A custom implementation is needed
Median	Midpoint value in a frequency distribution	✔ Eliminates the impact of the extreme values ✘ Does not provide standard deviation
Most Frequent Value	The most frequent value observed	✔ Represents the majority in the observations ✘ Does not provide standard deviation

Last but not least, we provide a comparison of the different approaches to the handling of N/A values. We investigate the substitution of N/A values for 0 and different interpolations of the N/A values from the surrounding values, see Table IV. Next, we apply mean aggregation over the data with handled N/A values to demonstrate the impact of the selected method for N/A handling. We expect that different approaches to N/A values handling will have an impact on the resulting profile characteristics. Based on the results, we discuss the pros and cons of the evaluated approaches to N/A values handling.

Apart from the N/A values handling, there are also several ways of computing the mean value of characteristics concerning N/A values. Table V introduces four approaches that can be used to compute means. The taxonomy introduced in the table is used in the rest of the paper.

B. Temporal Stability

The main goal of the temporal stability research area is to explore the temporal patterns present in the dataset. We explore the presence of the temporal behavior pattern at the host level and investigate whether it is possible to label the hosts based on the temporal pattern. The stability of the labeling, i.e., the stability of the temporal patterns in host behavior, is investigated as well.

In our study, we choose to explore the following temporal behavior patterns: (1) diurnal pattern that should be significant for human-operated hosts or servers hosting services used mostly by humans from a similar timezone, (2) weekday pattern where we should be able to differentiate between business days and weekend.

To study the diurnal pattern, we aggregate the hosts' behavior over individual hours in a day and observe the frequency of communication over the year. If a host communicated in a given hour, we set the observation to 1, 0 otherwise. We sum the observations for the given hour over the year and compute the share of the hours in which the host communicates to the total number of the particular hours in the year (365). For example, we can find that a host communicated in the interval 8:00 — 8:59 in 40% days in a year. We do this for all hours in a day to get a day profile for each host. The day profile enables us the discover the diurnal patterns. We inspect the day profile distribution in the SUB_WORK and SUB_SRV subnets. The weekday pattern is explored analogously; only instead of the sum over an hour, we aggregate the data over days in a week.

Next, we inspect the labeling of the hosts based on their temporal patterns. Based on the previous analysis, we define the following categories of labels: day talker, night talker, business day talker, and weekend talker. From these basic labels, we can derive additional labels, such as day talker only, i.e., a host that is a day talker and not a night talker. We expect that these labels will be assigned based on the prevailing patterns in the host behavior. However, the threshold values for these observations need to be determined. For example, on average, how many of the night-hours in a day does a host need to communicate at minimum to be labeled as a night talker? Or, how many hours are needed to make a significant difference between a day talker and a night talker? To answer these questions and determine the relevant threshold, we investigate the distribution of the labels under different thresholds. Based on this analysis, we identify a suitable method and thresholds for labeling the hosts in a network.

Last, we use the identified method and threshold setting for host labeling to investigate the stability of such labels in time. We split the dataset into training and testing dataset. The training dataset covers the first six months of the year, while the testing dataset contains data from two months from the second half of the year (September and October). We compute host labels in both datasets, compare the host labels from the testing dataset with the host labels from the training dataset, and compute the ratio of change.

C. Profile Characteristics Stability

The main goal of the profile characteristics stability research area is to analyze how the volume characteristics are frequently used to compute a host profile behavior in time. Specifically, we aim to assess the stability of the characteristics presented in Table III in time. Before we proceed to the description of the methodology related to this research area, we need to define how we assess the stability of the profile characteristics in time.

The host characteristics present in the dataset are time series by nature. The notion of stability of a time series in time is usually represented as *stationarity* of the time series. Simply stated, stationarity means that the statistical properties of a process that generates a time series do not change over time. As a result, the parameters of the process generation, such as mean and standard deviation of time series, remain the same over time. The stationarity requires shift-invariance and equally distant observations. However, we expect that the time series representing host behavior contains a significant portion of missing observations, and the time distances between individual non-N/A values are not equal. Hence, the stationarity is unsuitable as a measure of stability in this case.

Still, despite the missing observations, we can investigate the statistical parameters of the time series, such as mean and standard deviation. Since we explore the volume characteristics of the host profile, we expect significant differences in the characteristics among different hosts in the dataset, e.g., the volume of transferred bytes by a server is usually higher by orders than the volume of transferred bytes by a workstation. However, the variability of the characteristics can be similar. To be able to compare the variability of the characteristics of hosts with different volume scales of the characteristics, we use the coefficient of variation metrics.

The coefficient of variation (c_v) is a standardized measure of variability. It is defined as the ratio of the standard deviation to the mean

$$c_v = \frac{\sigma}{\mu} \tag{1}$$

and shows the extent of variability in relation to the mean of the population. Hence, the coefficient of variation allows us to compare the mean and standard deviation of hosts with different absolute values of the computed behavioral characteristics. We compute the coefficient of variation for all hosts in a network, compare the distribution of the coefficients of variation over the whole dataset, and elaborate on the stability of the host behavior. An analogous investigation will be executed for the selected subnets to identify differences in variability among different types of hosts.

Last, we apply the density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm to identify how many clusters with similar variability of the time series there are. DBSCAN clustering identifies clusters as areas with a high density of objects separated by areas with a low density of the objects in its distance-based neighborhood. The main advantage of the DBSCAN algorithm is that the number of clusters does not have to be specified. The optimal number of the clusters is identified automatically based on the minimum number of points in the neighborhood parameter. Further, DBSCAN can identify arbitrarily shaped clusters compared to the K-Means that creates only the convex clusters. Finally, DBSCAN is robust to the outliers. The disadvantages of the algorithm are the lack of determinism, its heavy dependence of

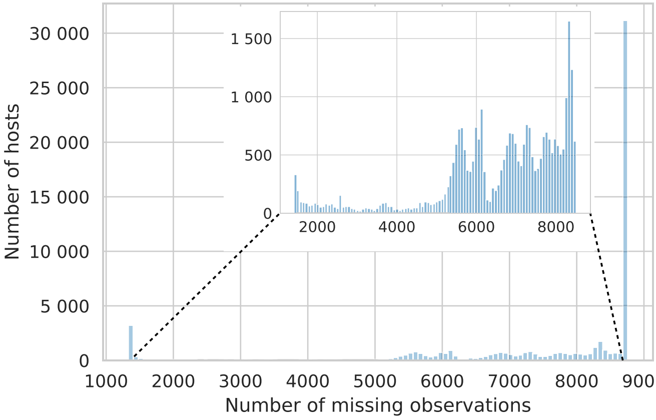


Fig. 2. Distribution of N/A values per host over the dataset.

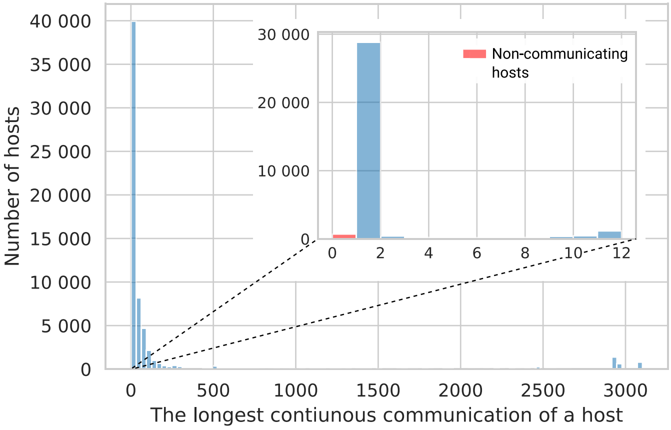


Fig. 3. Distribution of the longest uninterrupted communication of a host.

the chosen distance measure, and low performance on datasets with significant differences in densities. Based on the characteristics of the discovered clusters, we explore the main types of variability present in host behavior. For the evaluation of the clustering performance, we employ the Silhouette coefficient. The Silhouette coefficient compares the mean intra-cluster distance and the mean nearest-cluster distance for each sample. The coefficient ranges from -1 to 1, where 1 is the best value and -1 the worst. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar than the assigned one.

V. Results and Discussion

Having described the methodology for our study, we present the results in this section. The results are presented by the research areas, and they provide answers to the research questions presented in Table I.

A. Data Availability

The main goal of the data availability research area is to analyze the distribution of the N/A observations in the dataset. The primary exploration of the dataset shows that there are 1329 observation time slots (15.171%) with N/A values at all hosts. After an investigation of logs from monitoring infrastructure, we discovered that these missing observation time slots were caused by misconfiguration of the monitoring infrastructure that resulted in the fact that no data were collected. This outage period covers the second half of August and the first half of September data. As for the hosts with N/A values for all observations, i.e., nonactive hosts, there were 703 hosts (1.073%) that did not communicate at all.

Next, we compute the total number of N/A values for each host in the dataset and explore the distribution of the total numbers of N/A values per host in the dataset. The distribution is depicted in Figure 2. Given the quantiles of the distribution of N/A values ($q_{25} = 6747$, $q_{50} = 8509$, $q_{75} = 8750$), we can see that majority of hosts communicated in less than 22.98% of observations in the whole year. The increasing frequency of the hosts having more than 5500 missing observations in a year can be caused by the presence of a higher portion of the hosts that communicate only during business days working hours. For these hosts, the number of the missing observations in a year should theoretically be 5628 (number of business days in the year × 12h). A key takeaway message is that considering a typical workstation-like behavioral profile, more than 64.24% observation can be missing given a one-hour observation window, and the methods for host-profile computation need to account for such a volume of missing observations.

The large share of the hosts with a high number of the missing observations will not hinder a behavior profile computation if a host communicates continuously over an extended period of time. For example, it is more feasible to compute a behavioral profile for a host that communicates continuously over only two months in a year than for a host that communicates sporadically in random hours over the whole year, even when the number of missing observations would be the same. Hence, we compute the length (in the number of observations) of the longest continuous communication for all hosts in the dataset.

The distribution of the lengths is depicted in Figure 3. We observe that there are approximately 45% of hosts with the longest continuous communication length lower or equal to 1 hour. For these hosts, it would be difficult to compute a behavioral profile. The median of the length of the longer sequence is 12 hours, which represents workstation-like behavior. In the zoomed-in segment, we can observe that number of hosts with the longest communication length equal to zero is the same as the number of hosts that did not communicate at all (703), which confirms our observation on nonactive hosts number described above. The quantiles for the longest communication lengths are as follows: $q_{25} = 1$, $q_{50} = 12$, $q_{75} = 68$, $q_{90} = 511$, and $q_{95} = 2401$.

The next analysis focuses on the impact of the data aggregation on the number of N/A values observed. To be able to compare the volume of missing observations for different aggregations (i.e., different total numbers of observations), instead of the absolute number of the N/A observations, we compute the share of the N/A values to the total number of observations. The share of the N/A values with different aggregations are computed for each host. The means of the shares over the whole dataset and the selected subnets are presented in Figure 4.

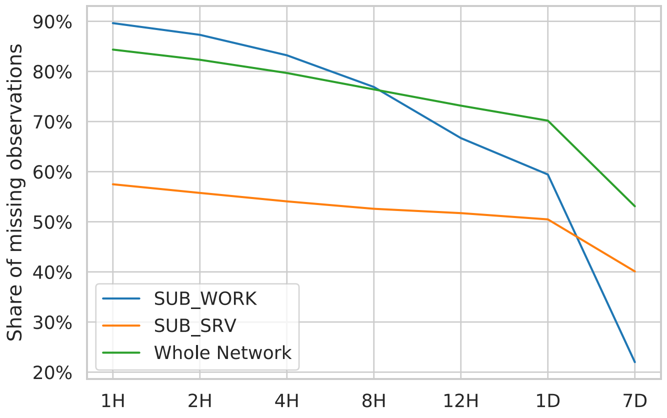
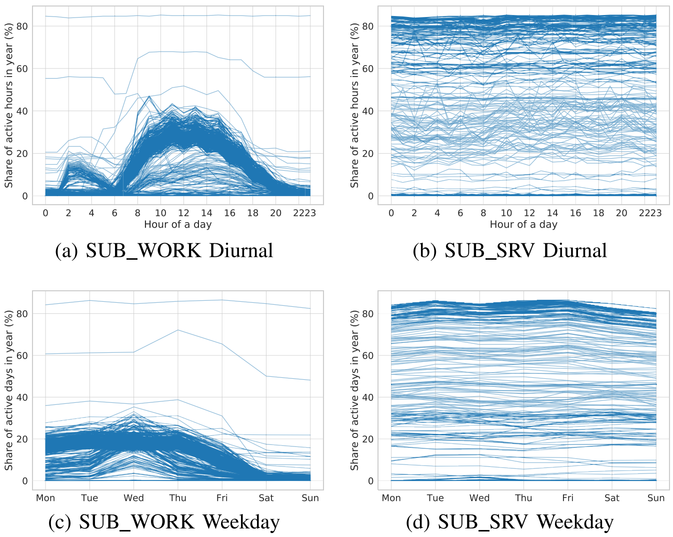


Fig. 4. Impact of aggregation level on the mean share of N/A values.



(a) SUB_WORK Diurnal

(b) SUB_SRV Diurnal

(c) SUB_WORK Weekday

(d) SUB_SRV Weekday

Fig. 5. Temporal patterns.

As expected, with the increasing level of the aggregation, the share of the N/A values per host is decreasing. There is a significant drop for the aggregation level 1 day (1D). For the server-like behavior, the higher level of aggregation reduces the share of the N/A observations insignificantly, only by a few percentage points. However, the impact of the aggregation on the number of the N/A values is significant for the workstation segment. From 89.63% of the missing values on average at the aggregation level 1H, the share drops to 66.69% of mean N/A values per host at aggregation level 12H. The rate of decrease of the share values increases for aggregation levels higher than 4H. Hence, if no higher granularity is explicitly requested, we recommend utilizing the 4H or 1D aggregations for computing workstation's behavior profiles to optimize the number of the N/A observations in the dataset.

Given the large number of N/A observations present in data, which we demonstrated in the analyses above, the selection of the method for handling N/A values plays a significant role in host behavioral profile computation. We compare the following four methods for N/A value handling, as presented in Table IV in Section IV-A: none, replace (by zero), padding, and interpolation (linear). We use these methods to remove N/A values from the dataset.

To demonstrate the impact of the different N/A handling methods, we compute a mean volume of transferred flows by the hosts. For the mean computation, we used standard implementation in the NumPy library (Std-Mean). Leaving N/A observation unchanged (none approach, see Table IV) resulted in a mean equal to 2.742×10^2, while the replace method resulted in a mean equal to 1.034×10^2. Both methods seem reasonable; one leaves the handling N/A values to the implementation of the mean algorithm, while the other handles the N/A values by itself. For profile computation, we suggest using the latter way of handling N/A values if we are not sure how the aggregation functions are implemented. The padding and interpolation methods had the mean between these values (2.383×10^2 and 1.814×10^2). We do not suggest using these methods computation profiles of hosts that are expected to show a scatter communication patterns as these methods could smooth the gaps in communications that are important for host behavior profiling.

B. Temporal Stability

The main goal of the temporal stability research area is to shed light on temporal patterns present in the host behavior. We first explore the temporal patterns present in the whole dataset. We compute the number of communicating hosts for (1) each hour in a day, (2) each weekday, and (3) month in a year.

We identify a strong diurnal pattern present in the dataset, where the lowest host activity is at 4 AM and the highest activity at noon. At the peak, there is a more than 50% increase in the number of active hosts compared to the hour with the lowest host activity. As for the weekend day, we confirm the business-day and weekend pattern, with decreasing activity on Friday. The activity during the year correlates with the schedule of the academic year. The lower activity in July and August represents holidays, the peak in September start of the academic year.

Further, we inspect the temporal patterns for selected subnets. The patterns for each of the subnets are presented in Figure 5. A line in the figure represents a share of the active observations in a year for a single host. The diurnal pattern with the peak at noon and a smaller peak at 3 AM is present at the SUB_WORK segment. The peak culminating at noon represents the typical daylight activity. The smaller peak at 3 AM is caused by the updates of the workstations planned by the central management system. Similarly, the weekday pattern is observable at the SUB_WORK, which reflects the fact that the majority of the hosts in the SUB_WORK subnets are used by the employees of the university. Hosts in the SUB_SRV segment, on the other hand, do not show any significant diurnal pattern.

Next, we explore how to label hosts based on temporal patterns present in their behavior. As mentioned in Section IV-B, we identify the following patterns summarized in Table VI. We derive the *day period* based on the diurnal pattern observed for SUB_WORK in Figure 5. The labels are not exclusive, i.e., a host labeled as a *day talker* can also be labeled as a *night talker*. To capture the exclusivity, we suggest to derive the *day*

TABLE VI
LABELS OVERVIEW

Label	Description	Threshold
Day talker	A host communicating during a day. The day period starts at 6 AM and ends at 6 PM.	Average number of active hours during day period. **Suggested threshold: Active hours: 6**
Night talker	A host communicating during night. The night period includes 12 AM - 6 AM and 6 PM - 12 PM of a day.	Average number of active hours during night period. **Suggested threshold: Active hours: 5**
Business day talker	A host communicating during business days.	Average number of active hours during business days and average number of active days during business days. **Suggested thresholds: Active hours: 8 Active days: any{1...5} (no impact)**
Weekend talker	A host communicating during weekend days.	Average number of active hours during weekend and average number of active days during weekend. **Suggested thresholds: Active hours: 8 Active days: 1**

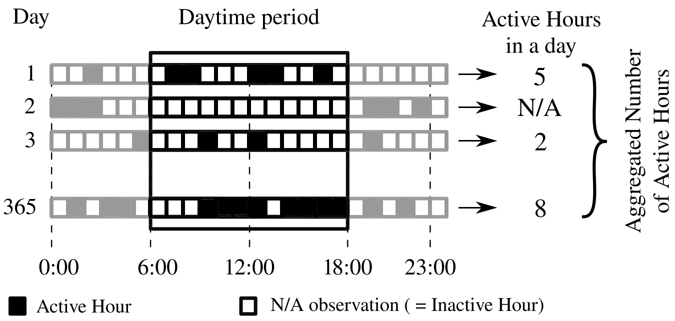


Fig. 6. Computation of the average number of active hours for a single host for *day talker* labeling.

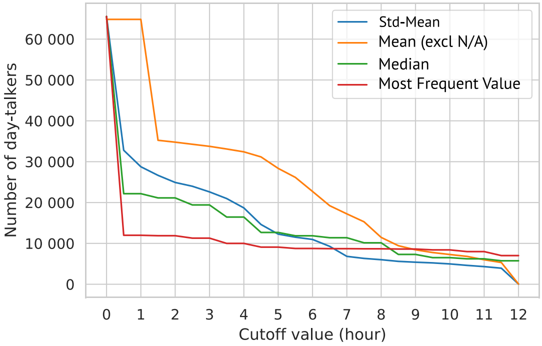


Fig. 7. Cutoff analysis for *day talker* label assignment.

talker only label as a host that is labeled as a *day talker* and is not labeled as a *night talker*. The *night talker only*, *business day talker only*, and *weekend only* labels can be defined analogously. Using this approach, we can also define an *all day talker* label as a host that is labeled both as a *day talker* and a *night talker* at the same time. Analogous, *all week talker* label is defined as a host that is labeled as both *business day talker* and *weekend talker*.

As described in the Methodology section, we need to identify the threshold for assigning the hosts with labels. A host can be labeled as a day talker if it communicates more than x hours on average during the day period (as defined in Table VI). First, we computed the *averaged number* of active hours during the day for each host. The computation of an *averaged number* of active hours for a single host is depicted in the Figure 6. The result of the final aggregation of active hours in a day is influenced by how the aggregation methods handle the N/A values (Day 2 in Figure 6 where the host did not communicate during the daytime period). As we have shown in the data availability analysis, the N/A values are present frequently in the dataset. Hence, we need to select a suitable approach that would reduce the differences introduced by a different approach of the aggregation methods to N/A values. To illustrate the differences introduced by the aggregation functions, we compute the averaged number of active hours for the example presented in Figure 6. The averaged number is 3.75 $(= 15/4)$ using a standard mean function and 5 $(= 15/3)$ using a function that omits N/A values, making a difference of 1.25 in the aggregated number of active hours.

To avoid the bias, we use four different methods for computation of the *averaged number* of active hours for a host: standard implementation of the mean function (Mean-Std), a custom implementation where N/A values are not included in the mean (Mean (excl N/A)), Median, and Most Frequent Value function (see Table V). We use all these functions to compute the average numbers of active hours during the daytime period (four average numbers in total) for each host in the dataset.

Once we computed the *averaged numbers* of active hours during the day for each host, we compared the number of hosts labeled as day talkers based on the different cutoff values x. The relation of the number of the hosts labeled as day talkers and the cutoff value x (i.e., the minimum number of hours communicated on average during the day period) is depicted in Figure 7. The number of hosts labeled as day talker decreases as the number of active hours needed to label a host increases. However, for the number of active hours greater than six, the number of the labeled host remains nearly constant (except for the Mean-Custom). Hence, setting the threshold higher than six makes no significant difference for labeling. We did the analogous analysis for the night talker label, and the threshold beyond which there is no difference in the number of labeled hosts is five hours. We selected these cutoff values for assigning labels to hosts. To decrease the impact of the implementation of the aggregation method used for the computation of the average number of active hours, we used all four averaging methods with a given cutoff. The day/night talker labels were assigned based on the majority voting principle (3 methods (out of 4) suggesting day/night talker label were needed to assign the given label).

For the business day/weekend labeling, we combine the averaged number of active hours during business day/weekend

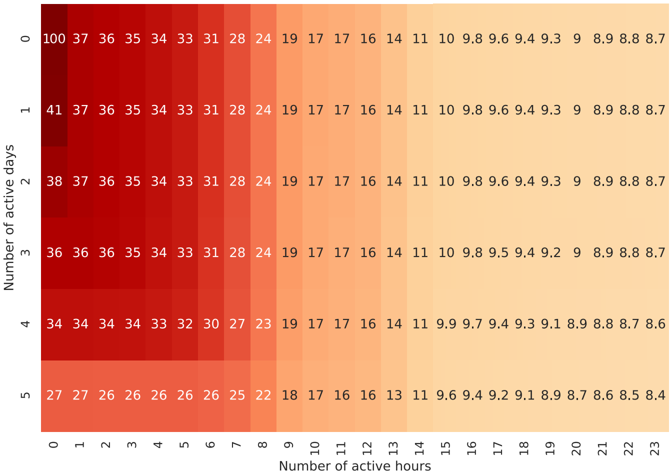


Fig. 8. Cutoff analysis for *business day talker* label assignment (Median, share of hosts labeled as business day talkers).

and the averaged number of active days during the business day/weekend indicators - see Table VI. Analogous to day/night talker labeling, we executed the cutoff analysis for both the labeling indicators and each of the aggregation approaches. A host was labeled as *business day talker* when its averaged number of active hours was greater or equal to the active hours cutoff value and its averaged number of active days was greater or equal to the active days cutoff value. The result for the aggregation method Median is depicted as a heatmap in Figure 8.

The heatmap shows the share of the hosts labeled as business day talker based on the different combinations of active hours and active days set as the threshold for labeling. The darker is the heatmap color, the higher is the share of the hosts labeled as *business day talker*. The change in the color of annotation highlights the significant change in the share of the labeled hosts. Looking at the hour (vertical) dimension of the heatmap, we observe that there is a significant drop in the share of the business day talkers (approx. 5%) between the 8th and 9th hour. The difference in the shares of labeled hosts decreases as cutoff hour increases. For cutoff hours greater than 14, the share remains almost unchanged. On the day (horizontal) dimension, a significant drop is present between the cutoffs of 4 days and of 5 days, where the share of the labeled hosts decreases by 7% for the hour 0. For hour cutoff greater than one, the impact of the day cutoffs between 0 and 4 makes no significant difference in the shares of the labeled hosts. In general, the average number of the active business days does not play a significant role for labeling the hosts as business day talker, except if we want the business day talkers to communicate all business day explicitly, which we do not recommend due to the earlier presented high share of N/A values present in the data used for host profile computations. As a suitable cutoff value for hour label, we recommend the 8-hour cutoff as this setting represents the most significant difference in the labeled volumes. The analogous analysis was done for weekend talker labeling with the optimal cutoff settings being 8 for active hours and 1 for active days. Suggested thresholds for the individual labels are summarized in bold in Table VI.

TABLE VII
LABEL STABILITY

Label	**Value**	**Unchanged (%)**	**Changed (%)**
Day talker	True	74.87	25.13
	False	98.49	1.51
Night talker	True	79.21	20.79
	False	98.52	1.48
Business day talker	True	73.99	26.01
	False	94.80	5.20
Weekend talker	True	67.31	32.69
	False	98.40	1.60

Last, we evaluated the stability of the labeling in time. We assign the labels to host from both training and testing datasets (see Section IV-B for datasets definition) using the cutoff setting suggested in the previous section. We compare the labels of hosts in the training and testing dataset. The results are presented in the Table VII. The True labels remain unchanged from 67% to 79% while the False labels remain unchanged for more than 94% of hosts. The higher stability of the False labels is explained by the higher share of the False labels in the training dataset. Nevertheless, these results demonstrate the fact that the host temporal patterns are changing in time, and it is necessary to take this behavior drift into account when profiling hosts in a network.

C. Profile Characteristics Stability

To evaluate the stability of the characteristics used for profile computation, we employ the coefficient of variance, as defined in Section IV-C. As the first step, we compute the coefficient of variance for all characteristics and all hosts in the dataset without any additional data aggregation. The summary statistics of the computed coefficients are presented in Table VIII, column No Profiling. The mean of the characteristics (μ) that is needed to compute the coefficient of variance of a host (see Eq. (1) for the coefficient of variance formula) is computed using the Mean (excl N/A) function. The mean (μ_{c_v}) of the coefficients of variance shows the average variability for a given characteristic over all hosts, and the standard deviation (σ_{c_v}) demonstrates how the variability for a given characteristic varies among the hosts in the whole dataset.

We observe the highest mean at the Flow Duration (DUR) characteristics followed by the bytes (BYT) and packets (PKT) characteristics, where the average standard deviation of the characteristics is more than two times the mean value of the series. The variability of these characteristics also varies significantly among the individual hosts. The protocols (PROTO) and AS number (AS) characteristics show the lowest average variance, which is given by the limited value set these characteristics.

Next, we show how we can improve the stability of the characteristics by applying the results of the temporal stability analysis. The analysis of the temporal stability demonstrated the presence of temporal patterns in the hosts' behaviors. These patterns significantly influence the variability of the

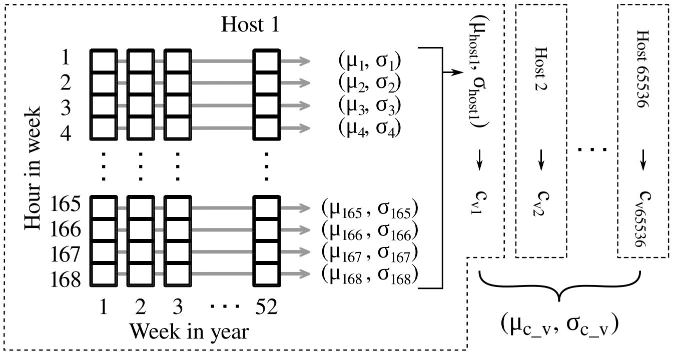


Fig. 9. Computation of μ_{c_v} and σ_{c_v} using week profiling for a selected host's characteristics.

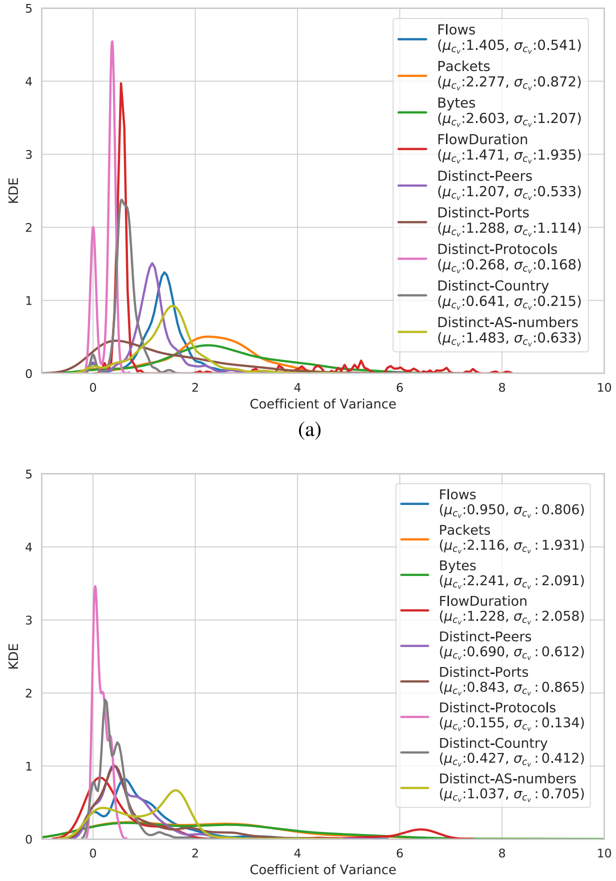
TABLE VIII
DISTRIBUTION OF THE COEFFICIENT OF VARIANCE OVER THE WHOLE DATASET

Characteristics	No Profiling		Week Profiling	
	μ_{c_v}	σ_{c_v}	μ_{c_v}	σ_{c_v}
# of flows (FL)	1.140	1.684	0.990	0.721
# of packets (PKT)	2.688	4.526	1.571	1.154
# of bytes (BYT)	4.148	6.451	1.838	1.375
Flow duration (DUR)	11.159	21.272	1.710	2.301
# of peers (PEER)	0.974	1.515	0.831	0.629
# of ports (PORT)	1.516	2.697	0.908	0.962
# of protocols (PROTO)	0.142	0.158	0.181	0.146
# of AS numbers (AS)	0.359	0.405	0.442	0.307
# of countries (CTRY)	0.966	1.479	0.949	0.881

characteristics. To eliminate the impact of the temporal patterns, we define a host profile as a set of 168 consecutive hours (i.e., all hours in one week). Hence, there are 52 observations for each hour in a week in the dataset that form time series (as a year has 52 weeks). Using this profile, we compute the overall characteristics of the distribution of the coefficient of variance values μ_{c_v}, σ_{c_v} in the dataset. We compare these characteristics with the case when no profiling is applied. The computation of the mean and standard deviation of the distribution of the coefficient of variance values μ_{c_v}, σ_{c_v} is depicted the Figure 9.

First, we compute the mean and standard deviation for each of these 168 time series in a host profile (μ_i, σ_i), $i = 1 \cdots 168$, using Mean (excl N/A) aggregation function. Next, we average these 168 means and standard deviations to obtain an average mean $\mu_{host\ i}$ and average standard deviation $\sigma_{host\ i}$ for a host i. The average is computed using Mean (excl N/A) function. These computed values are than used to obtain coefficient of variance for a host $c_{v\ i}$ using Eq. (1). The characteristics of the distribution of coefficients of variance μ_{c_v}, σ_{c_v} are then computed as an average of $c_{v\ i}$ over all hosts (i.e., for $i = 1, \ldots, 65536$). The resulting means and standard deviations of the distribution of coefficients of variance μ_{c_v}, σ_{c_v} based on week profiling are presented in the Table VIII, column Week Profiling. The profiles that take into account the temporal patterns show significantly lower variability of the profile characteristics compared to the characteristics with no profiling. All coefficients are close to one, which means that the variability is nearly equal to the mean value. Also, the variability of the characteristics differs less across individual hosts compared to the characteristics with no temporal profiling applied.

Next, we inspect the stability of the profile characteristics among different types of hosts. We apply the week profiling described above and compute the coefficients of variance for all hosts in the selected server and workstation subnets. We estimate the distribution of the coefficients of variance for the selected SUb_WORK and SUB_SRV subnets using the Gaussian Kernel Density Estimate (KDE) method. The resulting distributions are presented in Figure 10. The higher are the peaks in the distribution, the more frequent the value of the coefficient of variance is.



(a)

(b)

Fig. 10. Distribution of the coefficients of variance and their descriptive characteristics in (a) SUB_WORK and (b) SUB_SRV segments (estimated using Gaussian Kernel Density Estimate (KDE)).

From the shapes of the distributions and associated μ_{c_v} and σ_{c_v} values, we observe that variability of the host profile characteristics is lower in the subnets with server-like hosts compared to the subnets with workstations (the peaks in distribution, i.e., most frequent values of the coefficients of variance, are closer to zero). The coefficients of the characteristics with expected high variability, i.e., the number of bytes and packets, remain almost the same for both subnets. In contrast, the SUB_WORK subnet shows higher coefficients of variability at the number of distinct peers and distinct ports used. This difference reflects the fact that servers usually communicate with a large number of hosts that remain more or less stable on average, while the usage of the workstations

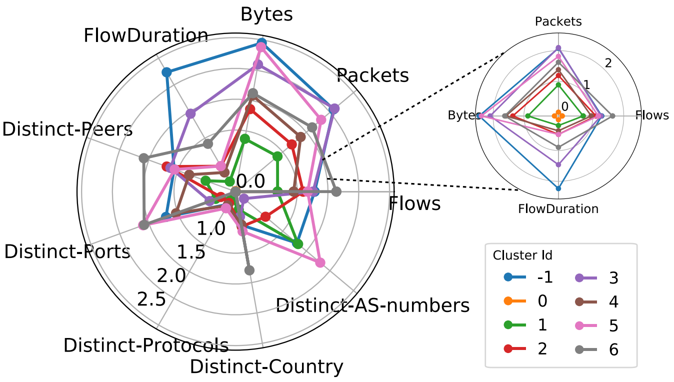


Fig. 11. Representation of clusters of hosts with similar variability of behavior characteristics.

usually depends on one person whose behavior is more random. However, the level of variability σ_{c_v} among the hosts in the SUB_SRV segment is higher than in the SUB_WORK segment, which can be caused by significantly different levels of the usage of the servers in the SUB_SRV segment.

Last, we explore the presence of the clusters of hosts with similar variability of the profile characteristics. We apply the DBSCAN clustering algorithm on computed coefficients of variance of the host profile characteristics, i.e., vectors $(c^1_{v_i}, c^2_{v_i}, \ldots, c^9_{v_i})$ representing the coefficients of variance of the nine computed profile characteristics for hosts $i = 1, \ldots, 65536$ are the inputs for clustering. For determination of the similarity (i.e., nearest neighbors), we use the Euclidean distance function. This setting should group the hosts with a similar combination of the coefficients of the variance values into one cluster.

For the identification of the clusters, we need to identify the DBSCAN's hyperparameters ϵ and *min_points* first. The parameter ϵ sets the maximum distance between two samples to be considered in the same cluster. The parameter *min_points* defines a minimum number of samples in a ϵ-neighborhood of the point to be considered for a cluster. To reduce the number of unassigned hosts and keep the number of clusters at a reasonable level, we select *min_points* $= 100$. The hyperparameter ϵ was estimated based on elbow analysis. The elbow analysis computes the distance of n-nearest neighbors for all data points. Since we set *min_points* $= 100$, we set $n = 101$ as the point itself is included in the n-nearest neighbor search. The computed distances to 101th nearest neighbor are then ordered and plotted. The ϵ is determined at the distance where the rate of the change in the shorted distances increases significantly.

Based on our analysis, we set the DBSCAN hyperparameters to *min_points* $= 100$ and $\epsilon = 0.3$. With these settings, the clustering algorithm identifies seven different clusters. The Silhouette coefficient for these clusters is 0.543, which indicates relatively well-separated clusters. The average values of the profile characteristics of the individual clusters are depicted in the radar chart in Figure 11.

The majority of the hosts are assigned to cluster 0. Clusters 1, 3, 4, 5 contain more than 1000 hosts. The rest of the clusters include about 300 hosts. Cluster -1 represents noise hosts that were not assigned to any cluster. The shape of the clusters is similar, which means that there is a similar ratio of the variance between the individual characteristics. The significant differences between the clusters are in the value of the coefficient of variability. Cluster 0 is the closest to the center of the polar chart and represents hosts with the lowest variability in behavior. The hosts from other clusters show increased variability in their behavior. Looking closely at the most frequently used characteristics for host profiling – the number of flows, packets, bytes, and flow duration, the difference of the variability between cluster 0 and the other clusters is even more significant. Further, the higher variability in numbers of bytes correlates with the higher variability in the numbers of packets. Applied to a real-world use case, these hosts should be put under closer security monitoring as their behavior in time is more volatile than the average. Moreover, the anomaly detection methods or behavior prediction of the hosts from the clusters with higher profile characteristics variability will show a higher number of errors. Having this information enables the network managers to handle the detection and anomaly alerts more efficiently.

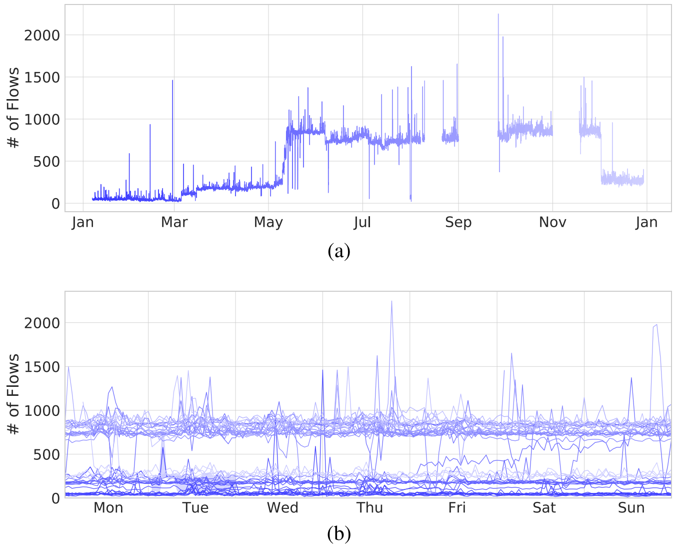
D. Use Cases

The previous section provides a more in-depth insight into the data used for the host profile computation. We discussed the artifacts that need to be taken into account when creating a host profile, such as missing observations, temporal patterns, and the stability of the characteristics used for the profile computation.

In this section, we present three selected use cases of three hosts from our dataset. The use cases aim to demonstrate the rationale behind our analyses, its relevance for a host behavioral modeling, and the already-mentioned necessity to take long-term behavior into account. We also show how the host profiling can be leveraged in cybersecurity operations. The selected use cases are the following: Use case 1: Volume change, Use case 2: Temporal pattern change, and Use case 3: Suspicious activity.

Use case 1: Volume change represents a situation where the host's behavior significantly changes in the year in the volumes of the observed characteristics. The host 133.250.163.107 presented in this use case represents a server hosting a login page for an H2020 project Web presentation. The period of the increased traffic refers to the final stage of the H2020 project, where the administrators frequently accessed the Web pages to upload new deliverables, share information, and disseminate project activities. Moreover, the login form was probably embedded into the Web pages, which further increased the traffic. Once the H2020 project had terminated, the traffic volume to the login page has decreased significantly.

Use case 1 is depicted in the Figure 12. The subfigure (a) represents the host's behavior over the year. We observe a slight increase in the number of connections during March, a significant increase in the number of the connection in May, and a decrease in December's number of connections. Each week in the plot has its shade of the blue where weeks at the beginning of the year are the darkest, and the blue goes brighter as the weeks are getting closer to the end of the year. This color scheme holds for the subfigure (b), where we plot



(a)

(b)

Fig. 12. Use case 1: Volume change *(host 133.250.163.107)*, where (a) represents behavior over the year, and (b) computed week profiles.

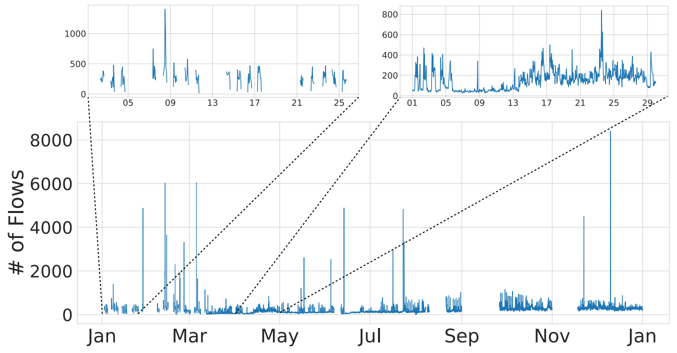
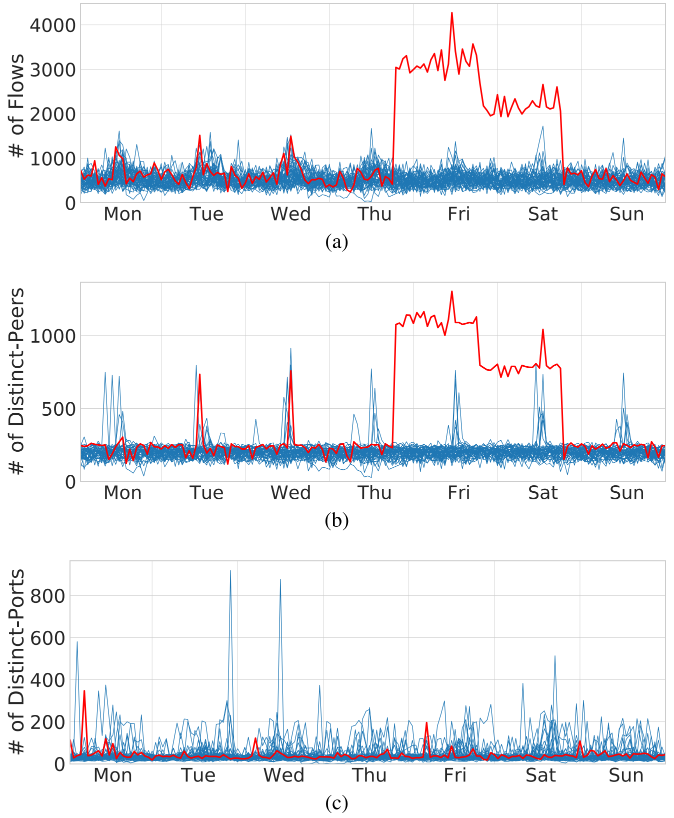


Fig. 13. Use case 2: Temporal pattern change *(host 133.250.174.37)*.

the week profiles computed for a given host so that the colors of the individual weeks match in both subfigures. The subfigure (b) allows us to demonstrate the host profile in the different parts of the year. We observe two volume levels; one representing the start and the end of the year, and the other representing the middle of the year. This use case demonstrates that the host profiles need to be continuously updated to reflect changes in the hosts' behavior. Computing the profile only at the beginning of the year would cause it not to fit the host profile for the majority of the second half of the year.

Use case 2: Temporal pattern change represents a situation where the host changes its temporal behavior during the year. The host 133.250.174.37 represents a regular workstation used when its owner was on-site at the university. However, during the year, the owner of the workstation needed to work remotely. Hence, remote access tools were deployed, and the workstation became an asset with 24/7 operation. The change happened on March 13th, 2019.

We demonstrate in Figure 13 how this change reflects in the data observed from the network telemetry used for host profile computation. Before the change point, the data would include a high portion of N/A observations during the night, rendering the host as *day talker* and *business day talker*. The variability of the data is relatively stable following the diurnal and weekdays patterns. After the change, the number of N/A observation drastically reduces, and the data variability increases. The increase of the observed data variability can be explained by the deployed tools for the remote access that generate additional traffic and keep the connection alive. Moreover, remote access to the workstation's secondary memory or the necessity to transfer data for local processing further increases the data's variability.



(a)

(b)

(c)

Fig. 14. Use case 3: Suspicious activity in week profile *(host 133.250.178.62)*, (a) number of flows, (b) number of distinct peers, (c) number of distinct ports.

Use case 3: Suspicious activity illustrates how host profiling can assist in cybersecurity operations. The host profiles can be used to determine abnormal behaviors of hosts in a monitored network. Using the host profiling, we identified the host 133.250.178.62, whose network traffic showed abnormal activity that significantly differed from its normal behavior profile. The host represents a management machine in a cloud used to control deployment in a cloud.

The observed anomaly, along with the host's usual week profile, is depicted in the Figure 14. We observed the anomaly for the number of flows (FL), packets (PKT), bytes (BYT), or the number of distinct peers (PEERS). However, the deviation was not observed in the number of distinct ports characteristic (PORT). Such anomalous observation represents a situation when the host contacts a significantly higher number of hosts than usual (high values of FL and PEER characteristics), asking for a usual number of services (non-anomalous number of distinct ports). This behavior can represent a scanning activity of the host for a single service across multiple machines. From the security point of view, such a suspicious activity can mean an infection of the host with a malware that is now scanning for possible targets to attack.

It is important to note that such an anomaly is neglectable from the whole network point of view (in terms of volume) and would probably not be detected. However, from the host point of view, the anomaly is significant and easily detectable thanks to the computation of the host behavior profile. The host profiles enable us to monitor the security in a network with a high level of detail.

All selected use cases represent examples of hosts' behavior that may be of interest to network operators. As the dataset has been released for public use, these use cases can serve as relevant testing cases for the development of host profiles and the examination of the profiling algorithms on the long-term real-world data.

VI. Related Works

Host behavior analysis includes a wide range of topics from host application classification, over anomaly or attack detection, to network segments profiling. A recent survey by Wang *et al.* summarizes the traffic-behavioral profiling of network end-targets in [1]. The authors cluster the existing host profiling techniques to the following three clusters: connection-based, statistics-based, and deep-information profiling. For each cluster, relevant works are shown, including the comparison of accuracy, privacy, or scalability of profiling techniques presented by the surveyed works. The authors also provide an overview of the challenges and future research directions. They highlight the shortage of datasets, the need for spatial/temporal characteristics, or analysis of the evolution of traffic behavior. The results of the host behavior analyses are frequently used for Internet traffic classification. The overview of the techniques and issues of the Internet traffic classification can be found in [15], [16]. Surveys focusing on encrypted traffic include [17], [18]. In the rest of this section, we elaborate in detail on selected papers relevant to the scope of this article.

A fundamental work for host profiling is a work by Karagiannis *et al.* presented in [5], [19]. The authors introduce BLINC, a tool for flow classification. The tool implements the classification of host behavior on three different levels - social level, functional level, and application level. The authors introduce *graphlets*, a graph-like representation of the behavior patterns, for the classification. The tool uses 5-minute intervals to create behavioral patterns, and the results are evaluated on three datasets of packet traces with a maximum duration of 43 hours. Hence, the long-term temporal patterns are not addressed by the evaluation. The work of Karagiannis *et al.* was further extended by Himura *et al.* in [20] by using synoptic graphlets.

Unsupervised host behavior classification from connection patterns is presented by Dewaele *et al.* in [6]. The proposed classification is based on nine features evaluating host connectivity, dispersion, and exchanged traffic content. The features are based mainly on the information from IP addresses and packet distribution in the connections. For the unsupervised classification, the authors developed a minimum spanning three clustering techniques able to handle non-convex clusters. The authors claim to use one-year traces from the Japan-to-USA backbone. However, only one 15 minute sample is captured per day, and only five samples per month are used in the paper. Hence, the stability of the classification results is not addressed sufficiently.

Clustering and profiling of IP hosts based on network flows are discussed by Jakalan *et al.* in [2], [3]. The authors identify 15 features that can be used to cluster IP hosts. The features leverage statistics based on the individual bytes of the IPv4 addresses. The features include the number of peers, the ratio of the entropy of the second destination IP byte to the entropy of the fourth destination IP address byte, or the ratio of the number of source ports per the number of peers. DBSCAN algorithm is used for the identification of the clusters in the host behavior. Using manual analysis of the resulting clusters, the authors were able to identify hosts sending HTTP requests cluster, P2P traffic cluster, or cluster of servers. The dataset used for the experiment included 1 hour of NetFlow data from a backbone link. The collected data were reduced for analysis by excluding 10% flows and included only 10% of the source IPs in the dataset.

The identification of the significant behaviors of hosts of interest from massive traffic data and interpretation of these behaviors is discussed in paper [21] by Xu *et al.*. Authors employ relative entropy of src/dst IP addresses and ports to extract significant behavior clusters and provide an approach to define the class of the extracted behavior clusters. Further, they extract the temporal properties of the behavior classes such as popularity and membership volatility. Still, a dataset with a maximum duration of 24 hours only is used in this article. Authors extend their work by focusing on the behavior profiling for network security monitoring in [22] and by behavior analysis of Internet traffic via bipartite graphs in [24]. In the latter publication, the authors create a profile for /24 segments and use time windows from 10s to 5 minutes for clustering. Network prefix-level traffic profiling is also evaluated by Jiang *et al.* in [25]. On a one-month dataset of network flows, the authors compute characteristics such as daily aggregated traffic volume, traffic distribution in space and time, or flows size distribution. K-means algorithm, in combination with RMSE error, is used to identify the clusters. Among others, the membership stability in the clusters over time is evaluated with the conclusion that nearly all networks exhibit traffic characteristics that are stable over time. The stability of the characteristics of individual hosts is not evaluated at all. Li *et al.* leverages supervised machine learning to classify host roles using sFlow in [26]. The 24 scaled features are used for binary classifications of the hosts, such as hosts in public places vs. hosts from personal offices. The dataset used for the evaluation spans over thee months. The temporal stability of the classification models is not discussed in this article.

Apart from identifying the significant behavior of the hosts in a network, there has already been an effort to identify a significant behavior of the whole segments in a network. Proença *et al.* create the digital signatures of the network segment using flow analysis in [27]. The authors compare the following methods' performance to create the signature of a network: ant colony algorithm, Holt-Winters exponential

TABLE IX
RELATED WORKS OVERVIEW

Article	Approach	Data Type	Dataset Size	Dataset Origin	Availability	Temporal Stability	Characteristics Stability
[19]	Graphlets	Packet headers	3 traces (43.9h, 24.6h, 33.6h)	**CAIDA**	–	✔	–
[5]	Graphlets	Full packets	2 traces (1 M, 2 W)	Office Building	–	–	✔
[20]	Synoptic Graphlets	Packet headers	2 traces (12 x 15m, 2 x 30m)	**MAWI**	–	–	–
[6]	Minimum Spanning Tree	Packet headers	2 traces (7 x 15m, 5 x 15m)	**MAWI**	–	–	–
[3], [2]	DBSCAN	Network Flows	1 trace (1h)	CERNET backbone	–	–	–
[21], [22]	Relative Entropy	Packet headers	5 traces (1d, 1d, 3h, 3h, 3h)	ISP backbone	–	✔	✔
[23]	Spectral Clustering	Packet headers	1 trace (1 M)	**CAIDA**	–	✔	–
[24]	Bipartite Graphs	Packet headers	1 trace (1 M)	**CAIDA**	–	✔	–
[25]	Gaussian Mixture Models	Network Flows	1 trace (1 M)	ISP Backbone	–	✔	✔
[26]	Decision Trees, SVM	Network Flows	1 trace (3 M)	Campus Network	–	–	–
[27]	Ant Colony, Holt-Winters, PCA	Network Flows	2 traces (2 M, 3 M)	Campus Network	–	–	–
[11]	Statistical analysis	Network Flows	1 trace (2 M)	Campus Network	–	✔	–
[28]	Feature Distribution Clustering	Network Flows	2 traces (2 M, 2 M)	GEANT, Abilene	–	–	–
[29]	Coefficient of Variation	Network Flows	5 traces (24h, 96h, 24h, 96h, 24h)	**Campus and Company Network**	–	✔	–
[8]	Hierarchical Agglomerative Clustering	Network Flows	1 trace (4 W)	Residential Complex	–	✔	–

Dataset Size: m: minutes, h: hours, d: days, M: months, W: weeks Dataset Origin in **bold**: dataset available for public

smoothing, and principal component analysis. The methods are compared on the three-month network traffic data containing approximately 300 devices. Velan *et al.* propose characteristics that enable the description of the basic properties of the network segments in [11]. The characteristics also take day-night patterns into account. The characteristics were demonstrated on a dataset capturing two months of network data.

Host profiling as a means for anomaly mining is discussed in [28]. Authors use sample entropy as a summarization tool and show that by using features distribution, anomalies naturally fall into distinct clusters. Network traces for a 20 day period with sampling 1:100 are used as a dataset in this article. The issue of missing observation is not discussed even though it might have a significant effect on the resulting entropy of the characteristics.

Stability of the characteristics used for host profiling is investigated in [8], [23], and [29]. The authors of [23] explore the behavior similarity of the Internet end hosts using bipartite graphs. They evaluated the temporal stability of the behavior clusters on the one-month CAIDA dataset. They conclude that 71.8% of all end hosts in the monitored network traffic do not change the clusters during a one-hour time period, which correlates with our results of labels stability analysis on the one-year dataset. Stability in multi-device user environments is evaluated in [8]. During a four-week observation, the author finds the host behavior profiles static with an average probability of changing its profile between 3-19% over any consecutive 24 hours. The duration of the dataset is 28 days, which prevented the authors from creating a week-long profile of the host and abstract from the weekday temporal patterns. Analogous to our notion of stability, the authors of [29] employ the coefficient of variance to assess the variability of the host interaction. Using the detection of changes in the host variations, they detect malicious activities of a mail server, for example. However, the dataset used for variation assessment spans from 24 to 96 hours only.

Table IX summarizes the related works overview. The datasets used in the majority of the related works span from one day to a few months of traffic data, which prevents the authors from creating more robust and complex host behavior profiles and evaluation of the presented method over a long period. Moreover, the maximum length of the used publicly available datasets is one month. Given the short-lived host profiles, the authors do not address the issue of missing observations in the papers, and no information on the data availability is provided. There are papers that address the stability of the behavioral clusters. However, we discovered only three papers that focus on the stability of the characteristics used for host profile computation specifically. Other papers mainly address the stability of the clustering results.

VII. Conclusion

This article presents a comprehensive study of the properties of the host behavior in the computer network using a one-year-long dataset. The main goal of the study was to evaluate the characteristics used to capture host behavior from a long-term perspective. We captured one year of network traffic from /16 university network, shared the resulting dataset [10] and its analysis [9] for public use, and evaluated the host profile characteristics computed from the one-year dataset. The evaluation of the characteristics covered three main research areas: data availability, temporal stability, and stability of the host profile characteristics. Moreover, we demonstrated the caveats of the long-term host profiling on the three use cases representing the real-world hosts' behavior. Network managers can exploit the results of our study as suggestions for building a robust host behavioral profile applicable in real-world network settings.

The data availability research area explored the distribution of the missing observations when building host profiles. We demonstrated that the majority of the hosts communicated less than 22.98% in a year. Considering a typical workstation-like behavioral profile, more than 64.24% of observations can be missing. Hence, the host profiling algorithms need to take into account such a high share of missing observations. We also investigated the impact of the aggregation window size of the profile characteristic on the number of missing observations. We demonstrated that the aggregation window does not affect the servers while it has a significant impact on the workstations.

The temporal stability research confirms the diurnal and weekday patterns present in host behavior in the long term. Based on these observations, we proposed a method for labeling hosts by the temporal patterns, e.g., day talkers and night talkers. The evaluation of the stability of the labeling showed that over 70% of the hosts remained with the same labels.

The profile characteristics stability investigated how the characteristics vary in time. Using the coefficient of variance measure, we compared the variability of the observed characteristics used for host profiling. We showed how to utilize week profiling of host behavior to decrease the characteristics' variability and increase the anomaly detection performance as a result. Last, we showed how to discover clusters of hosts with either low or high variability of the profile characteristics using the DBSCAN clustering algorithm.

Our study, based on the one-year-long network data, provides a deeper understanding of the host behavior in a computer network and allows for better decision making in network and security management. The study also provides a baseline for improving the machine learning and data analytics algorithms as it shows how to build a robust and stable host behavioral profile, including a method for labeling the host behavior. The publicly available dataset can be used for further research. For example, we plan to explore automated approaches to adaptive network segmentation management, i.e., identification of typical behavioral profiles of network subnets and classification of the hosts to these segment profiles. Moreover, we can define derived features for host profiles to provide a measure for a relative risk to network security – host trustworthiness.

References

[1] P. Wang, Y. Zhou, C. Zhu, and R. Yue, "Role classification with netflow data in intranet," in *Proc. 10th Int. Conf. Adv. Comput. Intell. (ICACI)*, Xiamen, China, Jun. 2018, pp. 279–282.

[2] A. Jakalan, J. Gong, and S. Liu, "Profiling IP hosts based on traffic behavior," in *Proc. IEEE Int. Conf. Commun. Softw. Netw. (ICCSN)*, Chengdu, China, Jun. 2015, pp. 105–111.

[3] A. Jakalan, G. Jian, W. Zhang, and S. Qi, "Clustering and profiling IP hosts based on traffic behavior," *J. Netw.*, vol. 10, no. 2, p. 9, Mar. 2015.

[4] S. Chang and T. E. Daniels, "Correlation based node behavior profiling for enterprise network security," in *Proc. IEEE 3rd Int. Conf. Emerg. Security Inf. Syst. Technol.*, Athens, Greece, 2009, pp. 298–305.

[5] T. Karagiannis, K. Papagiannaki, N. Taft, and M. Faloutsos, "Profiling the end host," in *Passive and Active Network Measurement* (Lecture Notes in Computer Science), vol. 4427. Heidelberg, Germany: Springer, 2007, pp. 186–196.

[6] G. Dewaele *et al.*, "Unsupervised host behavior classification from connection patterns," *Int. J. Netw. Manag.*, vol. 20, no. 5, pp. 317–337, Aug. 2010.

[7] C. Hammerschmidt, S. Marchal, R. State, and S. Verwer, "Behavioral clustering of non-stationary IP flow record data," in *Proc. 12th Int. Conf. Netw. Serv. Manag. (CNSM)*, Montreal, QC, Canada, Oct. 2016, pp. 297–301.

[8] T. Bakhshi and B. Ghita, "Traffic profiling: Evaluating stability in multi-device user environments," in *Proc. 30th Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, Crans-Montana, Switzerland, Mar. 2016, pp. 731–736.

[9] T. Jirsik and P. Velan. (Mar. 2020). *One-Year Host Behavior Dataset*. [Online]. Available: https://github.com/CSIRT-MU/HostBehaviorInComputerNetwork-OneYearStudy

[10] T. Jirsik. (May 2020). *Host Network Traffic 2019*. [Online]. Available: https://doi.org/10.5281/zenodo.3799932

[11] P. Velan, J. Medková, T. Jirsík, and P. Čeleda, "Network traffic characterisation using flow-based statistics," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp. (NOMS)*, Istanbul, Turkey, Jun. 2016, pp. 907–912.

[12] B. Claise, B. Trammell, and P. Aitken, "Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information," Internet Eng. Task Force, RFC 7011, Sep. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc7011.txt

[13] R. Hofstede *et al.*, "Flow monitoring explained: From packet capture to data analysis with NetFlow and IPFIX," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2037–2064, 4th Quart., 2014.

[14] A. Moore, D. Zuev, and M. Crogan, "Discriminators for use in flow-based classification," Dept. Comput. Sci., Queen Mary Westfield College, London, U.K., Rep. RR-05-13, Aug. 2005.

[15] Y. Xue, L. Zhang, and D. Wang, "Traffic classification: Issues and challenges," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, vol. 8. San Diego, CA, USA, Jan. 2013, pp. 28–31.

[16] T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 4th Quart., 2008.

[17] P. Velan, M. Cermak, P. Celeda, and M. Drasar, "A survey of methods for encrypted traffic classification and analysis," *Int. J. Netw. Manag.*, vol. 25, pp. 17–31, Oct. 2015.

[18] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 76–81, May 2019.

[19] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel traffic classification in the dark," in *Proc. Conf. Appl. Technol. Archit. Protocols Comput. Commun.*, 2005, pp. 229–240.

[20] Y. Himura, K. Fukuda, K. Cho, P. Borgnat, P. Abry, and H. Esaki, "Synoptic graphlet: Bridging the gap between supervised and unsupervised profiling of host-level network traffic," *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1284–1297, Aug. 2013.

[21] K. Xu, Z.-L. Zhang, and S. Bhattacharyya, "Profiling Internet backbone traffic," in *Proc. Conf. Appl. Technol. Archit. Protocols Comput. Commun. (SIGCOMM)*, vol. 35, 2005, p. 169.

[22] K. Xu, Z. L. Zhang, and S. Bhattacharyya, "Internet traffic behavior profiling for network security monitoring," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1241–1252, Dec. 2008.

[23] K. Xu, F. Wang, and L. Gu, "Network-aware behavior clustering of Internet end hosts," in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 2078–2086.

[24] K. Xu, F. Wang, and L. Gu, "Behavior analysis of Internet traffic via bipartite graphs and one-mode projections," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 931–942, Jun. 2014.

[25] H. Jiang, Z. Ge, S. Jin, and J. Wang, "Network prefix-level traffic profiling: Characterizing, modeling, and evaluation," *Comput. Netw.*, vol. 54, no. 18, pp. 3327–3340, Dec. 2010.

[26] B. Li, M. H. Gunes, G. Bebis, and J. Springer, "A supervised machine learning approach to classify host roles on line using sFlow," in *Proc. 1st Ed. Workshop High Perform. Program. Netw. (HPPN)*, 2013, p. 53. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2465839.2465847

[27] M. L. Proença, G. Fernandes, L. F. Carvalho, M. V. O. de Assis, and J. J. P. C. Rodrigues, "Digital signature to help network management using flow analysis," *Int. J. Netw. Manag.*, vol. 26, no. 2, pp. 76–94, Mar./Apr. 2016. [Online]. Available: http://doi.wiley.com/10.1002/nem.1892

[28] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, p. 217, Oct. 2005.

[29] D. Lee and N. Brownlee, "On the variability of Internet host interactions," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, New Orleans, LO, USA, 2008, pp. 2403–2408.

Tomas Jirsik received the Ph.D. degree in informatics from the Faculty of Informatics, Masaryk University, Czech Republic. He is currently a Senior Researcher with the Institute of Computer Science, Masaryk University and a Member of the Computer Security Incident Response Team, Masaryk University, where he leads national and international research projects on cybersecurity. His research focus lies on the network traffic analysis with a specialization in host profiling. His research further includes network segmentation approaches via machine learning and host fingerprinting in network traffic.

Petr Velan received the Ph.D. degree in informatics from the Faculty of Informatics, Masaryk University, Czech Republic. He is currently a Senior Researcher with the Institute of Computer Science, Masaryk University and a Member of the Computer Security Incident Response Team, Masaryk University, where he participates on Cybersecurity Research Projects. His research focus lies on the network traffic monitoring and analysis with a specialization in network flow monitoring.