MIRAGE-APP×ACT-2024

A Novel Dataset for Mobile App and Activity Traffic Analysis

Idio Guarino*, Domenico Ciuonzo†, Antonio Montieri†, Antonio Pescapè†
*University of Verona, †University of Napoli Federico II
idio.guarino@univr.it, {domenico.ciuonzo, antonio.montieri, pescape}@unina.it

***Abstract*—This paper presents MIRAGE-APP×ACT-2024, a novel dataset originating from the efforts of the MIRAGE project, which collects traffic and corresponding ground-truth data from human-generated mobile-app interactions. By providing detailed insights into traffic patterns, the dataset supports advancements in mobile network optimization, security, and application performance evaluation. To this aim, we present an initial characterization and modeling of MIRAGE-APP×ACT-2024. This new release aims to facilitate further research and development in mobile network traffic analysis, focusing on interactive, multi-activity apps and activity-level analysis.**

***Index Terms*—Android apps; mobile apps; mobile traffic; user activity; reproducible research; open dataset.**

I. INTRODUCTION

As in many experimental research fields, replicability and reproducibility are crucial for significant progress. In network traffic analysis, essential for profiling, management tasks, and understanding traffic patterns, the lack of public datasets has hindered advances [1]. Existing research often relies on private datasets, limiting repeatability and broader analysis. Smartphones now lead communication, with apps generating significant and rapidly changing traffic. This, along with encryption and privacy concerns, poses new challenges for network traffic analysis. The need for public, human-generated mobile traffic datasets, especially those categorized by specific user activities like chat or video call, is greater than ever.

In response, we introduce MIRAGE, a system for capturing and creating ground truth for mobile-app traffic. The **contributions** of this paper are fourfold: (i) we survey publicly available encrypted-traffic datasets, highlighting their characteristics and limitations (Sec. II); (ii) we describe the updated MIRAGE architecture, designed to generate accurate, reproducible datasets of mobile-app traffic, focusing on both apps and user activities (Sec. III); (iii) we release[1] and describe the format of MIRAGE-APP×ACT-2024 dataset, arranged by apps and specific user activities—e.g., chat or video-call (Sec. IV)—fostering replicability and extending its application to various use cases; (iv) we provide a first brief characterization of the dataset, demonstrating its suitability for a wide range of tasks (Sec. V). By releasing this dataset, we aim to support the research community in advancing mobile-app traffic analysis, enabling more nuanced studies that consider the context of user activities.

This work is partially supported by the "RESTART" Project and the "xInternet" Project within the PRIN 2022 program (D.D.104-02/02/2022), funded by MUR. This manuscript reflects only the authors' views and opinions, and the Ministry cannot be considered responsible for them.

[1]https://traffic.comics.unina.it/mirage/mirage-2024.html

Table I: Datasets collected using the MIRAGE system. All datasets are publicly available for download.[2]

Dataset	Year	Capture Span	Label Space	👤
MIRAGE-2019 [2]	2019	05/17 – 05/19	40 apps	○
MIRAGE-VIDEO [3]	2020	06/19 – 03/20	4 categories / 8 apps	○
MIRAGE-COVID-CCMA-2022 [4]	2022	04/21 – 12/21	9 apps / 3 activities	◐
MIRAGE-APP×ACT-2024	2024	04/21 – 12/23	20 apps / 5 activities	●

👤 denotes the availability of activity-level labeling.

II. RELATED WORK

Reproducibility and up-to-date evaluations pose significant research challenges in mobile network traffic analysis. Data availability has long been an issue and only recent years have witnessed an increasing attention toward benchmarks for real-world evaluations (see [1] for a recent review). Accordingly, in what follows, we discuss only the most recent and related datasets to our MIRAGE-APP×ACT-2024, highlighting peculiarities, differences, and limitations.

The closest dataset to ours is **UTMobileNetTraffic2021** [5], collected in 2019 and containing +29h of mobile traffic with 16 apps and up to 3 activities per app. However, despite its diversity, such a dataset is bot-generated (instrumented via a BASH script) and does not reflect the most recent trends. A second dataset related to our proposal is **NUDT_MobileTraffic** [6], which was collected during May-Jul. 2020 (i.e. during the COVID pandemic). It contains the traffic generated from 350 apps and excited by human users, but the traffic is operated within an ad-hoc VPN. Also, the dataset does not contain activity-level labels. The third dataset discussed here is **ITC-Net-blend-60** [7], collected during Oct.-Dec. 2021 and contains the traffic generated from 60 apps. Here too, no activity-level labeling is provided.

Tab. I summarizes the datasets collected using the MIRAGE system [2] and released publicly over the past five years.[2] All datasets are *human-generated*, capturing traffic from various Android apps. The MIRAGE-2019 [2] includes traffic of typical app usage across different categories, focusing on common functionalities such as service registration, login, and regular interactions. In contrast, MIRAGE-VIDEO [3] captures traffic from mobile video apps into four custom

[2]https://traffic.comics.unina.it/mirage/

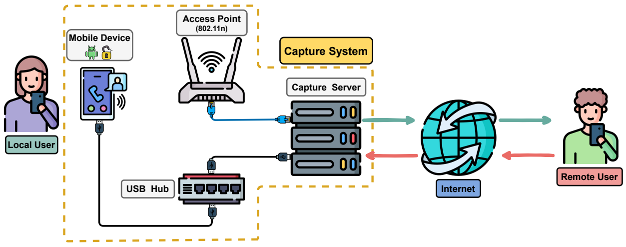


Fig. 1. *Capture System* of the enhanced MIRAGE architecture.[3]

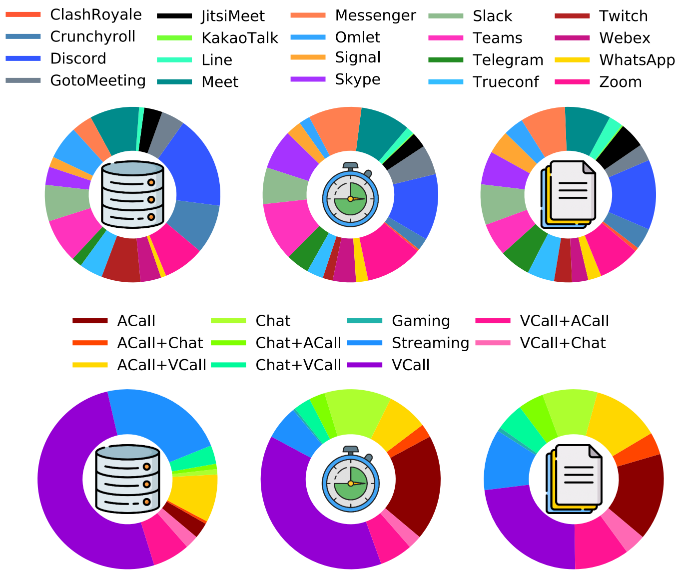


Fig. 2. Overview of MIRAGE-APP×ACT-2024: amount of traffic data, capture time, and number of capture sessions. Breakdown by app (top) and activity (bottom).

categories (Cloud VR, Short Video, Video Chat, and Video on Demand), each with unique purposes and functionalities. The MIRAGE-COVID-CCMA-2022 [4] features specific activities on communication and collaboration apps, whereas the MIRAGE-APP×ACT-2024 involves multiple user activities (and their interactions) and covers more apps and activities. As a result, we aim to address the lack of *human-generated traffic* and detailed *activity-level labeling* and to provide a current representation of *real-world mobile network traffic*.

III. ENHANCED MIRAGE ARCHITECTURE

For the collection of MIRAGE-APP×ACT-2024, we enhance the original MIRAGE architecture [2] in its *Capture System* (see Fig. 1) to gather traffic from modern mobile apps in a more effective fashion. The *Capture Server* is a workstation equipped with an IEEE 802.11 *Access Point*. This allows connectivity to the *Android Mobile Device*, which generates traffic when the experimenter uses the apps.

The server is connected to the Internet using a wired connection and performs network address translation. To meet current mobile app networking needs (especially for video or audio calls), the access point supports the *802.11n* standard for better performance and stability.

Each mobile device is connected to the Capture Server via a *USB hub*, allowing the use of *Android Debug Bridge (ADB)* to send commands to the device and receive the responses. Notably, our setup requires a rooted mobile device and supports capturing from multiple devices simultaneously.

In each capture session, the system collects traffic from a *target app* used by a *local user* on a device identified by its *MAC address*[4], producing a PCAP traffic trace and log files for ground truth generation.[5] A local user starts a session by connecting a mobile device to the USB hub. This automatically triggers capturing traffic from the Capture Server's wired connection using `tcpdump` and saving log files. These log files map each socket descriptor (`<IP:port>` pairs) to the *Android package name* by executing the Linux `netstat` command on the device, ensuring reliable *labeling* of each biflow at the *app level*. When the log file does not contain an *exact* match for some biflows, they are labeled with the name of the *most-common* package within the log file.[6] Such ground-truth generation method allows us to label exactly $\approx 78\%$ of the biflows. Also, the ground truth is enriched by manually assigning the *activity label* based on the knowledge of the (multiple) activities completed by the user during the capture (one or two activities are performed sequentially, see Sec. IV).

IV. MIRAGE-APP×ACT-2024 DATASET

The MIRAGE-APP×ACT-2024 dataset was collected through a crowdsourced campaign involving over 240 volunteers, including students and researchers.[7] The campaign took place between April 2021 and December 2023 at the University of Naples "Federico II", by exploiting the *GARR* network infrastructure with a bandwidth of up to 100 Mbps.

We collected traffic data of 20 popular apps, listed at the top of Fig. 2. We selected the apps based on their surge in popularity and traffic volume during the capture span[8] and the possibility of doing multiple activities. Indeed, each app was used to perform at least one of the following activities: *Chat* (`Chat`): two participants exchange textual messages and/or multimedia content; *Audio-call* (`ACall`): two participants transmit only audio; *Online Gaming* (`Gaming`): the user plays interactive games in real-time; *Video-call* (`VCall`): multiple attendees transmit both video and audio; *Video-streaming* (`Streaming`): the user watches videos in real-time.

Volunteers used four mobile devices running Android 10: a Xiaomi Mi 10 Lite, a Google Nexus 6, and two Samsung Galaxy A5.[9] In each capture session, experimenters performed one or two *specific activities* on a target app for $7-80$ minutes. Each capture involved connecting the device and launching the target app to perform the prescribed set of activities.

[3]This diagram has been designed using images from *flaticon.com*.
[4]MAC filtering allows us to distinguish traffic from multiple devices.
[5]To minimize background traffic, only the target app has network access.
[6]The dataset explicitly marks the biflows labeled exactly.
[7]The privacy of the volunteers was protected using fictional accounts.
[8]Sandvine, "The Mobile Internet Phenomena Report", May 2021.
[9]All devices were equipped with the custom firmware *LineageOS 17.1*.

Table II: MIRAGE-APP×ACT-2024 is released in JSON format: one file per capture session. A JSON file contains three types of traffic data for each biflow: (i) *Per-packet data*; (ii) *Per-flow features* extracted from sets of upstream, downstream, and complete IP packet lengths and inter-arrival times; (iii) *Per-flow metadata* of complete biflow (`BF`) and related upstream (`UF`) / downstream (`DF`) flows.

Data	Name	Description
Per-packet Data	`timestamp`	Timestamp expressed as Unix Epoch time
	`src_port`	Source transport-layer port
	`dst_port`	Destination transport-layer port
	`packet_dir`	Packet direction (0 upstream, 1 downstream)
	`IP_packet_bytes`	Number of bytes in IP payload
	`IP_header_bytes`	Number of bytes in IP header
	`L4_header_bytes`	Number of bytes in L4 header
	`L4_payload_bytes`	Number of bytes in L4 payload
	`iat`	Inter-arrival time
	`TCP_win_size`	TCP window size (0 for UDP packets)
	`TCP_flags`	TCP flags (empty for UDP packets)
	`L4_raw_payload`	Byte-wise raw L4 payload (integer $\in [0, 255]$)
Per-flow Features	`min`	Minimum
	`max`	Maximum
	`mean`	Arithmetic mean
	`std`	Standard deviation
	`var`	Variance
	`mad`	Mean absolute deviation
	`skew`	Unbiased sample skewness
	`kurtosis`	Unbiased Fisher kurtosis
	`q_percentile`	q^{th} percentile ($q \in [10:10:90]$)
Per-flow Metadata	`BF_device`	MAC address of the mobile device
	`BF_label`	Android-package name
	`BF_activity`	Activity type
	`BF_app_category`	Android-package category name ‡
	`BF_label_version_code`	Android-package version code
	`BF_label_version_name`	Android-package version name
	`BF_labeling_type`	Exact or most-common labeling
	`{BF,UF,DF}_num_packets`	Number of packets
	`{BF,UF,DF}_IP_packet_bytes`	Total bytes in IP packets
	`{BF,UF,DF}_L4_payload_bytes`	Total bytes in L4 payloads
	`{BF,UF,DF}_duration`	(Bi)flow duration in seconds
	`{UF,DF}_MSS`	TCP Maximum Segment Size (0 for UDP flows)
	`{UF,DF}_WS`	TCP Window Scale factor (0 for UDP flows)

‡ Category provided by the Google Play Store.

For captures where the user performed multiple activities in sequence, each activity lasted for 3 minutes followed by 1 minute of silence.

Fig. 2 presents an overview of the resulting MIRAGE-APP×ACT-2024 dataset, including the amount of traffic data (), capture time (), and capture sessions () broken down by apps and activities. Overall, the dataset comprises 2244 sessions, totaling ≈208 GB and ≈490 hours of traffic.

After the collection campaign, we processed the PCAPs and enriched them with the ground truth. We organized the resulting data in JSON files (one for each PCAP) containing information at the biflow level (i.e., the bidirectional sequence of packets sharing the quintuple). Tab. II reports the extracted data categorized into *three* groups: (i) *Per-packet data*: 11 header fields plus the L4 payload of all packets of a biflow; each entry identifies a list with a length equal to the number of packets in the biflow. (ii) *Per-flow features*: data extracted from the complete biflow along with its upstream and downstream flows; this results in 17 statistical features computed on the sets of upstream, downstream, and complete IP packet lengths and inter-arrival times, totaling 102 features. (iii) *Per-flow metadata*: ground-truth information related to the app (e.g., package name and version, category, activity) and the device (i.e., MAC address), along with various counters related to the complete biflow and upstream/downstream flows. We publicly

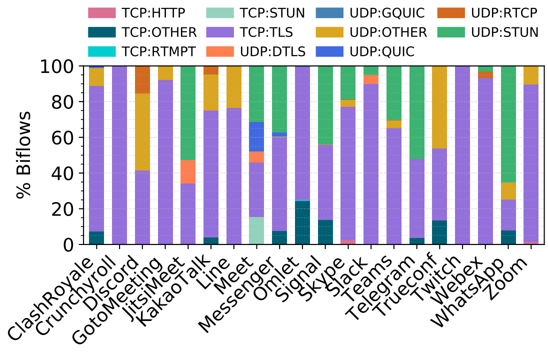


Fig. 3. Protocol distribution in terms of biflows. Data refers to *exactly* labeled biflows.

release MIRAGE-APP×ACT-2024 in such JSON format.[1]

V. Dataset Characterization and Ending Remarks

Characterization by Protocol. Fig. 3 breaks down the biflows percentage for every protocol apps adopt. We observe a large share of TLS traffic for most apps ranging from ≈ 52% (`Messenger`) to 100% (`Crunchy` and `Twitch`). Conversely, for `Discord`, `JitsiMeet`, `Meet`, `Signal`, `Telegram`, and `WhatsApp`, we note a prevalence of STUN traffic ranging from ≈ 31% (`Meet`) to ≈ 65% (`WhatsApp`). Notably, `Discord`, `JitsiMeet`, and `Meet` have a significant share of RTCP, DTSL, and QUIC traffic, respectively.

Characterization by Activities. Fig. 4 depicts the evolution of traffic volume (in KB) of `Trueconf` when executing multiple user activities sequentially (cf. Sec. IV). Values are computed across different traces whose traffic is aggregated into non-overlapping 5 s intervals. As expected, `Chat` generates a traffic volume ≈ 3 orders of magnitude fewer than `ACall` and `VCall`, with the latter being more network-intensive. This makes the different activities distinguishable, despite being performed in the same capture session. Interestingly, the sequence `VCall+Chat` keeps a moderate traffic volume during `Chat`: this is likely due to a residual network activity of `VCall` after the closure of the communication.

Traffic Modeling via Markov Chains. We aim to disclose the peculiar characteristics of a given app by modeling its traffic

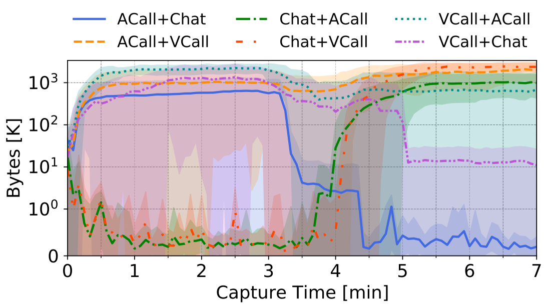
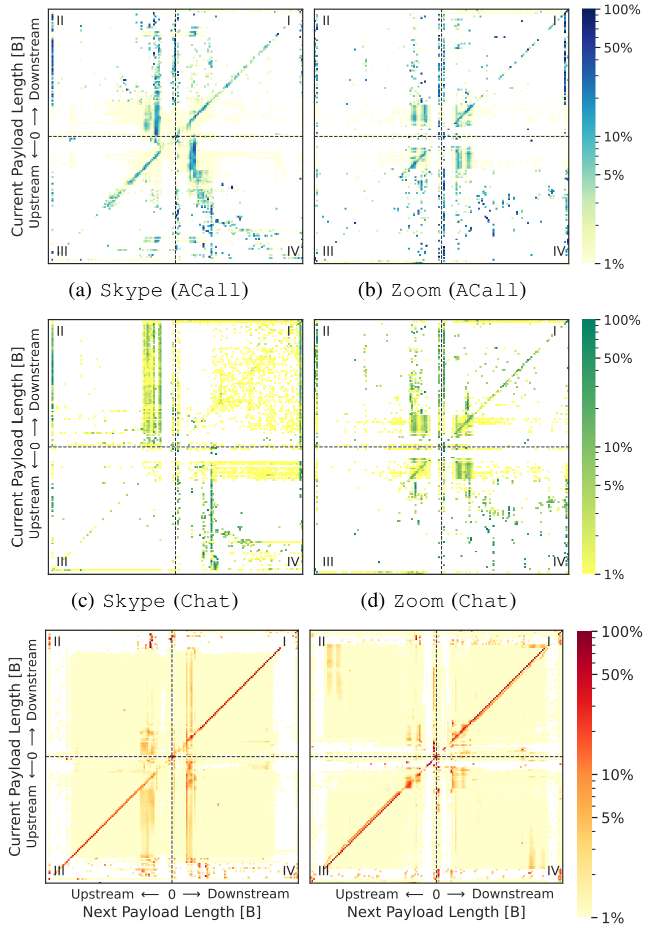


Fig. 4. Traffic volume of `Trueconf` within 5s intervals when performing multiple activities sequentially. Values are reported as $mean \pm std.dev$ across different captures. The y-axis is in log scale.



(a) `Skype` (`ACall`) (b) `Zoom` (`ACall`)
(c) `Skype` (`Chat`) (d) `Zoom` (`Chat`)
(e) `Skype` (`VCall`) (f) `Zoom` (`VCall`)

Fig. 5. Transition matrices of payload length and packet direction for `Skype` (a, c, e), and `Zoom` (b, d, f), and related to `ACall` (a, b), `Chat` (c, d), and `VCall` (e, f) activities. Data refers to *exactly* labeled biflows.

through a first-order (*Multimodal*) *Markov Chain* [8] based on the user activity. We exploit this modeling to understand the distinctive behaviors of mobile apps and related activities by examining the relationships between subsequent packets and providing an intuitive visual representation.

Specifically, for a given $\langle app, activity \rangle$ pair, we consider both the L4-payload length (`PL`) and the direction (`DIR`) of packets in each biflow.[10] We then derive the corresponding *transition probability matrix* ($\boldsymbol{P}$).[11] In $\boldsymbol{P}$, $\langle (p_i, d_i), (p_j, d_j) \rangle$ represents the probability that the next packet will have a `PL` of p_j bytes and a direction d_j, given that the last observed packet had a `PL` of p_i bytes and a direction d_i.

Fig. 5 shows the matrices obtained for `Skype` and `Zoom` when performing the `ACall`, `Chat`, and `VCall` activity. From a visual inspection, *distinct patterns* can be identified based on the specific $\langle app, activity \rangle$.

[10] As pre-processing steps, we remove null-payload packets and discretize the `PL` using an adaptive binning method based on K-means. We choose 80 bins by considering both the quantization error and the number of bins.

[11] Matrices are learned via *Maximum-Likelihood* estimation.

For both apps, we observe a *dark pattern on the main diagonal* for `ACall` and `VCall`, namely, a trend to generate pairs of packets with equal `PL` and `DIR` in both upstream and downstream, reflecting continuous traffic generation due to active user participation. Moreover, for `VCall`, the highly probable values are less sparse and are mainly concentrated along the main diagonal. Conversely, for `ACall`, `PL` is typically ≤ 500 B, while for `VCall`, it ranges in $[60, 1300]$ B.

On the other hand, especially for `ACall`, we observe some *darker areas*, indicating that apps are more likely to generate packets with similar characteristics. Interestingly, for `Zoom`, these areas do not depend on the direction of the packets (i.e., they appear in all quadrants). Differently, for `Skype`, they only appear when the observed pairs of packets have opposite directions (i.e., in the II and IV quadrants). This suggests a stronger correlation between the traffic exchanged by the two communicating parties.

Finally, for `Chat`, some *vertical patterns* appear in the II and IV quadrants, highlighting a high probability of downstream (resp. upstream) `PL`s that do not depend on the current upstream (resp. downstream) `PL`. These patterns also indicate that both apps often generate either very small ($\leq$ 100 B) or very large ($\geq$ 1460 B) `PL`s. Interestingly, for `Zoom`, the patterns appear in all quadrants, suggesting that the `DIR` of the previous packet has less impact on the `PL` of the next one.

Ending Remarks. In this work, we described the enhanced MIRAGE architecture for capturing mobile-app traffic and building the related ground truth. The resulting MIRAGE-APP×ACT-2024 aims to support replicable traffic analysis associated with user activities. We showed exemplifying traffic analysis tasks concerning characterization and modeling at app and activity levels. We expect that MIRAGE-APP×ACT-2024 will be used in future research on mobile traffic analysis, in the wake of our previous datasets [2, 3, 4].

REFERENCES

[1] G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapè, "AI-powered internet traffic classification: Past, present, and future," *IEEE Commun. Mag.*, 2024.

[2] G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé, "MIRAGE: Mobile-app traffic capture and ground-truth creation," in *ICCCS'19*, 2019, pp. 1–8.

[3] A. Montieri, G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapè, "Packet-level prediction of mobile-app traffic using multitask deep learning," *Computer Networks*, vol. 200, p. 108529, 2021.

[4] I. Guarino, G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapè, "Contextual counters and multimodal deep learning for activity-level traffic classification of mobile communication apps during covid-19 pandemic," *Computer Networks*, vol. 219, p. 109452, 2022.

[5] Y. Heng, V. Chandrasekhar, and J. G. Andrews, "UTMobileNetTraffic2021: A Labeled Public Network Traffic Dataset," *IEEE Networking Letters*, vol. 3, no. 3, pp. 156–160, 2021.

[6] S. Zhao, S. Chen, F. Wang, Z. Wei, J. Zhong, and J. Liang, "A large-scale mobile traffic dataset for mobile application identification," *The Computer Journal*, vol. 67, no. 4, pp. 1501–1513, 2024.

[7] M. Bayat, J. Garshasbi, M. Mehdizadeh, N. Nozari, A. Rezaei Khesal, M. Dokhaei, and M. Teimouri, "ITC-Net-blend-60: a comprehensive dataset for robust network traffic classification in diverse environments," *BMC Research Notes*, vol. 17, no. 1, p. 165, 2024.

[8] M. Shen, M. Wei, L. Zhu, and M. Wang, "Classification of encrypted traffic with second-order Markov chains and application attribute bigrams," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1830–1843, 2017.