THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

The attention leak

Behavioral inference from temporal metadata in network traffic

Citation for published version:
Donaire, A, Borrego, C & Juarez, M 2026, 'The attention leak: Behavioral inference from temporal metadata in network traffic', *Journal of network and computer applications*, vol. 254, 104560, pp. 1-13. https://doi.org/10.1016/j.jnca.2026.104560

Digital Object Identifier (DOI):
10.1016/j.jnca.2026.104560

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Journal of network and computer applications

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 17. Nov. 2026

Journal of Network and Computer Applications 254 (2026) 104560

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Research paper

The attention leak: Behavioral inference from temporal metadata in network traffic

Alejandro Donaire [a], Carlos Borrego [a,*], Marc Juarez [b]

[a] Department of Information and Communications Engineering Autonomous, University of Barcelona, Barcelona, Spain
[b] School of Informatics, University of Edinburgh, Edinburgh, Scotland, United Kingdom

ARTICLE INFO

Keywords:
Network traffic analysis
Privacy-preserving communication
Behavioral traffic fingerprinting

ABSTRACT

Human attention is a dynamic and limited resource, essential for navigating complex tasks and environments. In the digital age, this resource is increasingly strained by frequent task switching, driven by constant notifications and fragmented workflows. While the cognitive costs of task switching are well-documented, its implications for digital privacy remain underexplored. This study investigates a novel privacy threat: the inference of task switching from network traffic by network service providers. We demonstrate that behavioral signatures, such as task durations and switching patterns, can be extracted from temporal metadata, potentially revealing sensitive information about users' cognitive states and habits. We introduce a machine learning-based approach to infer these patterns and empirically validate its effectiveness on network traffic traces. To counter this threat, we propose a mitigation strategy that obfuscates task-switching signals without substantially impacting network performance. Our findings bridge cognitive science and network security, highlighting a new dimension of privacy risk in modern digital infrastructures and offering practical pathways toward more privacy-preserving communication systems.

1. Introduction

In the digital era, task switching has become a pervasive feature of modern cognition, shaped by an environment of constant notifications, overlapping responsibilities, and fragmented workflows. While the ability to shift flexibly between tasks is often framed as a hallmark of adaptability, research has consistently demonstrated its cognitive burden: frequent context changes incur performance decrements known as *switch costs*, manifesting as higher error rates and delayed response times when performing a task (Monsell, 2003). Beyond these immediate effects, habitual multitasking may impair the brain's capacity for sustained attention, reduce working memory efficiency, and contribute to a diminished sense of cognitive control (Uncapher et al., 2017). As digital technologies increasingly mediate work and personal life, understanding the cognitive and behavioral consequences of task switching is essential for designing environments that support, rather than undermine, human focus and productivity.

Patterns of attention and distraction that emerge during the execution of one or multiple tasks may be considered a form of private data. When applied to computer-based activities, this perspective suggests that metrics such as the duration spent on a given task, as well as window or application switches, can serve as behavioral signatures—implicit descriptors of an individual's working style. These subtle traces, often overlooked, offer insight into cognitive rhythms and task engagement, raising important questions about the boundaries of privacy in digitally mediated environments.

This study investigates the privacy risks posed by network service providers, focusing on their potential to infer users' behavioral rhythms from network traffic metadata. In particular, we examine how providers may extract fine-grained temporal patterns such as pauses, interruptions, and fluctuations in activity that reflect underlying cognitive and behavioral states. We argue that the core privacy threat lies not merely in identifying isolated events, but in reconstructing structured patterns of activity and inactivity that characterize how individuals work, rest, focus, or disengage. Such inferred dynamics may expose sensitive aspects of users' routines, productivity cycles, attention stability, or stress responses. Our work therefore centers on protecting these higher-level behavioral signatures embedded in network traces, highlighting a overlooked vector for privacy leakage within digital infrastructures.

Concretely, we present a methodology that demonstrates how the analysis of network traffic alone can be used to infer task-switching patterns. The resulting estimates serve as a proxy for user characterization or profiling. We use machine learning techniques to infer task switching from network traffic features extracted from packet streams and test it using real and synthetic data.

* Corresponding author.
E-mail addresses: alejandro.donaire@uab.cat (A. Donaire), carlos.borrego@uab.cat (C. Borrego), marc.juarez@ed.ac.uk (M. Juarez).

https://doi.org/10.1016/j.jnca.2026.104560
Received 11 December 2025; Received in revised form 21 May 2026; Accepted 27 July 2026
Available online 6 August 2026
1084-8045/© 2026 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

This study makes three primary contributions. First, it presents evidence that inter-task intervals observable in network metadata constitute a meaningful privacy risk, connecting cognitive science with network-level threat models. Second, it proposes a methodology for detecting such patterns using machine learning, illustrating the potential for user profiling from network traffic. Third, it proposes a mitigation strategy designed to obfuscate inter-task signals, enhancing privacy without significantly degrading network performance.

The remainder of the study is structured as follows: Section 2 reviews related work on attention, task switching, profiling, and machine learning. Section 3 introduces the general methodology. Section 4 presents the implementation of this methodology, together with the datasets, experiments, and results. Section 5 discusses a mitigation strategy, and Section 6 concludes with implications and future work.

2. Background and related work

In this section, we review relevant literature from the foundational domains underlying this work: human attention, task switching, digital behavioral profiling, and machine learning. On one hand, literature suggests that machine learning can be a promising approach to inferring cognitive states, such as attention or focus, from the more measurable proxy of task-switching logs in digital environments. On the other hand, prior work indicates that these cognitive states constitute a form of behavior that could be leveraged for digital profiling.

2.1. Human attention and task switching

Attention is a fundamental cognitive process in humans, allowing us to focus on a subset of all incoming sensory information (Herrmann and Knight, 2001). More precisely, it can be seen as a mechanism that allocates cognitive resources, enhancing the processing of relevant features of the environment while suppressing the processing of irrelevant features (Polk et al., 2008). This mechanism improves perception, reduces reaction times, and boosts learning, among other benefits (Mishra and Gazzaley, 2012; Keller et al., 2020). A key distinction is often made between top-down (goal-directed) and bottom-up (stimulus-driven) attention, referring to whether attention is primarily guided by our internal goals and knowledge or by salient external stimuli in the environment, respectively (Corbetta and Shulman, 2002).

Closely related to attention is task switching. Formally, task switching is a cognitive process that involves dynamically and flexibly adapting to changing behavioral contexts or goals, which enables individuals to efficiently switch between different tasks (Jamadar et al., 2015). Each cognitive task requires appropriately configuring a distinct set of mental operations and resources, known as a task-set (Monsell, 2021). A relevant goal in task-switching research is understanding the balance of flexibility and stability (e.g., maintaining cognitive focus on the current task without compromising the ability to rapidly switch to an emergency) (Monsell, 2003).

In recent years, especially with the rise of attention-grabbing technologies, this balance is being increasingly disrupted. For instance, studies have found that smartphone notifications, even when ignored, elicit task-irrelevant thoughts, impairing cognitive performance (Stothart et al., 2015). More strikingly, even the mere presence of one's own smartphone has been linked to reduced focus (Ward et al., 2017).

In a 2017 review (Uncapher et al., 2017), which considers media broadly as "all forms of mediated communication of information or data", such as TV, radio, or social media, they report that heavier media multitaskers tend to demonstrate reduced performance compared to lighter media multitaskers in domains such as working memory, sustained attention, and inhibitory control. They distinguish these groups based on the frequency with which individuals "juggle multiple media streams simultaneously". Similarly, another study observed a positive correlation between levels of media multitasking and self-reports of attentional failures, as well as spontaneous and deliberate mind wandering (Ralph et al., 2014). Indeed, task-switching frequency is directly related to focus and attention, which indicates it can be a reasonable proxy for these cognitive functions, or cognitive control more generally.

Generally, there seems to be a wide and natural variety of task switching profiles. One study, for instance, analyzed how users perform multiple tasks within online search sessions using large-scale search logs, and identified three user typologies based on the average number of tasks per session: focused users, multi-taskers, and supertaskers (Mehrotra et al., 2016). Another study distinguishes between Heavy Media Multitaskers and Light Media Multitaskers using a tool called the Media Multitasking Index (Ophir et al., 2009). As we discuss later in more detail, the existence of these profiles opens the possibility for digital behavioral profiling in terms of user attention.

When considering the question of how to reliably measure task-switching in individuals, existing literature provides several approaches. Early laboratory research on task-switching, such as in Rogers and Monsell (1995), required participants to directly alternate between simple, predefined cognitive operations. In another study focusing on digital environments, researchers used a custom screen-capture software that collected screenshots every 5 s. These were then coded into content categories, and task-switching was defined as changes of category (Yeykelis et al., 2014). In a similar but more straightforward way, a study employed a software agent that captured the active window (either the application name, for non-browser tasks, or the domain name, for browser-tasks), along with timestamps. A task was then defined as the activation or reactivation of applications or websites (Judd, 2015). Early research from Microsoft also logged the active windows, but included additional information such as mouse and keyboard activity. Researchers then manually identified "task episodes", defining a switch as the transition from one episode to another (Iqbal and Horvitz, 2007). At a broader scale, a different work analyzed a month of search-engine logs and, using a latent-structural SVM task-extraction framework, reconstructed higher-level goals, aiming to distinguish genuine task boundaries (Mehrotra et al., 2016). These works highlight that there is a clear diversity in settings and approaches to measuring task switching, ranging from controlled experiments to real-world digital scenarios, with studies differing in granularity and degree of post-processing.

2.2. Digital behavioral profiling

A broad body of work shows that individuals can be profiled or even uniquely identified by their digital behavior logs generated through everyday device usage. Indeed, modern surveys agree that virtually any facet of device interaction can feed into user models that either authenticate or describe their characteristics (Auliya et al., 2021). Early foundational studies demonstrated this using various data modalities. For example, Yang showed that patterns in web browsing sessions can serve as a fingerprint (Yang, 2010) . Similarly, a relevant result by Montjoye et al. found that call and location records from mobile phones were highly distinctive, and could be used for user identification in a large database of over one million individuals (de Montjoye et al., 2013).

More recently, many studies have confirmed that the collection of apps a person uses, and how they use them, is a rich source for profiling. For instance, researchers in 2018 showed that analyzing mobile network logs to see which apps a user connects to can reveal identity (Tu et al., 2018). Another study by Welke et al. used large-scale mobile app data and found that usage patterns limited to just the top 60 globally most-used apps could uniquely identify 99.4% of users (Welke et al., 2016).

Likewise, researchers have mined device logs to predict attributes like age, gender, or personality. One study found that the set of apps on a user's phone can reliably indicate gender (Seneviratne et al., 2014). Another showed that smartphone usage patterns, such as frequency

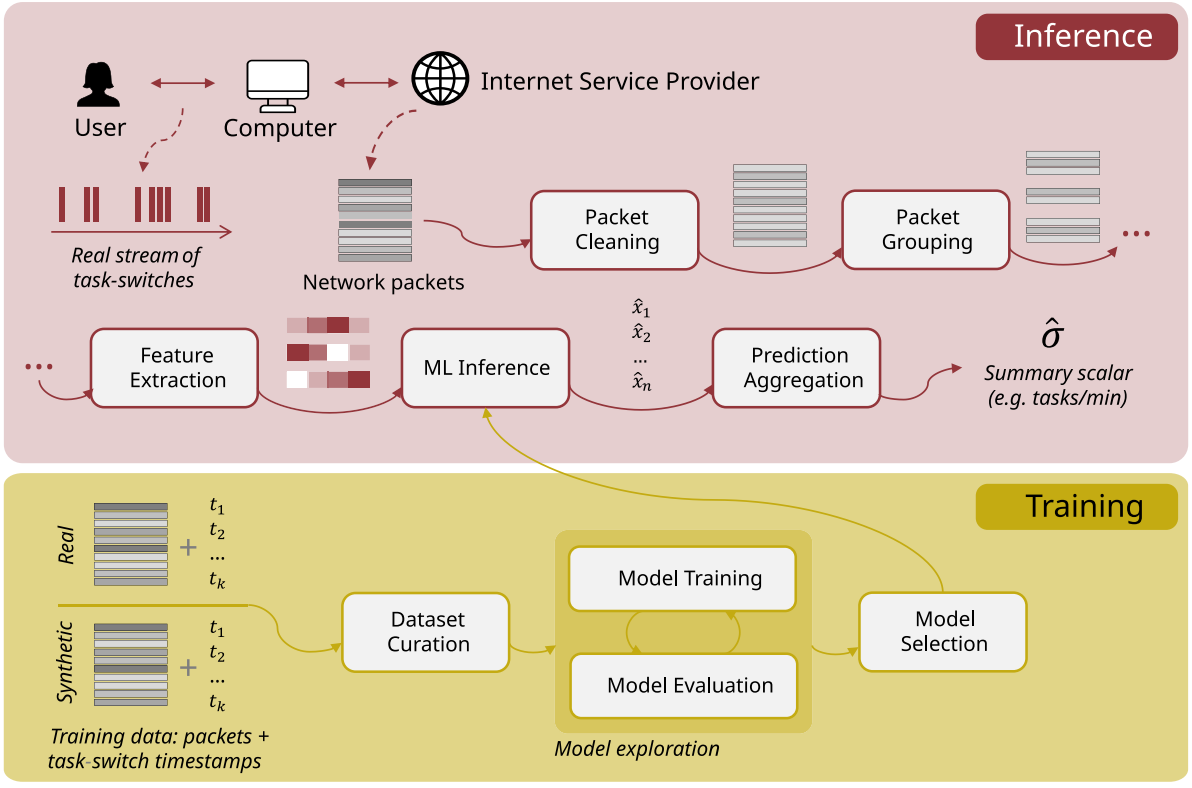

Fig. 1. Overview of the proposed methodology for inferring task switching dynamics from network data, illustrating the separation between training and inference pipelines.

of communication or mobility routines, are correlated with Big-Five personality traits (Montjoye et al., 2013). A further study also showed that by analyzing phone logs, it is possible to classify users along personality dimensions better than random chance (Chittaranjan et al., 2011).

A notable research direction employs network traffic and metadata for user profiling. A study from 2016, for instance, used application-specific traffic metadata to build unique behavioral profiles (Alotibi et al., 2016). Another work applied hierarchical clustering to Internet traffic traces, revealing interpretable clusters of hosts with common usage profiles (Kissel, 2006). Notably, one approach proposed a fingerprinting framework that identifies individual users behind domestic routers using only NetFlow records, achieving over 90% accuracy in large-scale real-world deployments without relying on IP addresses or payloads (Verde et al., 2014).

2.3. Machine learning approaches

Machine learning (ML) addresses the pervasive challenge of automatically discovering complex, often non-linear mappings between high-dimensional observations and desired targets. It offers a general-purpose solution for supervised learning tasks such as classification, regression, and sequence labeling, making it an ideal tool for inferring behavioral patterns from network traffic. In this work, we employ algorithms from three distinct families of ML models. First, we utilize classical algorithms, including Logistic Regression, Support Vector Machines (Vapnik, 2013), K-Nearest Neighbors (Cover and Hart, 1967), Bernoulli Naive Bayes, and Random Forests (Breiman, 2001). These models provide high interpretability and serve as strong baselines for our different approaches. Second, we leverage gradient-boosting ensembles, which extend the decision-tree paradigm by iteratively refining models to minimize residual error. We specifically employ implementations such as XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018), which are particularly effective for tabular data. Third, we also employ deep learning architectures. We focus specifically on Temporal Convolutional Networks (Lea et al., 2016), which apply layered convolution operations to time-series data, efficiently capturing long-range temporal dependencies across time intervals.

A large body of work has studied the application of machine learning techniques to the analysis of encrypted traffic in the security domain (Alwhbi et al., 2024; Dong et al., 2025). Most closely related to ours are studies that apply machine learning-based traffic analysis on private communications to infer sensitive information about Internet users, such as identifying sensitive applications running on a smartphone (Taylor et al., 2017), identifying specific IoT devices (Msadek et al., 2019), and, more recently, inferring LLM responses (McDonald and Or, 2025). Among these, website fingerprinting techniques have attracted considerable attention from the security community due to their reported success in inferring web browsing information over encrypted (Siby et al., 2019) and anonymized (Rimmer et al., 2018; Sirinam et al., 2018) protocols, effectively undermining the confidentiality properties that these protocols aim to provide. These techniques exploit similar traffic features to ours but have a different goal: identifying the destination of web communications, instead of predicting task switches at the client side. Although some of these studies have been criticized for making unrealistic assumptions in the evaluation setting (Juarez et al., 2014), the consensus is that such attacks pose a relevant threat. Consequently, a wide range of countermeasures have been proposed to mitigate the attacks (Mathews et al., 2023).

Existing research on digital behavioral profiling and task switching has progressed largely in parallel, leaving an important blind spot at their intersection. Profiling studies have typically aimed at re-identification or high-level attribute inference using different artifacts, such as full browser sessions, application sets or location traces. Yet, they have not fully exploited other, more fine-grained clues like those embedded in users' task-switching rhythms. Conversely, task-switching research has relied on client-side tools to analyze cognitive

costs or productivity effects, without attempting to infer switching behavior indirectly. To our knowledge, no prior work has shown how the signatures of task-switching observable in network or other low-level metadata can be harnessed for behavioral segmentation, leaving a significant research gap that this study seeks to address.

3. General methodology

3.1. Motivation and central premise

Online activity is characterized by continuous shifts between applications, as users engage in a multitasking flow shaped by diverse digital tasks and contexts. These task-switching patterns constitute a valuable behavioral signal for characterizing and profiling users through their interaction routines. In this work, we operationalize task switching at the level of observable application changes (i.e., foreground window transitions), which we treat as a measurable proxy rather than a direct representation of cognitive state changes. Measuring such sensitive information directly, however, would typically require highly intrusive client-side instrumentation with privileged access to the device, comparable to the operating system itself. Because such access is impractical, this data is generally considered safe. The central premise of this work is that this limitation can be overcome, assuming the following: (i) users inevitably generate low-level network traffic simply by operating a connected device; (ii) this traffic is sufficiently rich to act as a proxy for their behavior; and (iii) modern statistical and machine learning models can exploit this proxy to infer users' task-switching behavior and, consequently, their cognitive patterns, such as activity rhythms, pauses, and fluctuations in engagement, with sufficient accuracy to support user characterization and profiling. Importantly, the visibility of this traffic depends on the network context: local networks and ISPs can observe the original IP flow when no VPN is used, while a VPN encapsulates and encrypts the traffic, preventing these observers from accessing it, leaving only the VPN provider with full visibility.

The central contribution of this work is a general framework that establishes the viability of extracting task-switching dynamics from network traffic. In this methodology, we map patterns in packet-level metadata to a scalar summary, denoted by $\hat{\sigma}$ (with ground-truth σ). This scalar may represent the expected number of task switches in a time window, the average interval between switches, or any equivalent measure of switching dynamics.

We define a task switch as a transition in foreground focus, captured either by a change in the active window or a change in domain name within a browser. This operationalization is a pragmatic simplification: while it does not fully capture the cognitive complexity of context switching, it offers a reasonably reliable proxy for our study's objectives. By deriving $\hat{\sigma}$ directly from network traffic, we enable behavioral profiling without client-side instrumentation—thereby revealing the privacy risks we aim to quantify and mitigate.

3.2. Methodology overview

Fig. 1 summarizes the general methodology we propose, separating the training and inference pipelines. At a high level: (1) as the user interacts with the computer, packet-level traffic is generated and transmitted to their ISP's network; (2) these packets are captured, cleaned, and grouped into meaningful groups; (3) informative features are extracted from each group; (4) the features are fed to a trained ML model; and (5) the group-level outputs are aggregated to yield a final scalar summarizing the user's activity for a time period. We select the model after an exploration phase using both synthetic and real datasets to construct training and evaluation splits. In the remainder of this section, we describe the components of the general framework in more depth, step by step (with specific details in Section 4).

Packet capture. Network data is collected. We consider low-level user traffic activity in the form of *network packets*, the atomic units of data transmission on TCP/IP-based networks. A packet consists of an IP header and a payload. The payload may carry a UDP user datagram or a TCP segment, each of which has its own transport-layer header. For our analysis, we rely exclusively on header metadata from all layers, including the source and destination IP addresses in the IP header, the transport-layer protocol (UDP or TCP), the source and destination ports in the UDP or TCP header, and, in the case of TCP segments, control flags such as SYN, ACK, and FIN.

Packet cleaning. Raw traces are preprocessed to reduce noise. This includes discarding malformed or truncated packets with missing or inconsistent values, standardizing timestamp representations, dropping unused fields, and merging redundant fields, among other operations.

Packet grouping. Structure is imposed on the stream of clean packets. Packet traffic is bursty, temporally non-uniform, and shaped by stochastic protocol behavior. Learning a mapping from packets to a target is difficult, as the underlying data manifold is highly complex. We therefore group packets into higher-level blocks, such as time intervals or fixed-length packet windows, simplifying feature design, improving learnability, and reducing computational and memory costs. Although some temporal detail is lost, the representation becomes less noise-sensitive and more meaningful, improving performance.

Feature extraction. Once the packets are grouped, each group is transformed into a fixed-length feature vector. Drawing on networking domain knowledge, we extract features such as summary statistics and counts that capture meaningful traffic patterns, yielding a compact, tabular representation, bridging the gap between packets and structured model input.

ML inference. The extracted features are fed into a trained machine learning model. The specific modeling objective may vary depending on the implementation of the methodology (concrete examples discussed in the next section). Importantly, the model's output need not directly represent the final task-switch count (or equivalent targets); it may instead produce intermediate signals that are further reorganized in the subsequent aggregation step. Both the choice of model and the nature of its outputs are closely tied to earlier design decisions, particularly the strategies used for packet grouping and feature extraction.

Prediction aggregation. To improve interpretability and comparability, postprocessing can be applied to the model's outputs. For example, predictions may be merged or transformed to express task-switch counts per unit of time (if not already in this format) or converted into other intuitive metrics. This step also facilitates consistent evaluation across models with different output formats.

Training. To develop our models, we rely on both real and synthetic data, each consisting of network packet traces paired with timestamps marking task switches (or equivalent indicators). After collection, the data undergoes processing steps analogous to those in the inference pipeline. The ground-truth labels are then used to construct datasets suitable for supervised training and evaluation. Further details on data collection and dataset construction are provided in the next section. Model selection follows an iterative process of training, evaluation, and refinement, with the goal of achieving robust generalization while making efficient use of the available data.

4. Implementation and results

The general methodology outlined in Section 3 can be instantiated in several ways, depending on the strategies used for grouping packet data, modeling, and interpreting outputs. We introduce three approaches to implementing the general framework: *macroscopic*, *microscopic*, and *mesoscopic*, corresponding to different levels of granularity in treating network data: dataset-level, packet-level, and fixed-length time intervals, respectively.[1] Their implementation and performance

[1] The implementation code is available at the project's GitHub repository (Donaire, 2025).

are presented in Sections 4.2, 4.3, and 4.4, and compared in Section 4.5. Although the mesoscopic approach achieves the best overall performance, the other two are included for several reasons. First, their limitations were instrumental in shaping the final design: the macroscopic pipeline shows that coarse statistical summaries are insufficient, while the microscopic pipeline demonstrates practical challenges at the packet level. Second, they achieve objectives of independent interest (e.g., correlation with primary-service IP addresses or accurate packet-level discrimination) that may prove useful in other contexts. Third, presenting all three approaches highlights the diverse attack surface on user privacy and better illustrates the trade-offs between accuracy, interpretability, and computational cost, thereby sketching a richer landscape for future research. Finally, we emphasize that the macroscopic and microscopic approaches are deliberately presented as endpoints of the granularity spectrum: coarse dataset-level summarization and fine packet-level analysis. Their contrasting strengths and limitations help motivate the mesoscopic design and clarify the relevant trade-offs, even where their absolute explanatory power is limited.

4.1. Training data collection

The network activity data used to develop our models include both synthetic and real trace datasets. In both cases, the packets originate from genuine network connections and correspond to TLS traffic; the difference lies in the underlying generation process. Synthetic traces are produced by an automated controller that launches applications under controlled conditions, whereas real traces are collected from participants using their computers without intervention.

To generate synthetic network traces that simulate user behavior when switching between applications (e.g., instant messaging, video streaming, or online shopping), we create separate Docker containers (Merkel et al., 2014) for each type of network activity, automating access to such services through the internet. A complete list of the websites used to generate the data is available at the study's GitHub repository. Docker containers provide isolation, ensuring traffic from each application remains distinct and can be easily labeled by destination. This setup enables precise attribution of traffic to specific services. Finally, the traces from all containers are merged to recreate realistic usage patterns, capturing transitions between applications as they occur during actual usage.

To collect real data, we deployed a lightweight, non-intrusive data collection tool on participants' computers with informed consent, following the regulation of the UAB Research Ethics Committee (CERec) (Autonomous University of Barcelona, 2022). The study involved 22 participants (10 male and 12 female) aged 21–56 years (mean: 31.9, median: 30.5, min: 21, max: 56). Once installed, participants use their devices as they normally would, without specific constraints. The tool requires only brief setup and runs locally, launching two time-synchronized background processes: a packet sniffer based on Wireshark (Sanders, 2017) that records network traffic, and a focus logger that timestamps every change of foreground application window title. Across all participants, this procedure yielded approximately 106 h of packet captures and 11,506 recorded task-switch events.

4.2. Macroscopic approach: dataset-level statistics

Within the methodology introduced in the previous section, the macroscopic approach omits the packet grouping and ML inference steps and instead directly computes dataset-level features from network data.

Formally, given a dataset of network traces D, where the user has performed σ task-switches, we aim to identify an easily computable function $f : D \to \mathbb{R}$, such that $f(D)$, a single summary value (our $\hat{\sigma}$), is strongly correlated with σ. To this end, we evaluate a set of candidate functions f, each tested individually — from simple counts (e.g., number of unique IPs) to more involved heuristics — to determine which provides the strongest correlation.

Table 1
Macroscopic approach candidate functions $f(D)$.

Name	Description
UI (Unique IPs)	Number of distinct remote IP addresses observed. Proxy for diversity of services.
UP (Unique Ports)	Distinct source or destination TCP/UDP ports. Proxy for service heterogeneity.
S+AS SYNs	Count of TCP packets with SYN or SYN+ACK flags. New connections proxy.
Rare TCP Triplets	Frequency of low-prevalence 3-packet TCP flag sequences (e.g., [A,AS,AP]). Intended to grow with task diversity.
Send-Var Minima	Count of local minima in 250-packet rolling variance of the send/receive direction flag (added manually). Targets regime transitions.
Start-Pattern (10w)	Occurrence of 4-step pattern (UDP send, UDP receive, TCP SYN, TCP SYN/ACK) in any 10-packet window.
Start-Pattern (4c)	Same as above, but requires strict consecutive occurrence of the 4 steps. More specific.
Heavy-IP90	Counts IPs (recv. direction) with total byte volume above the 90th percentile.
Heavy-IPMean	Same as above, but uses mean volume as the threshold. More sensitive to skew.
Main-IP (learned)	Lightweight classifier that counts likely primary service IPs after filtering noise. See main text for details.

Dataset and metrics

To assess each candidate function, we use 84 synthetic datasets of 15 s each, grouped into seven mixing-level tiers. A tier is defined by a mixing percentage that specifies the degree of temporal overlap between tasks. Formally, given a sequence of task traces T_i with duration d_i (defined as the difference between the timestamps of the last and first packets of the trace), an overlap of $X\%$ implies that task T_{i+1} starts $(1 - X/100)d_i$ seconds after the start of task T_i. Each tier contains 12 datasets. Each dataset simulates a different number of task switches, ranging from 3 switches to 14 (see Section 4.1 for details on synthetic data). This setup yields meaningful yet tractable results, and the use of multiple mixing levels helps control for spurious effects introduced by the synthetic data generation process. We apply each candidate function to every dataset and, within each mixing tier, compute (i) the Pearson correlation coefficient r (Pearson, 1901) between the true number of task switches and the function outputs, and (ii) Spearman's rank correlation ρ (Spearman, 1961) between their respective rank orders. Averaging r and ρ across tiers yields concise measures of linear and monotonic association, respectively. And (iii), we report the absolute slope of an ordinary least squares line fitted to the mean function value as a function of mixing percentage, capturing any systematic bias as mixing increases. In addition, we perform a statistical significance test to compare the correlations obtained by the best-performing candidate functions. Because the correlations share the same ground-truth variable, Steiger's Z-test for dependent correlations (Steiger, 1980) is applied on the pooled datasets (N = 84) at a 95% confidence level ($\alpha = 0.05$).

Candidate functions

Table 1 describes each candidate function $f(D)$. Raw host-diversity proxies (UI and UP functions) tend to overestimate behavioral diversity

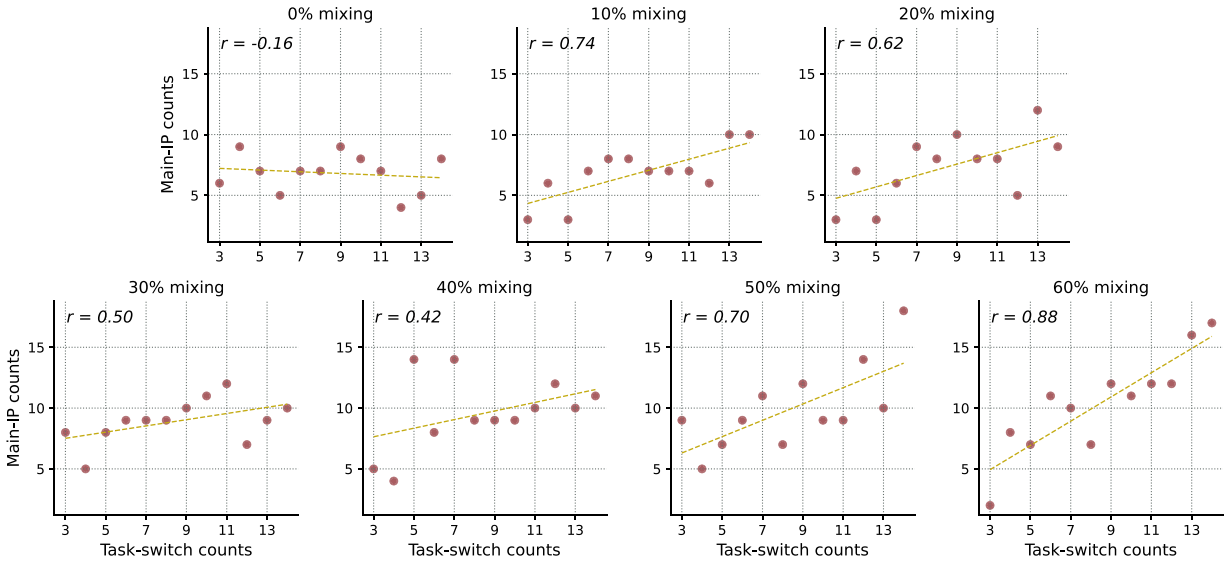

Fig. 2. Main-IP function values versus ground-truth task-switch counts, grouped by mixing level. Each subplot corresponds to one tier. Linear regression lines are overlaid, and Pearson's r values are shown per subplot.

because a single action (e.g., loading a page) often fans out to many auxiliary hosts such as content delivery networks (CDNs), analytics, or ads. The more elaborate *Main-IP* function aims to count only IP addresses likely to correspond to the primary services active in a trace. Specifically, we retain receive-direction packets with TCP flags indicating established transfers ("A" or "AP") and exclude large well-identified blocks associated with auxiliary web-service providers that support service requests, such as those handling analytics, advertising, or similar functions. For each remaining destination IP address in the dataset, we calculate two features: (i) maximum packet length and (ii) maximum 150-packet rolling mean of packet length (falling back to the overall mean when fewer than 150 packets are present). Using these two features, we analyze single web connection traces and manually identify the highest activity, highest throughput IP for each connection, labeling them as "Main-IP". Then, a lightweight logistic regression classifier is trained to predict whether an IP is Main-IP or not. The function value $f_{\text{Main-IP}}(D)$ is the number of predicted positives.

Results

Table 2 presents the performance of all functions (see Table 1), reporting their average linear and rank correlation with task-switch count, and how their outputs are affected by task mixing. To illustrate the behavior of the best-performing function, *Main-IP*, Fig. 2 shows its output across all mixture datasets, grouped by mixing level. Each subplot corresponds to one tier, comparing Main-IP predictions to ground-truth task-switch counts. A fitted linear regression line is overlaid, together with the corresponding Pearson correlation coefficient.

The best-performing candidate function is *Main-IP*, which is supported by Steiger's Z-test comparing their correlations with task-switch counts when contrasted with *Start (10w)* ($Z = 3.230$, $p = 0.001$) and *Start (4c)* ($Z = 2.374$, $p = 0.018$), the two other most promising functions exhibiting relatively high correlation and low mixing bias. These results indicate that *Main-IP* achieves a statistically stronger association with task-switch counts than these competing functions. In general, correlations with task-switch counts are modest (Pearson correlation peaking at 0.53), indicating that only a limited portion of variance is captured by these functions. A key failure mode is traffic fan-out: a single action can trigger requests to an arbitrary number of auxiliary hosts (CDNs, advertising, analytics), each with highly variable throughput. This distorts metrics, inflating host-diversity statistics (e.g., *UI*), or weakening regime-change detectors (e.g., *Send-Var Minima*). The more elaborate *Main-IP* function partially suppresses these effects, mildly improving results. However, further evaluations on real data show that its performance drops substantially, likely due to distribution shift (more homogeneous IP address behavior and markedly different feature ranges). In summary, macroscopic approaches are easily affected by the randomness inherent in network dynamics. These observations motivate moving to lower-level approaches that can exploit finer-grained packet information to control for these effects.

Table 2
Macroscopic functions performance: mean Pearson correlation $\bar{r}$, mean Spearman rank correlation $\bar{\rho}$ (computed across mixture tiers), and $|m|$, the absolute slope of mean function value vs. mixing percentage (lower is better). Bold indicates best per column.

Function	$\bar{r}$	$\bar{\rho}$	$\lvert m\rvert$
UI	−0.47	−0.39	0.015
UP	0.19	0.18	0.012
S+AS SYNs	0.18	0.19	0.012
Rare Triplets	0.14	0.09	0.018
Send-Var Min.	0.19	0.17	0.015
Start (10w)	0.18	0.18	0.007
Start (4c)	0.30	0.30	**0.007**
Heavy-IP90	0.08	0.09	0.012
Heavy-IPMean	−0.46	−0.41	0.015
Main-IP	**0.53**	**0.53**	0.008

4.3. Microscopic approach: packet-level segmentation

At the opposite end of the granularity spectrum, the *microscopic* approach segments traffic at the level of individual packets, iteratively deciding for each new packet x_t whether it belongs to an existing task or a new one.

Within the general framework of Section 3, the *packet grouping*, *feature extraction*, and *ML inference* stages form a closed loop. Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ denote the set of unique tasks identified up to the current moment, each represented by a buffer of previously assigned packets. When a new packet x_t arrives, we compute a feature vector $\phi(T_i \cup \{x_t\})$ for every T_i, where ϕ is a feature extraction function described in more detail below. This feature vector summarizes the joint characteristics of the candidate task and the candidate packet. A trained binary classifier then independently evaluates each pair (T_i, x_t), outputting a probability $p_i \in [0, 1]$ that the incoming packet belongs to that specific buffer. These

probabilities are unnormalized across tasks, as the set of possible tasks is open-ended. This also ensures that the decision thresholds defined below remain consistent regardless of the number of active tasks.

The decision rule is thresholded to control drift and fragmentation. We consider three cases:

- If $\max_i p_i < \alpha$, a new task T_{k+1} is opened with x_t as its first element.
- If $\max_i p_i \geq \beta$, the packet is assigned to $T_{\arg\max_i p_i}$.
- If $\alpha \leq \max_i p_i < \beta$ (the "gray zone"), the packet is omitted and added to a residual cluster, $T_{res} \notin \mathcal{T}$, which is unique.

After all packets are processed, the resulting set $\mathcal{T}$ represents the collection of distinct and unique tasks identified during the session. The aggregation step can then produce a final summary scalar $\hat{\sigma}$ by reconstructing the chronological sequence of packet assignments and counting the transitions between these distinct task buffers. However, because the reliability of this count depends fundamentally on the accuracy of the packet segmentation, the evaluation below focuses on the quality of the resulting clusters.

The features extracted by ϕ for each pair of task buffer and candidate packet ($T_i \cup \{x_t\}$) are handcrafted to capture meaningful differences and similarities between existing task packets and the new candidate. For example, we compute differences in payload size, the proportion of buffer packets sharing the same send/receive direction as the candidate, and whether the candidate's TCP flag appears in other packets of that buffer. For IP address information, which is typically more challenging to use directly, we extract similar features, such as whether the non-host IP address in the candidate packet appears in the buffer. In total, ϕ returns a feature set comprising more than 30 descriptors.

Model selection and optimization

At the core of this approach is a binary classifier that, given a pair $\{T_i, x_t\}$, estimates the probability that the candidate packet x_t belongs to task buffer T_i. To select a suitable learner, we evaluated a broad set of algorithms, including Logistic Regression, Random Forests, SVMs, XGBoost, LightGBM, Bernoulli Naive Bayes, and a feedforward neural network, on a dataset of curated synthetic pairs. This dataset included varying buffer sizes (1–50 packets) to simulate the accumulation of packets over time, and different websites (80) to capture task diversity. CatBoost consistently yielded the best results, achieving an F_1 score slightly above 0.90 at a 0.5 probability threshold. This performance relies on four optimization tweaks: (i) Sample rebalancing: the source and destination IPs of a new packet belonging to an existing task are often already present among the IPs of the packets in the buffer, allowing the model to learn a trivial IP-matching solution. To force the learner to detect less obvious connections (such as third-party resources), we oversampled positive pairs where the candidate packet's IP was not present in the task buffer; (ii) Feature enrichment by adding Autonomous System Number (ASN) features; (iii) Exclusion of large well-identified blocks associated with auxiliary web-service providers as in the macroscopic *Main-IP* function (Section 4.2); and (iv) Hyperparameter tuning, performed via scikit-learn's `RandomizedSearchCV`, which specifically optimized the `learning_rate`, tree `depth`, the `l2_leaf_reg`, and `border_count`.

Dataset and metrics

To evaluate the practical viability of the microscopic pipeline, we embed the CatBoost classifier into the full packet-assignment loop. Then, we run it on a synthetic trace that interleaves five application tasks over 12167 packets, similar to the datasets used in Section 4.1. Because the loop produces a partition of packets into predicted tasks, evaluation is cast as a clustering problem and measured with the *Adjusted Rand Index* (ARI). ARI is a pairwise agreement metric between cluster elements (packets, here), correcting the Rand Index for chance. It yields 1 when two partitions are identical and an expected value of 0 under random assignment; formally, letting n_{ij} be the number of items shared by true cluster i and predicted cluster j, $a_i = \sum_j n_{ij}$, $b_j = \sum_i n_{ij}$, and $n = \sum_{ij} n_{ij}$,

$$\mathrm{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \frac{\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}}{\binom{n}{2}}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \frac{\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}}{\binom{n}{2}}}.$$

While ARI measures overall agreement between predicted and true partitions, it cannot distinguish between, and is relatively insensitive to, over- and under-segmentation. To address this, we propose a new metric, the Overclustering (OC) metric, which quantifies the relative surplus of predicted clusters compared to the ground truth:

$$\mathrm{OC} = 100\,\frac{c_{\mathrm{pred}} - c_{\mathrm{true}}}{c_{\mathrm{true}}}\ (\%)$$

where c_{pred} and c_{true} denote the number of predicted and true clusters, respectively. As explained above, packets whose maximum assignment probability falls in the gray zone $\alpha \leq \max_i p_i < \beta$ are left unlabeled. We also track the proportion of such packets, referred to as "omitted".

Table 3
Microscopic approach performance: ARI, OC, and omitted-frame percentages for base and optimized CatBoost under varying decision probability thresholds. Bold indicates best per column.

Model (α/β)	ARI	OC (%)	Omitted (%)
Base (0.2/0.8)	0.386	1180	**10.9**
Base (0.1/0.9)	0.329	1380	19.2
Base (0.01/0.99)	0.364	180	93.4
Optimized (0.2/0.8)	**0.483**	500	12.1
Optimized (0.1/0.9)	0.445	400	26.1
Optimized (0.01/0.99)	0.466	**−40**	93.1

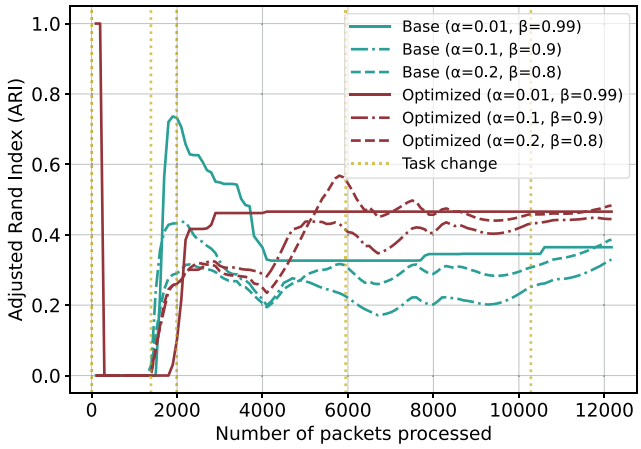

Fig. 3. Evolution of ARI during packet-by-packet processing in the microscopic approach, illustrating the stability of the quality of predicted task partitions as more task switches occur.

Results

We evaluate three decision-threshold pairs, $(\alpha, \beta) \in \{(0.20, 0.80), (0.10, 0.90), (0.01, 0.99)\}$, comparing the fully optimized classifier with its pre-optimization counterpart. Table 3 reports ARI, OC and omitted-packet rates.

To better illustrate the stability of the system in time, Fig. 3 plots ARI as packets stream in. These results show that behavioral signals can be extracted at the packet level but also reveal inherent limitations. The optimized CatBoost classifier consistently achieves higher ARI scores than the base model, reaching a peak of 0.483. This shows promising

discriminative power, while simultaneously indicating that many packets from the same task are misclassified into different tasks. Such misclassifications inevitably introduce cumulative errors into the packet-classification loop, leading to fragmentation and high OC. Tighter decision probability thresholds (α/β) considerably mitigate this issue, reducing OC and stabilizing ARI as packets are processed, but at the cost of omitting more packets (up to 93.4%). While omitting packets reduces segmentation quality, it provides noticeable gains in computational efficiency, although the approach remains resource-intensive. In summary, working at the finest granularity introduces challenges related to classification stability and computational costs due to the sheer volume of packets, suggesting that a middle-ground, mesoscopic approach may offer a better balance between resolution and robustness, minimizing the limitations observed at the extremes.

4.4. Mesoscopic approach: time intervals

Our third approach to implementing the general methodology, the mesoscopic approach, occupies a middle ground between the macroscopic and microscopic pipelines. Rather than relying on coarse dataset-level summaries or packet-level assignments, it groups packets into fixed-length time intervals, preserving enough temporal detail to capture switching dynamics while avoiding the instability and computational overhead of per-packet classification.

Formally, after packet cleaning, the traffic trace is segmented into consecutive intervals of duration t. For each interval I_j, a feature vector $\mathbf{x}_j \in \mathbb{R}^d$ is extracted from the packets in I_j. This vector encodes statistical descriptors of the interval traffic and serves as the structured input to a machine learning model f_θ. The model outputs a scalar prediction

$$\hat{\sigma}_j = f_\theta(\mathbf{x}_j),$$

which directly estimates the number of task switches in the interval. Unlike the macroscopic and microscopic pipelines, no further aggregation is required, since $\hat{\sigma}_j$ already represents the final inference target.

For each interval, we extract 185 handcrafted features, such as the number of unique source and destination ports, variance in packet payload, and Shannon entropy of IP addresses, along with various mathematical transformations. Unlike the macroscopic descriptors, these features are specifically designed for smaller time scales, rather than aggregate packet traces, and fundamentally aim to distinguish intervals containing a task switch from those that do not. We evaluated four feature sets: the full 185 features, a 40-feature set selected via recursive feature elimination (RFE), a 65-feature set created by thresholding for correlation and variance, and a 40-feature subset of the latter, reduced using principal component analysis (PCA).

The remainder of this section analyses the mesoscopic pipeline through a sequence of focused experiments. In *Base model selection*, we evaluate several machine learning models across different feature sets and interval lengths to identify the most effective baseline configuration. In *Imbalance*, we examine the impact of the strong class imbalance caused by the predominance of intervals without task switches and test several imbalance-aware training strategies. In *Hyperparameters and architecture*, we explore architectural variations and hyperparameter adjustments of the best-performing model, the Temporal Convolutional Network (TCN), to better understand the factors influencing its performance. In *Network-active and network-inactive tasks*, we investigate how restricting the prediction target to tasks that inherently generate network traffic affects feature correlations and model accuracy. Finally, in *Generalization*, we evaluate the ability of the TCN model to generalize across users through cross-user training and testing experiments.

Base model selection

From a modeling perspective, predicting the number of task switches per interval can be approached in multiple ways, each with advantages and limitations. It can be posed as a regression problem, directly estimating switch counts, or as a classification problem, treating discrete switch counts (e.g., 0 vs. ≥ 1) as categories, which is natural when counts are low, particularly with short windows. The formulation can also be temporal, leveraging sequential dependencies across intervals, or non-temporal, relying only on interval-level statistics. A central difficulty, for instance, emerges with short intervals (e.g., under 10 s), where the signal is sparse and most intervals contain no switches, leading to class imbalance and zero-inflation, especially for classification. To explore this design space, we evaluate five base models. Four of them — XGBoost Regressor, Random Forest Regressor, Random Forest Classifier, and K-Nearest Neighbors — generally work well with tabular data and provide reliable baselines. The fifth, a custom non-causal Temporal Convolutional Network (TCN), explicitly exploits sequential dependencies across consecutive intervals. The remainder of this section reports experiments with these models, examining feature set selection, interval lengths, imbalance mitigation, hyperparameter tuning, data filtering, and generalization.

We evaluate all combinations of the five base models, four feature sets, and interval lengths of 1 s, 2 s, 5 s, and 10 s. Ensuring comparability across these heterogeneous settings — including both regression and classification formulations and different interval lengths — posed a methodological challenge. To address this, we randomly selected 43 one-hour splits from the real dataset and reserved the rest for other experiments. For each configuration, models were trained on 42 splits and tested on the remaining one using a leave-one-out approach. Interval-level predictions $\hat{\sigma}_j$ were then aggregated within each split S_k to estimate the total number of task switches, denoted $\hat{\Omega}_k = \sum_{j \in S_k} \hat{\sigma}_j$, with ground truth $\Omega_k = \sum_{j \in S_k} \sigma_j$.

Performance is measured at the split level using two complementary metrics. First, mean absolute error (MAE):

$$\text{MAE} = \frac{1}{N} \sum_{k=1}^{N} |\hat{\Omega}_k - \Omega_k|,$$

where $N = 43$ is the number of evaluation splits. Second, the Pearson correlation coefficient between predicted and true totals:

$$r = \frac{\sum_{k=1}^{N} (\hat{\Omega}_k - \overline{\hat{\Omega}})(\Omega_k - \overline{\Omega})}{\sqrt{\sum_{k=1}^{N} (\hat{\Omega}_k - \overline{\hat{\Omega}})^2} \sqrt{\sum_{k=1}^{N} (\Omega_k - \overline{\Omega})^2}}, \tag{1}$$

which quantifies the linear association between predictions and ground truth independently of scale. Together, these metrics assess both absolute accuracy (MAE) and relational fidelity (Pearson's r) across configurations.

The results, presented in Fig. 4, show that the full feature set consistently improves model performance. Regarding interval length, 10-second windows — and especially 5-second windows — perform better than shorter ones. Among all models, the TCN and XGBoost achieve the best results when using all features with 5-s intervals. While the TCN achieves the best absolute performance metrics (MAE = 24.91, Pearson's $r = 0.634$), a statistical analysis across the 43 evaluation splits shows no significant difference between the TCN and XGBoost (Wilcoxon W = 457.0, $p = 0.85$ for MAE; Steiger's Z = 1.061, $p = 0.29$ for Pearson's r). Despite this statistical parity, we select the TCN as our primary model for the remainder of the study because of its stronger inductive bias for sequential data.

Imbalance

The vast majority of intervals contain no task switches: for example, 87% of 10-s intervals and 92% of 5-s intervals. Such imbalance is known to hinder model performance (Japkowicz and Stephen, 2002). To mitigate this, we train a TCN classifier (rather than regressor) with imbalance-aware strategies. We define the task as a binary classification

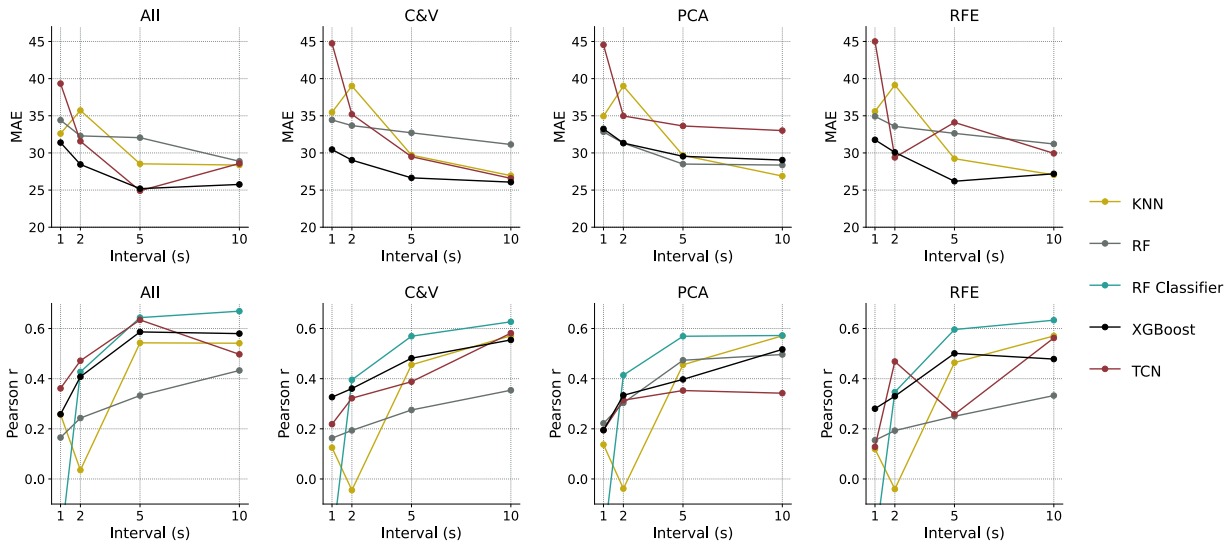

Fig. 4. Performance of mesoscopic base models: MAE and Pearson's r across feature sets and interval sizes. Models include XGBoost Regressor, Random Forest Regressor (RF), Random Forest Classifier (RF Classifier), K-Nearest Neighbors (KNN), and Temporal Convolutional Network (TCN). MAE results for RF Classifier, which are substantially higher than the plotted range, are omitted to preserve visual clarity.

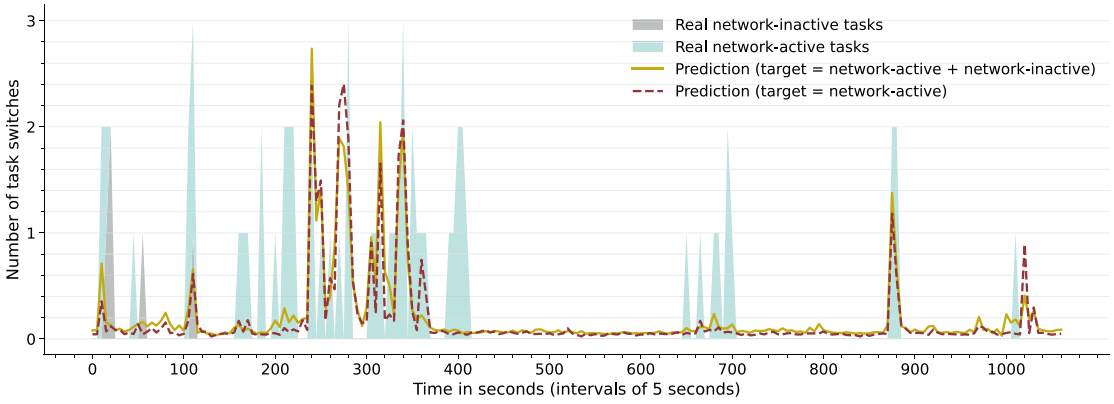

Fig. 5. Standard and target-modified TCN predictions for a representative test split spanning approximately 200 five-second intervals.

distinguishing between intervals with zero switches and those with one or more, serving as a proxy for the total count given the rarity of multiple switches per interval. The imbalance-aware strategies include focal loss (Lin et al., 2017), weighted loss (Elkan, 2001), and undersampling of zero-switch intervals. However, none of these variants produced meaningful gains over the baseline. This suggests that either the imbalance is difficult to mitigate or that performance bottlenecks stem from other challenges, such as signal sparsity or limited feature separability.

Hyperparameters and architecture

To refine and understand the best-performing base model, the TCN, we experiment with hyperparameters and architectural choices. Reducing the number of hidden channels (the width of intermediate convolutional layers, controlling the number of learnable features per layer) from 48 to 24 improves MAE (+0.2%, reaching 24.97) and Pearson correlation (+1.6%, reaching 0.644), while other adjustments, such as optimizer changes, have a negligible effect. Architectural ablations show that performance drops when switching from non-causal to causal convolutions or when replacing dilated convolutions with standard ones. Removing residual connections does not degrade performance, suggesting that they may be less critical in this setting.

Network-active and network-inactive tasks

A further axis of analysis concerns the nature of the tasks themselves. Not all tasks generate network activity equally: while actions such as loading streaming applications necessarily involve communication with remote servers, others-like typing in a local text editor-produce little or no network activity. Despite their near-absence of packets, these *network-inactive* tasks are still logged by the data collection tool and incorporated into training. This is particularly challenging for the model, which is asked to predict task switches in intervals with no meaningful signal.

We hypothesize that excluding such tasks can improve performance, even at the cost of disregarding a subset of activities. As a practical heuristic, we restrict the dataset to a set of *network-active* activities that inherently generate observable traffic: email, instant messaging, voice and video calls, video and music streaming, social media interaction, e-commerce, and general web browsing.

To test this hypothesis, we construct a modified dataset with the same traffic but with adjusted target labels that count only switches between the network-active activities defined above. In our dataset, 75% of recorded task switches across users correspond to network-active tasks. While modern applications are increasingly network active in practice, in our experiments we explicitly retain and evaluate network

inactive cases to highlight the limitations of traffic based inference under these conditions.

Using the modified dataset, we first compare feature-target correlations under the two target variants. Features show stronger correlations with the network-active target, with a median 10.1% Pearson correlation increase, indicating network-active switches provide clearer traffic signal.

We then retrain the TCN model using the modified dataset. Pearson correlation with network-active target improved by 3.1% compared to the baseline TCN, showing that focusing on network-active tasks improves performance at the cost of omitting a selected group of tasks. We do not report MAE because the exclusion of network-inactive tasks reduces the total count of switches, effectively shrinking the scale of the target variable. Since MAE is not scale-invariant, comparing it to previous values would be misleading.

To provide a qualitative view of this difference, Fig. 5 presents predictions for a representative test split spanning roughly 200 five-second intervals. The ground truth is shown as stacked areas separating network-active and network-inactive tasks, alongside the outputs of both TCN models. The network-active TCN captures network-active task switches more accurately—for instance, around seconds 330 and 1010. The baseline TCN better detects some network-inactive switches (e.g., second 0, but misses others (e.g., second 30). Both models occasionally underestimate task-switch counts, most notably around seconds 400 and 680. We hypothesize that these underestimations result from preloaded pages or rapid tab switches.

Task switches involving network-inactive activities produce little or no detectable signal and account for 25% of switches in our dataset. Despite this, focusing on network-active tasks improves feature-target correlation and model performance, confirming our hypothesis.

Generalization

A final set of experiments examines the generalization capacity of the TCN model. We selected four users who shared overlapping activities, such as browsing and messaging, but also showed individual differences: one frequently watched online videos, another made long video calls, a third played games, and the fourth primarily browsed or used text editors. To test cross-user prediction, we designed two setups: (i) training on one user and testing on each of the four in turn, and (ii) training on three users and evaluating on the fourth, rotating across users.

In the first experiment, Pearson correlation (as defined in (1)) dropped by an average of 51.8% relative to intra-user baselines. However, examining individual user pairs revealed substantial variance in this degradation. Although some mismatches were more pronounced — for example, models trained on the user that video-called frequently consistently underestimated task-switch counts when applied to the user whose activity was mainly browsing — other user combinations exhibited considerably smaller drops, on the order of 20% to 30%. In the most favorable cross-user case, the degradation was only 17.3%, indicating that the model can retain meaningful predictive performance even when the adversary cannot perfectly replicate the victim's setting. The second experiment showed that mixing data from multiple users attenuates this problem, with average degradation decreasing to 43.8%.

These findings are particularly relevant for real-world deployments, where attackers rarely possess training data for their specific targets. Consequently, cross-user generalization serves as an important indicator of the practical viability of the attack. While the observed performance degradation highlights the impact of user-specific behavioral differences, the mean degradation alone does not fully characterize the victim's risk: from the victim's perspective, the most relevant case is the smallest degradation they could face, since they cannot know in advance whether their behavior resembles that of an attacker's available training users. The results therefore show that cross-user mismatch reduces attack performance on average, but does not eliminate the threat. They also demonstrate that degradation mitigation is possible. If target-specific data is unavailable, selecting training data from individuals with similar routines can reduce the performance mismatch. Even when such targeted selection is unfeasible, simply increasing the volume and behavioral diversity of the training dataset inherently improves overall generalization.

4.5. Approach comparison

In this section, we compare the microscopic, macroscopic and mesoscopic approaches, introduced in Section 4.2, 4.3, and 4.4, respectively. These approaches, in turn, are different implementations of the general methodology for inferring user task-switching dynamics from network data introduced in Section 3.

This comparison is intrinsically challenging: although all three approaches share the goal of extracting meaningful behavioral information from network data, they operate at different data granularity levels and pursue distinct modeling formulations. For this reason, we report quantitative metrics that are specific to the modeling formulation and data scale of each approach rather than enforcing a single unified metric across all methods. Moreover, the objective of this comparison is not to identify a fully optimized state-of-the-art architecture, but rather to evaluate whether different modeling granularities are capable of capturing task-switching behavioral signals from network traffic metadata. To make the comparison both relevant and informative, we provide a comparative table (Table 4) that reports not only quantitative metrics but also additional relevant dimensions that we consider important, as well as an overall qualitative assessment of performance. Accordingly, we intentionally rely on well-established and interpretable machine learning models to assess the existence and robustness of the underlying behavioral signal before exploring more sophisticated architectures in future work.

- **Training speed**: The macroscopic and microscopic approaches use lightweight models, while the mesoscopic approach employs a neural network that requires significantly longer training (several minutes in our experiments).
- **Inference speed**: Once trained, macroscopic and mesoscopic methods provide near-instantaneous predictions. The microscopic approach, however, as implemented here, relies on an internal loop with multiple predictions per packet, making it considerably slower (several minutes per configuration in Table 3).
- **Improvement potential**: The macroscopic approach is constrained by coarse summaries that overlook important fluctuations. The microscopic pipeline suffers from computational overhead and instability in long sequences. The mesoscopic approach strikes a balance between resolution and tractability, leaving room for improvement without major limitations.
- **Profiling potential**: The macroscopic approach enables only coarse profiling. The microscopic approach produces fine-grained predictions that can reveal subtle behaviors and, if further improved, could surpass the current capabilities of the mesoscopic approach. The mesoscopic approach also provides valuable insight into task-switching dynamics but is constrained by the temporal interval length, which can be shortened only at the cost of reduced performance.
- **Quantitative performance**: Both macroscopic and microscopic pipelines show that behavioral information can be extracted, but they remain fragile when parameters change. In contrast, the mesoscopic approach consistently provides good results even under varying conditions.

Overall, the mesoscopic approach shows a clear advantage for extracting behavioral signals as task-switch counts from network activity, addressing many limitations of the macroscopic and microscopic pipelines. While challenges remain, it provides the most balanced trade-off, combining robustness, strong performance, high profiling potential, room for improvement, and fast inference.

Table 4
Comparison of the macroscopic, microscopic, and mesoscopic approaches across relevant dimensions.

Approach	Best model/ function	Training speed	Inference speed	Improvement potential	Profiling potential	Quantitative performance	Qualitative assessment
Macroscopic	Main-IP function	High	High	Low	Medium	$\bar{r} = 0.53$ $\lvert m \rvert = 0.008$	Medium
Microscopic	CatBoost optimized	Medium	Low	Medium	High	ARI = 0.466 OC = −40%	Low
Mesoscopic	TCN 5 s intervals 185 features	Low	High	High	High	$r = 0.644$ MAE = 24.97	High

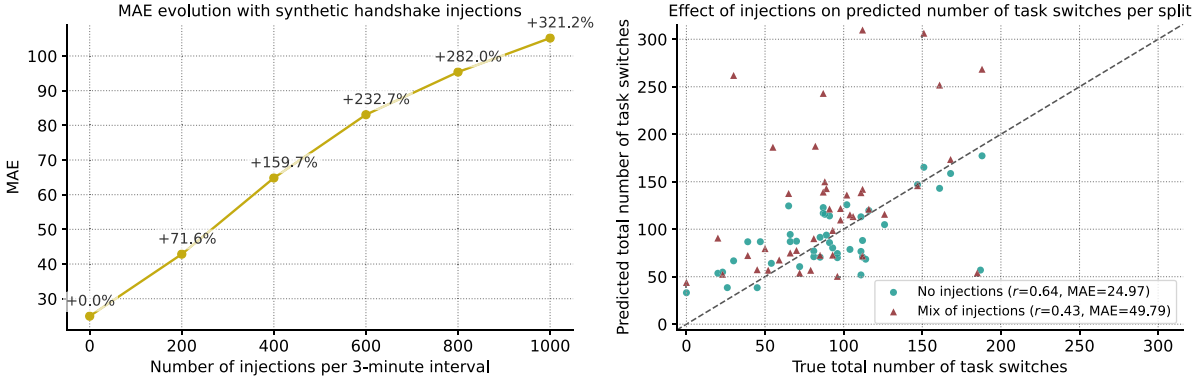

Fig. 6. Mitigation results on the mesoscopic TCN model. Left: mean absolute error (MAE) as a function of the number of injections per 3-minute interval. Right: predicted versus true task-switch counts for one-hour splits, with and without randomized injection levels.

5. Mitigation

To address the emerging privacy risk posed by the inference of task-switching dynamics from network traffic, we propose a mitigation strategy designed to disrupt this behavioral leakage. The challenge is to reduce the models' predictive power without excessive overhead or degraded user experience. In this section, we present a mitigation technique that balances privacy preservation with performance constraints, ensuring that the added noise effectively masks behavioral patterns while maintaining the integrity of networked applications. The objective of this mitigation is not to provide a universally robust defense against all possible adaptive adversaries, but rather to demonstrate that the identified behavioral leakage channel can be meaningfully disrupted through traffic perturbation. Accordingly, the proposed mechanism should be interpreted as a proof-of-concept privacy defense intended to establish feasibility and motivate future research on stronger adversarial models and more sophisticated mitigation strategies.

The mitigation strategy we propose consists of injecting additional TCP short connections into network traffic to obscure task-switching signals. It is guided by feature-importance and correlation analyses showing that our models rely heavily on connection-initiation proxies—such as the diversity of remote IPs or ports and counts of TCP SYN or SYN-ACK flags. From the user's perspective, the strategy involves a lightweight application that periodically makes short, plausibly legitimate connections to remote servers, chosen to prevent easy filtering by an attacker. To demonstrate its effectiveness, we conduct experiments in which, for each few-minute window, we synthesize triplets of packets (SYN, SYN-ACK, ACK). Their endpoints, directions, lengths, ports, and inter-arrival gaps are sampled from empirical distributions of the preceding window, and the triplets are inserted into the subsequent one with time stamps spaced according to observed differences. The number of injected short connections per window is configurable to vary over time, preventing model predictions from becoming a trivial multiple of the baseline.

Results

We evaluate the mitigation strategy on the mesoscopic TCN model (Section 4.4). Synthetic short connections are generated from traffic statistics computed over 3-minute windows, and evaluation follows the same leave-one-out procedure with 1-hour splits as in Section 4.4. We conduct two experiments: (i) injecting a fixed number of synthetic short connections into every split, progressively increasing the rate to test whether predictions can be systematically distorted, and (ii) injecting a variable number of short connections per split to assess whether the mitigation can make predictions not only inaccurate but also uncorrelated.

Fig. 6 shows the results of applying the mitigation strategy to the TCN. In the fixed-injection setting (left), MAE rises steadily, reaching 321.2% above baseline with 1000 injections per interval, confirming that synthetic short connections systematically distort predictions. In the variable-injection setting (right), MAE also increases and Pearson correlation drops by 33%, indicating also a disruption in linear association, making the underlying behavioral signal harder to recover.

We also assess the generalization of the mitigation strategy to our two other approaches by adapting it to their datasets and processing steps. In the macroscopic approach (Main-IP function), introducing 250 injections per synthetic dataset decreases both $\bar{r}$ and $\tilde{\rho}$ from 0.53 to 0.49 (−7.5%), while $|m|$ decreases by 1.3%. In the microscopic approach (Optimized model, $\alpha = 0.01$, $\beta = 0.99$), making 250 total injections decreases the ARI from 0.466 to 0.049 (−89%) and shifts the OC from −40% to 50%, while the percentage of omitted packets remains unchanged at 93.1%.

These results demonstrate that injecting short connections effectively disrupts task-switch inference, substantially reducing model performance. The overhead for the lightweight background application described above would be minimal; even at intensities comparable to our experiments (e.g., 1000 header-only packets every 3 min), the additional traffic would remain marginal.

6. Conclusions

This study introduced and examined a novel privacy threat: the inference of task-switching dynamics from network traffic metadata.

Our main contribution is an ML-based methodology that demonstrates that network metadata is vulnerable to exploitation, as it can be used to reconstruct behavioral rhythms and thereby reveal sensitive information about users' attention and routines. We compared three different approaches to implementing this methodology, each operating at a different data granularity level. All three approaches showed inference power, underscoring that the attack surface is broad. The mesoscopic approach provided the most balanced trade-off, combining solid performance with robustness and computational tractability. Finally, we proposed and evaluated a mitigation strategy based on injecting additional TCP short connections into the network traffic, which effectively disrupted inference while introducing marginal overhead.

CRediT authorship contribution statement

Alejandro Donaire: Validation, Software, Investigation, Formal analysis, Data curation. **Carlos Borrego:** Writing – original draft, Validation, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Marc Juarez:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by the Spanish Ministry under Grant PID2021-125962OB-C33; by the Catalan Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR) under Grant SGR2021-00643, and by the Plan de Recuperación, Transformación *y* Resiliencia funded with Next Generation EU funds through the project DANGER INCIBE-C062/23.

Data availability

Data will be made available on request.

References

Alotibi, G., Clarke, N., Li, F., Furnell, S., 2016. User profiling from network traffic via novel application-level interactions. pp. 279–285.

Alwhbi, I.A., Zou, C.C., Alharbi, R.N., 2024. Encrypted network traffic analysis and classification utilizing machine learning. Sensors 24 (11), 3509.

Auliya, S., Nugroho, L., Setiawan, N.A., 2021. A review on smartphone usage data for user identification and user profiling. Commun. Sci. Technol. 6, 25–34.

Autonomous University of Barcelona, 2022. Regulation on research ethics. Approved by the Governing Council on September 29, 2022, modified by Governing Council Agreement of September 25, 2025. [Online]. Available: https://www.uab.cat/doc/Reglament_etica_recerca.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. In: KDD 201916, ACM, pp. 785–794, [Online]. Available: http://dx.doi.org/10.1145/2939672.2939785.

Chittaranjan, G., Blom, J., Gatica-Perez, D., 2011. Who's who with big-five: Analyzing and classifying personality traits with smartphones. In: 2011 15th Annual International Symposium on Wearable Computers. pp. 29–36.

Corbetta, M., Shulman, G.L., 2002. Control of goal-directed and stimulus-driven attention in the human brain. Nature Rev. Neurosci. 3 (3), 201–215, [Online]. Available: https://www.nature.com/articles/nrn755.

Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory 13 (1), 21–27.

de Montjoye, Y.-A., Hidalgo, C.A., Verleysen, M., Blondel, V.D., 2013. Unique in the crowd: The privacy bounds of human mobility. Sci. Rep. (ISSN: 2045-2322) 3 (1), 1376.

Donaire, A., 2025. Attention-leak-paper-code. https://github.com/Aledonairesa/attention-leak-paper-code. (Accessed: 02 September 2025).

Dong, W., Yu, J., Lin, X., Gou, G., Xiong, G., 2025. Deep learning and pre-training technology for encrypted traffic classification: A comprehensive review. Neurocomputing 617, 128444.

Elkan, C., 2001. The foundations of cost-sensitive learning. In: International Joint Conference on Artificial Intelligence, vol. 17, (1), Lawrence Erlbaum Associates Ltd, pp. 973–978.

Herrmann, C.S., Knight, R.T., 2001. Mechanisms of human attention: event-related potentials and oscillations. Neurosci. Biobehav. Rev. (ISSN: 0149-7634) 25 (6), 465–476.

Iqbal, S., Horvitz, E., 2007. Disruption and recovery of computing tasks: Field study, analysis, and directions. In: In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI'07, ACM, pp. 677–686, [Online]. Available: https://www.microsoft.com/en-us/research/publication/disruption-recovery-computing-tasks-field-study-analysis-directions/.

Jamadar, S., Thienel, R., Karayanidis, F., 2015. Task switching processes. Brain Mapp.: An Encycl. Ref. 3, 327–335,

Japkowicz, N., Stephen, S., 2002. The class imbalance problem: A systematic study. Intell. Data Anal. 6 (5), 429–449.

Juarez, M., Afroz, S., Acar, G., Diaz, C., Greenstadt, R., 2014. A critical evaluation of website fingerprinting attacks. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 263–274.

Judd, T., 2015. Task selection, task switching and multitasking during computer-based independent study. Australas. J. Educ. Technol. 31, 193–207.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., 2017. Lightgbm: a highly efficient gradient boosting decision tree. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS '17, Curran Associates Inc., Red Hook, NY, USA, pp. 3149–3157.

Keller, A.S., Davidesco, I., Tanner, K.D., 2020. Attention matters: How orchestrating attention may relate to classroom learning. CBE—Life Sci. Educ. 19 (3), fe5, PMID: 32870089.

Kissel, E., 2006. Profiling and Clustering Internet Hosts. Tech. Rep.

Lea, C., Vidal, R., Reiter, A., Hager, G.D., 2016. Temporal convolutional networks: A unified approach to action segmentation. [Online]. Available: https://arxiv.org/abs/1608.08242 arXiv:1608.08242.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2980–2988.

Mathews, N., Holland, J.K., Oh, S.E., Rahman, M.S., Hopper, N., Wright, M., 2023. Sok: A critical evaluation of efficient website fingerprinting defenses. In: 2023 IEEE Symposium on Security and Privacy. SP, IEEE, pp. 969–986.

McDonald, G., Or, J.B., 2025. Whisper leak: a side-channel attack on large language models. arXiv preprint arXiv:2511.03675.

Mehrotra, R., Bhattacharya, P., Yilmaz, E., 2016. Characterizing users' multi-tasking behavior in web search. In: Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval. CHIIR '16, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450337519, pp. 297–300.

Merkel, D., et al., 2014. Docker: lightweight linux containers for consistent development and deployment. Linux J 239 (2), 2.

Mishra, J., Gazzaley, A., 2012. Attention distributed across sensory modalities enhances perceptual performance. J. Neurosci. 32 (35), 12294–12302.

Monsell, S., 2003. Task switching. Trends Cogn. Sci. 7 (3), 134–140.

Monsell, S., 2021. Control of mental processes. In: Unsolved Mysteries of the Mind. Psychology Press, pp. 93–148.

Montjoye, Y.-A., Quoidbach, J., Robic, F., Pentland, A., 2013. Predicting personality using novel mobile phone-based metrics. pp. 48–55. http://dx.doi.org/10.1007/978-3-642-37210-0_6.

Msadek, N., Soua, R., Engel, T., 2019. Iot device fingerprinting: Machine learning based encrypted traffic analysis. In: 2019 IEEE Wireless Communications and Networking Conference. WCNC, IEEE, pp. 1–8.

Ophir, E., Nass, C., Wagner, A.D., 2009. Cognitive control in media multitaskers. Proc. Natl. Acad. Sci. 106 (37), 15583–15587.

Pearson, K., 1901. LIII. On lines and planes of closest fit to systems of points in space. Lond. Edinb. Dublin Philos. Mag. J. Sci. 2 (11), 559–572.

Polk, T.A., Drake, R.M., Jonides, J.J., Smith, M.R., Smith, E.E., 2008. Attention enhances the neural processing of relevant features and suppresses the processing of irrelevant features in humans: a functional magnetic resonance imaging study of the stroop task. J. Neurosci. 28 (51), 13786–13792.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A., 2018. CatBoost: unbiased boosting with categorical features. Adv. Neural Inf. Process. Syst. 31.

Ralph, B.C., Thomson, D.R., Cheyne, J.A., Smilek, D., 2014. Media multitasking and failures of attention in everyday life. Psychol. Res. 78, 661–669.

Rimmer, V., Preuveneers, D., Juarez, M., Van Goethem, T., Joosen, W., 2018. Automated website fingerprinting through deep learning. In: 25th Annual Network and Distributed System Security Symposium. The Internet Society.

Rogers, R., Monsell, S., 1995. Costs of a predictable switch between simple cognitive tasks. J. Exp. Psychol. [Gen.] 124, 207–231.

Sanders, C., 2017. Practical packet analysis: Using Wireshark to solve real-world network problems. no starch Press.

Seneviratne, S., Seneviratne, A., Mohapatra, P., Mahanti, A., 2014. Your installed apps reveal your gender and more!. In: Proceedings of the ACM MobiCom Workshop on Security and Privacy in Mobile Environments. SPME '14, Association for Computing Machinery, New York, NY, USA, pp. 1–6.

Siby, S., Juarez, M., Diaz, C., Vallina-Rodriguez, N., Troncoso, C., 2019. Encrypted DNS → privacy? a traffic analysis perspective. arXiv preprint arXiv:1906.09682.

Sirinam, P., Imani, M., Juarez, M., Wright, M., 2018. Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 1928–1943.

Spearman, C., 1961. The proof and measurement of association between two things.

Steiger, J.H., 1980. Tests for comparing elements of a correlation matrix.. Psychol. Bull. 87 (2), 245.

Stothart, C., Mitchum, A., Yehnert, C., 2015. The attentional cost of receiving a cell phone notification. J. Exp. Psychol. [Hum. Percept.] 41 (4), 893.

Taylor, V.F., Spolaor, R., Conti, M., Martinovic, I., 2017. Robust smartphone app identification via encrypted network traffic analysis. IEEE Trans. Inf. Forensics Secur. 13 (1), 63–78.

Tu, Z., Li, R., Li, Y., Wang, G., Wu, D., Hui, P., Su, L., Jin, D., 2018. Your apps give you away: Distinguishing mobile users by their app usage fingerprints. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 2 (3).

Uncapher, M.R., Lin, L., Rosen, L.D., Kirkorian, H.L., Baron, N.S., Bailey, K., Cantor, J., Strayer, D.L., Parsons, T.D., Wagner, A.D., 2017. Media multitasking and cognitive, psychological, neural, and learning differences. Pediatrics 140 (Supplement_2), S62–S66.

Vapnik, V., 2013. The Nature of Statistical Learning Theory. Springer science & business media.

Verde, N.V., Ateniese, G., Gabrielli, E., Mancini, L.V., Spognardi, A., 2014. No nat'd user left behind: Fingerprinting users behind NAT from NetFlow records alone. In: 2014 IEEE 34th International Conference on Distributed Computing Systems. pp. 218–227.

Ward, A.F., Duke, K., Gneezy, A., Bos, M.W., 2017. Brain drain: The mere presence of one's own smartphone reduces available cognitive capacity. J. Assoc. Consum. Res. 2 (2), 140–154.

Welke, P., Andone, I., Bł aszkiewicz, K., Markowetz, A., 2016. Differentiating smartphone users by app usage.

Yang, Y.C., 2010. Web user behavioral profiling for user identification. Decis. Support Syst. (ISSN: 0167-9236) 49 (3), 261–271, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923610000497.

Yeykelis, L., Cummings, J.J., Reeves, B., 2014. Multitasking on a single device: Arousal and the frequency, anticipation, and prediction of switching between media content on a computer. J. Commun. (ISSN: 0021-9916) 64 (1), 167–192.

Alejandro Donaire Salvador is a researcher in the Department of Information and Communications Engineering at the Autonomous University of Barcelona (UAB). He holds a degree in Computational Mathematics and Data Analytics, as well as a master's degree in Computer Vision, both from UAB. His professional experience includes work in the Department of Computer Architecture and Operating Systems at UAB, and an internship as a Data Scientist at ClearPeaks. His interests lie in the application of Machine Learning and Deep Learning to real-world problems.

Carlos Borrego Iglesias is a teacher and researcher at the Department of Information and Communications Engineering, Autonomous University of Barcelona. He received his degree in Computer Science from the Polytechnic University of Madrid (UPM). His professional experience includes work at CERN (Geneva, Switzerland), CASPUR La Sapienza (Rome, Italy), and research centers PIC and IFAE in Barcelona. He completed his Ph.D. at the Autonomous University of Barcelona. His current research interests focus on privacy-preserving network protocols with a Machine Learning perspective.

Marc Juarez is a Lecturer in Cyber Security and Privacy at the University of Edinburgh's School of Informatics. Marc obtained his Ph.D. from the KU Leuven, under an FWO Ph.D. fellowship. At the KU Leuven, Marc worked in the Computer Security and Industrial Cryptography (COSIC) group, where he applied machine learning techniques to a wide range of security and privacy problems. In particular, Marc's dissertation studies the application of machine learning to web traffic fingerprinting attacks.