TELESAFE: Monitoring Energy Consumption for Work–Life Boundaries in Telework

Haoying Zhang
Inria, INSA CVL
haoying.zhang@inria.fr

Nicolas Anciaux
Inria, INSA CVL, U. Paris Saclay
nicolas.anciaux@inria.fr

Benjamin Nguyen
INSA CVL, Inria
benjamin.nguyen@insa-cvl.fr

Jose Maria de Fuentes
U. Carlos III de Madrid
jfuentes@inf.uc3m.es

***Abstract*—Since COVID-19, teleworking has become widespread, raising concerns about employees' work-life balance and well-being. We demonstrate $TELESAFE$, a lightweight, fully local and unsupervised system for detecting work-life *boundary crossings* (breaks during work time, overworking during personal time) based on household electricity consumption time series. Unlike traditional monitoring tools, $TELESAFE$ does not rely on machine learning, cloud services, or additional sensors. Instead, it leverages already available data from smart meters commonly deployed in homes, ensuring full user privacy. We showcase two use cases (breaks, overworking) and present a self-monitoring interface for teleworkers.**

***Index Terms*—Time series, Telework, Energy consumption.**

Code Availability: The source code and video are available at https://gitlab.inria.fr/haoying.zhang/boundary-crossing-detection-interface.git and https://iww.inria.fr/telesafedemo/fr/.

I. Introduction

Since COVID-19 pandemic, teleworking has firmly established itself as a new norm in many companies. According to FlexJobs[1], it is a well accepted work modality by employees, with 65% expressing a preference for full remote work. It contributes to productivity, as teleworkers are 35% more productive, and 73% declare working outside their regular hours. However, this shift has introduced significant health concerns for employees, as widely identified in working health research [1]. A major stress factor is boundary crossings, i.e., when professional and personal activities intrude on each other [2], [3]. Short breaks during work hours may promote recovery and focus, but too frequent interruptions or work spilling into personal time can cause stress [3].

Traditionally, such risks were addressed through in-person workplace inspections [4]. However, in teleworking contexts, external interventions are limited. This creates a pressing need for self-monitoring tools that enable teleworkers to independently assess and adjust their work-life balance, in alignment with privacy rights.

We introduce such a system in this work, with a strong emphasis on *privacy by design*. $TELESAFE$ processes all data entirely **locally**, without transmitting or storing any information externally. It is based on a **fully unsupervised algorithm**, requiring no training data, annotation, or model personalization. Unlike many existing approaches that rely on machine learning or intrusive sensors, $TELESAFE$ uses only data already available in many households: the aggregated electricity consumption time series from **smart meters**. No additional sensors or instrumentation are needed, making the system lightweight, ethical, and broadly deployable. Beyond individual use, the system could be added to smart meter dashboards, helping customers track well-being while preserving data sovereignty.

Constraints and challenges. Designing a personal monitoring tool that is both useful and privacy-preserving requires avoiding traditional AI approaches that rely on centralized learning, cloud services, or personal data collection. In compliance with legal frameworks such as GDPR, our solution ensures that all processing, including detection, classification, and result aggregation, is performed locally and fully under the user's control. These constraints make it difficult to use standard IoT or machine learning systems, but they encourage the design of tools that are simple, respect privacy, and give full control to the user.

Objective. To foster a healthier teleworking environment, we propose a self-monitoring system to detect and quantify *boundary crossings* behaviors, particularly *breaks* during work hours which can indicate beneficial rest periods, and *Overworking* during private hours which may signal stress:

*(*BREAK SCENARIO*) Break observance during work periods*:

> The teleworker carries out personal activities, which normally occur during private time (e.g., watching TV). When these appear during scheduled work periods, the system interprets them as breaks.

*(*OVERWORKING SCENARIO*) Work Intrusion in Private Time*:

> The teleworker continues work activities beyond working hours (e.g., computer usage). The presence of these during private time is interpreted as overworking, potentially linked to elevated stress levels.

Our approach in a nutshell. To achieve these objective under the stated constraints, we recently proposed $TELESAFE$ [5], a mechanism to detect work-private boundary crossings by analyzing aggregated, non-annotated time series. Our algorithm detects similar activities based on an elastic time-shift aware distance metric (DTW) and unsupervised clustering. The system classifies activities as private or work-related based on their temporal distribution, without

[1]See financesonline.com for data on remote work, last accessed Sept 2025.

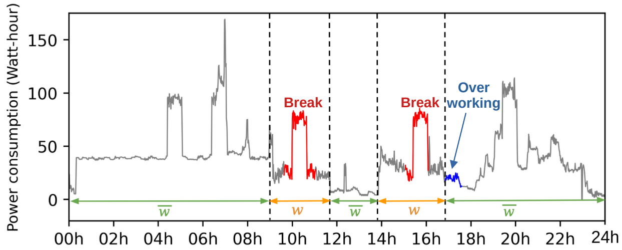


Fig. 1: Boundary crossings: break (red), overwork (blue).

needing to recognize specific devices or actions.

We demonstrate $TELESAFE$ on real and simulated electric consumption traces in a teleworking context, highlighting its effectiveness in identifying breaks and overworking episodes (Figure 1). An interactive GUI allows users to upload their own energy consumption data, visualize detected boundaries, and receive a weekly report with tailored advises, all while keeping their data private and under their control.

II. Related work

Existing approaches to activity monitoring often rely on intrusive sensing, large-scale data collection, or cloud-based processing, which raise significant privacy and deployment concerns. In contrast, our approach prioritizes user autonomy and privacy by leveraging non-intrusive, already-available data and processing it entirely on-device.

One way to monitor boundary crossings is using fine-grained device data with IoT – enabled device communication. Instrumenting all devices could track usage and, for computers, even purpose (e.g., via bossware). However, privacy regulations vary: some regions permit supervision (e.g., the US's Interguard), while others, like the EU, impose strict limits. Some works propose minimal instrumentation of the teleworker's space, such as using accelerometers and light sensors [6]. However, this could be rejected by workers due to the right to home inviolability, protected by constitutions and the EU Convention on Human Rights [7], which can only be bypassed in extreme cases.

To address the issue of intrusion, many studies focus on non-intrusive activity recognition, such as Non-Intrusive Load Monitoring (NILM), which analyzes aggregated time series (e.g., electricity consumption) to infer individual activities [8], [9]. Some approaches suggest using these tools for burnout detection [10], but offer no concrete solutions and rely on supervised learning, which requires large, annotated datasets from diverse teleworking environments.

III. $TELESAFE$ Overview

We now present the architecture of $TELESAFE$, a fully local, lightweight system for detecting work-life boundary crossings from household electricity consumption. The approach is unsupervised and relies only on smart meter data, in line with the privacy-by-design principle emphasized in the introduction. The processing is composed mainly by four steps: 1) Time series preprocessing, 2) boundary crossings detection, 3) Result display, and 4) Weekly teleworking report.

A. Data preparation and preprocessing

The input to $TELESAFE$ is a time series representing the household electricity consumption of teleworkers, obtained from sequential measurements over time [11], where each data point s_i reflects the aggregated energy usage at a specific time. Since no public dataset includes labeled telework-related activities such as breaks or overworking, we simulate these scenarios using a combination of real-world datasets. A simulator interface (demonstrated in our video) is used to generate synthetic profiles based on the following three sources:

- *Individual Household Electric Power Consumption (IHEPC)* from UCI [12]: This dataset provides detailed household power usage over nearly four years, sampled at one-minute intervals. We use it to simulate background consumption, serving as the ambient noise profile.
- *Orange4Home* [13]: This dataset captures teleworking activity within a smart home environment. We extract energy traces from office plug usage to model work-related consumption during typical work hours.
- *Tracebase* [14]: This dataset contains energy usage profiles for a range of household devices (e.g., dishwasher, dryer). We select devices across multiple categories and durations to simulate diverse personal activities.

To emulate boundary crossings, we inject Tracebase device traces into a time series aggregating a random Orange4Home telework profile and IHEPC daily consumption data.

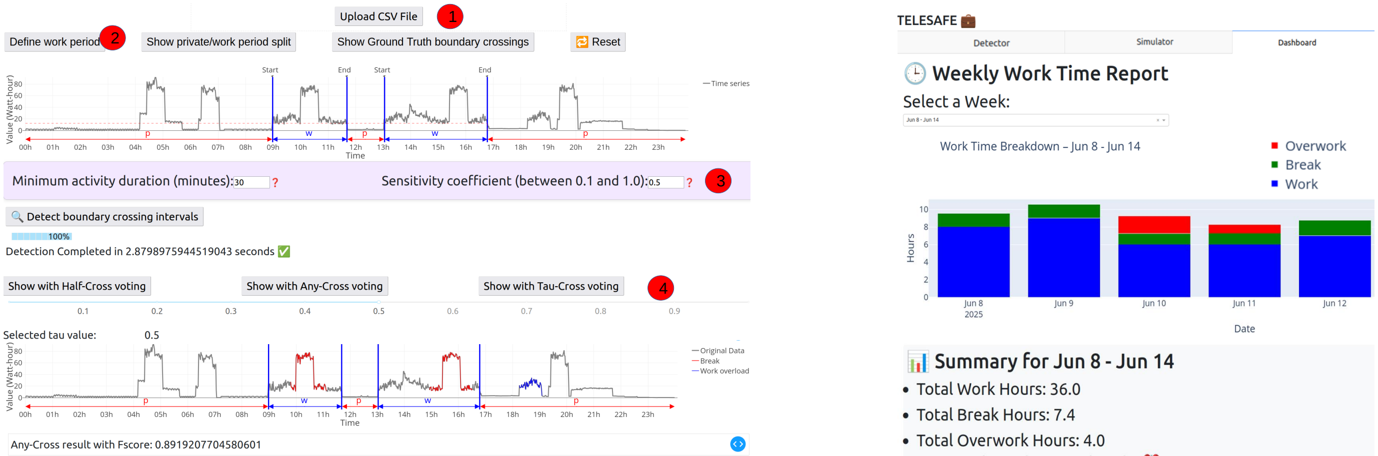
Assumption. Following our recent work [5], the classification of subsequences of time series is grounded on the assumption that personal activities (resp., work activities) occur more frequently during private hours (resp., work hours), enabling classification without knowing specific activities or devices.

Data preprocessing. The system accepts as input a time series of length n uploaded in the form of a data frame by the user. The data frame should contain at least a column of time, and a column of energy consumption in Watt hour. Users determine their own working hours depending on their employment contract, and the private periods are automatically defined as the complementary segments of the day. Typically, work periods are split into two segments before and after a lunch break as illustrated in Fig. 1, where work periods are marked as w (orange) and private periods as $\overline{w}$ (green).

B. Boundary Crossings Detection

The main objective of the detection algorithm is to identify similar activities that appear across different periods of the day. To this end, we segment the time series into fixed-length subsequences of length ℓ, where ℓ represents the minimum duration for an activity to be considered meaningful in the analysis (see ③ in Fig. 2a). The final output is the boundary crossings subsequences occurring during work (in red in Fig. 1) and private (in blue) periods.

Distance metric and threshold computation. The detection phase requires measuring similarity between subsequences, which is based on a distance metric. We use the Dynamic Time Warping (DTW) distance as the metric [15], well-suited



(a) Detector interface.

(b) Weekly report interface.

Fig. 2: Interface TELESAFE

for capturing out-of-phase patterns. Furthermore, a threshold of similarity (ϵ) should be calculated dynamically and without supervision. The method described in [5] is implemented as $\epsilon = mean(mp) - std(mp)$, where mp is a vector (also known as "Matrix Profile" [16]) storing the minimum distance values for each subsequence in private periods when compared to all the subsequences in work periods. We extend this definition to $\epsilon = mean(dists) - \alpha * std(dists)$ by introducing a sensitivity coefficient α (see ③ in Fig. 2a). This parameter, which ranges between 0 and 1, allows the user to adjust slightly the threshold while being set to 0.5 by default: a smaller α results in a looser threshold, capturing more similar patterns, while a larger α tightens the threshold, resulting in fewer detected pattern.

PWM matrix. We define two subsequences as similar if the distance between them, computed using our distance measure, falls below the threshold. We then store the indices of similar subsequences for each subsequence in a data structure called "Private/Work Matrix" (PWM). Specifically, if the distance between subsequences S_i and S_j is below the threshold ϵ, then $PWM[i,j] = PWM[j,i] = 1$; otherwise, $PWM[i,j] = PWM[j,i] = 0$. Computing the PWM for all the subsequences in a time series of length n has a worst case complexity of $O(n^2\ell^2)$. We provide an optimization below.

Overall $TELESAFE_{DBSCAN}$ algorithm. It follows a structured process [5]. First, we compute and store the distance values for each pair of subsequences in the teleworker's profile. Then, we determine the threshold ϵ as outlined earlier. Finally, we construct the PWM using ϵ and the stored distance values. A DBSCAN-based clustering algorithm is then applied to the PWM to group similar subsequences. Each cluster is analyzed by counting the number of indices within the work period w and the private period $\overline{w}$. If the majority of indices belong to w, the cluster is labeled as $WORK$; otherwise, it is labeled as $PRIVATE$. In case of ties, the label defaults to $PRIVATE$ to prioritize privacy. The subsequences appearing in $\overline{w}$ but labeled as $WORK$ indicate a overworking scenario, while those appearing in w but labeled as $PRIVATE$ indicate a break scenario. This algorithm can be parallelized to benefit from speed-ups when utilizing multicore computing resources.

Optimization. To optimize performance, we proposed a lightweight alternative, $TELESAFE\text{-}Lite$, and compare it with $TELESAFE{-}DBSCAN$ in [5]. The optimization consists of an improved PWM computation and clustering process, leading to only a slight loss in accuracy but a significant gain in performance.We first compute and store only the distances between subsequences from work and private periods, enabling threshold determination and partial PWM precomputation. We retain only the indices i where there exists a j such that $PWM[i,j] = 1$, treating these as candidate subsequences while eliminating those for which $PWM[i,j] = 0$ for all j, as they have no direct neighbor in the other period. This approach not only reduces the number of distance computations but also mitigates the risk of forming excessively large clusters in density-based clustering (e.g., DBSCAN), which can occur if the density threshold (ϵ) is too high. Finally, we complete the PWM between subsequences within the same category (work or private) only for the remaining candidates. Finally, we form clusters with only the direct neighbors of a subsequence in order to determine if it is of type $PRIVATE$ or $WORK$. If a candidate subsequence falls in working period w and the majority of its neighbors fall in $\overline{w}$, it is classified as a boundary-crossed break subsequence; conversely, if it falls in $\overline{w}$ and has more neighbors in w, it is classified as a boundary-crossed overworking subsequence.

Detecting overwork during the day-off. An overlooked but interesting scenario is the detection of overwork during days off, such as weekends or bank holidays, where no predefined work boundaries exist on these days: all time is considered personal by default. The presence of work activity during these periods directly indicates overwork. To identify such instances, unsupervised similarity detection methods can be effectively applied, for example using a matrix profile [16], [17] based on teleworker's typical work behavior without break. Importantly, this phase requires no supervision, and all necessary data remains available locally on the user's device.

C. Result Display

The detection algorithm outputs subsequences identified as boundary crossings. However, some of these may be false positives or may only partially overlap with an actual boundary crossing. This can lead to fragmented or noisy results. To reduce noise in the result visualisation, we propose voting mechanisms (see ④ in Fig. 2a). These mechanisms exploit the fact that each point in the profile is covered by ℓ overlapping subsequences, each being classified as 'boundary crossed' or 'not' by the detection algorithm.

Half-Cross vote. Our default voting method, Half-Cross vote, displays a point if the majority of the subsequences covering it are classified as boundary crossings.

Any-Cross vote. The Any-Cross vote maximizes coverage by displaying all points that are part of a boundary crossing subsequence. This method provides an upper bound for accurate detection, but also introduces noise.

Tau-Cross vote. The Tau-Cross vote provides greater flexibility by allowing users to set a threshold parameter τ. A point is displayed if at least a fraction τ of the subsequences covering it are classified as boundary crossed. Specifically, $\tau = 0.5$ corresponds to Half-Cross vote, while $\tau = 1/m$ corresponds to Any-Cross vote. It is important to note that this parameter is different from α (in section III-B), which influences the detection result. Unlike α, this parameter does not change the result of boundary crossing subsequence detection – it only affects how the results are visualized. Its purpose is to help users review the results based on their own knowledge.

D. Weekly teleworking report

After each detection, the break and overwork hours can be easily calculated by multiplying the length of the detected subsequence by the sampling frequency. The working hours are then determined by subtracting the break duration from the user-defined total working period. This aggregated data is stored locally on the teleworker's device. We propose an interface that visualizes the report by aggregating these key metrics (i.e., work, break and overwork hours), as illustrated in Fig. 2. A summary of this data is displayed below the figure, along with a personalized advice message designed to promote a healthier teleworking experience. The advice may include suggestions such as taking more breaks if insufficient rest periods were detected during the workday, or reducing work outside of designated working hours. In future versions, the $TELESAFE$ report interface could be adapted to energy suppliers' dashboards, offering value-added services to customers without compromising privacy.

IV. Demonstration Scenario

The scenario consists of three interactive parts. In the first part, the user can generate a synthetic time series profile – with overworking or break activities – using our simulator (interface shown in the video). The second part (shown in Fig. 2a) involves detecting boundary crossings from either simulated profile or real profile obtained from the smart meter uploaded by clicking the button ①. The time series is then displayed in the first graph. Next, the user defines the work periods by clicking the button ② and selecting the start and end points directly on the graph. They can visualize the segmentation by clicking "Show Private/Work Split". It is possible for the user to adjust the subsequence length and threshold parameter α that we discuss in Section III-B, by entering values in the text box ③, overriding the default settings if needed. Once configured, the detection process can be initiated by clicking the "Detect boundary crossings intervals" button, which runs our algorithm described in Section III-B. A message displaying the execution time confirms completion. After detection, the user can select a visualization strategy (described in Section III-C) in ④ to observe the detected boundary crossings intervals along with their assessment results in the second graph. The third part involves visualizing the teleworking report and summary for a selected week, along with personalized advice.

Acknowledgements. Supported by grants ANR-22-PECY-0002, ANR-23-CMAS-0019 funded by ANR, and PID2023-150310OB-I00 funded by MCIU, AEI and FEDER.

References

[1] C. E. Hall, S. K. Brooks, F. Mills, N. Greenberg, and D. Weston, "Experiences of working from home: umbrella review," *Journal of Occupational Health*, vol. 66, no. 1, 2024.

[2] E. M. Hunter, M. A. Clark, and D. S. Carlson, "Violating work-family boundaries: Reactions to interruptions at work and home," *Journal of Management*, vol. 45, no. 3, pp. 1284–1308, 2019.

[3] C. Weinert, C. Maier, and S. Laumer, "Why are teleworkers stressed? an empirical analysis of the causes of telework-enabled stress," 2015.

[4] U. S. D. of Labor, *Occupational Safety and Health Act of 1970 (OSH Act)*, 1970.

[5] H. Zhang, M. Brahem, N. Anciaux, B. Nguyen, and J. M. de Fuentes, "Telesafe: Detecting private/work boundary crossings in energy consumption trails in telework," *PVLDB*, vol. 18, no. 6, 2025.

[6] J. Wang, M. Aguirre Mehlhorn, K. Ditler, B. Effner, and T. M. Deserno, "Tracking activities of daily living in the home office," *Informatics and Technology in Clinical Care and Public Health*, vol. 289, p. 162, 2022.

[7] C. of Europe. (1950) European convention on human rights.

[8] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[9] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, "Hive-cote 2.0: a new meta ensemble for time series classification," *Machine Learning*, vol. 110, no. 11, pp. 3211–3243, 2021.

[10] Insight7, "8 real-time analytics tools that detect agent burnout," 2024.

[11] P. Esling and C. Agon, "Time-series data mining," *ACM Comput. Surv.*, vol. 45, no. 1, dec 2012.

[12] G. Hebrail and A. Berard. (2012) Individual Household Electric Power Consumption. UCI Machine Learning Repository.

[13] J. Cumin, G. Lefebvre, F. Ramparany, and J. L. Crowley, "A dataset of routine daily activities in an instrumented home," in *UCAmI 2017*. Springer, 2017, pp. 413–425.

[14] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz, "On the accuracy of appliance identification based on distributed load metering data," in *2012 Sustainable Internet and ICT for Sustainability (SustainIT)*. IEEE, 2012, pp. 1–9.

[15] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[16] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, "Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets," in *ICDM*, 2016, pp. 1317–1322.

[17] J. Paparrizos, Y. Kang, P. Boniol, R. S. Tsay, T. Palpanas, and M. J. Franklin, "Tsb-uad: an end-to-end benchmark suite for univariate time-series anomaly detection," *PVLDB*, vol. 15, no. 8, pp. 1697–1711, 2022.