



# TELESAFE: Detecting Private/Work Boundary Crossings in Energy Consumption Trails in Telework

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

Mariem Brahem  
Inria, INSA CVL, Univ. Orléans  
mariem.brahem@inria.fr

Nicolas Ancaux  
Inria, INSA CVL, Univ. Paris Saclay  
nicolas.ancaux@inria.fr

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

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

## ABSTRACT

Teleworking has become a social gain following the COVID-19 lockdowns. In many professions, remote work is becoming a common practice, either at the employee's home or in a shared space nearby. However, this creates an implicit private/work-life tension as private activities may be carried out during work time and vice versa. Detecting *boundary crossings* is of utmost relevance – they serve as evidence of the workers' breaks and right to rest. However, this must be achieved without excessive surveillance. Existing activity recognition techniques either do not address the border crossing problem or require a priori training.

To address this issue, this article proposes *TELESAFE*, a boundary crossing detector solution for teleworking. *TELESAFE* does not require any training nor instrumentation of the teleworker home and can be run locally in resource-constrained devices. To illustrate its suitability, it is applied on electric consumption trails so as to enable self and third-party assessment (e.g., work inspectors) on working conditions. Results on real-world datasets show a  $F_{score}$  over 90% for identifying private activities involving one or more devices with usage patterns of varying lengths. Interestingly, *TELESAFE* outperforms Machine and Deep-Learning approaches in the most complex settings, without the burden of training.

### PVLDB Reference Format:

Haoying Zhang, Mariem Brahem, Nicolas Ancaux, Benjamin Nguyen, and Jose Maria de Fuentes. *TELESAFE: Detecting Private/Work Boundary Crossings in Energy Consumption Trails in Telework*. PVLDB, 18(6): 1565–1578, 2025.  
doi:10.14778/3725688.3725690

### PVLDB Artifact Availability:

The source code, data, and/or other artifacts have been made available at <https://gitlab.inria.fr/haoying.zhang/telesafe-privateworkdetection>.

## 1 INTRODUCTION

Teleworking has emerged as a new norm in many societies, especially after COVID-19. According to FlexJobs [25], it is a well accepted work modality by employees as 65% would like to work

fully remote. It contributes to productivity, as teleworkers are 35% more productive, and 73% declare working outside their work hours. **Motivation and goal.** During work periods, workers (and, of course, teleworkers) are supposed to take breaks. This leads to a *boundary crossing* effect – private activities are carried out during working hours [31]. For instance, personal activities like using household appliances such as hair dryers or coffee machines, or even engaging in leisure activities like playing video games or watching TV, can overlap with work tasks during working hours. The observance of these breaks is of utmost relevance to prevent burnout and ensure basic conditions for the worker. Indeed, employee well-being is a basic premise in most modern countries (e.g., EU Directive [18]). However, 86% of teleworkers report experiencing burnout, according to ThinkRemote [54].

These boundary crossing effects may also happen the other way around – work activities being carried out during non-working periods. This raises issues like work overinvolvement, and it poses a conflict against the right to disconnection, which is regarded as a basic principle for mental health. Remarkably, workers report a number of health problems such as anxiety or stress (e.g., see [2]).

Thus detecting these boundary crossings is relevant so as to assess the quality of working periods (in which interruptions should exist to a certain extent) and private life (in which boundary crossings should be minimized). To address this matter, many countries have implemented the role of work inspector as a public authority in charge of supervising work activities. This is the case, for example, of France [21], Spain [20], Germany [23] and the US [45].

**Limits of existing works.** Detecting boundary crossings using fine-grained device data would be straightforward. In the last years, the Internet of Things (IoT) enables a direct communication with devices [35]. By instrumenting all devices, it would be possible to ascertain when they are used and, in the case of computer devices, even the purpose (e.g., by installing bosswares). However, this may pose some conflicts with privacy regulations. While some regions allow this kind of supervision (e.g., US's Interguard [32]), others have ruled against it. In fact, the European Union has already imposed strict limits in the context of telework [17]. This is due to the potential threat posed by time series analysis. For example, energy data could inadvertently reveal information about teleworkers' appliance use and environment [15], revealing their daily routines and behaviors [52]. Indeed, in the post-COVID context, this leads to power imbalances between employers and employees [56].

Other works have proposed a reduced instrumentation of the teleworker space. For example, [58] includes three low-cost sensors

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing [info@vldb.org](mailto:info@vldb.org). Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.

Proceedings of the VLDB Endowment, Vol. 18, No. 6 ISSN 2150-8097.  
doi:10.14778/3725688.3725690

(namely accelerometers and light sensors). However, such a transformation of the teleworker environment could be rejected by the worker based on the notion of inviolability of the home, which is present in Constitutions of dozens of countries. Remarkably, the EU Convention on Human Rights states the right of protecting one’s home and the avoidance of interferences from public authorities in this right [44]. This can only be circumvented under extreme circumstances and only when no other means can be implemented, which is not the case of the context described so far.

To address this issue, many contributions have been devoted to non-intrusive activity recognition. Among them, Non-Intrusive Load Monitoring (NILM) approaches rely on the analysis of aggregated time series (e.g., electric consumption trails of all devices together) to distinguish individual activities [8, 11, 34, 42]. However, detecting boundary crossings requires considering the notion of work and leisure periods. Thus, the same activity (e.g., using the coffee maker) may be regarded as a boundary crossing (during work hours) or as a normal task (during leisure time). Moreover, leading approaches generally depend on supervised machine learning systems [42], notably Deep Learning algorithms [34, 48]. These require extensive annotated data from various teleworkers’ household environments, as daily profiles vary with individual practices and behaviors. Previous works have already shown that performance (and thus device usage) varies significantly among teleworking days [4]. Nevertheless, no studies have yet demonstrated the effectiveness of such solutions in the so-variable teleworking context.

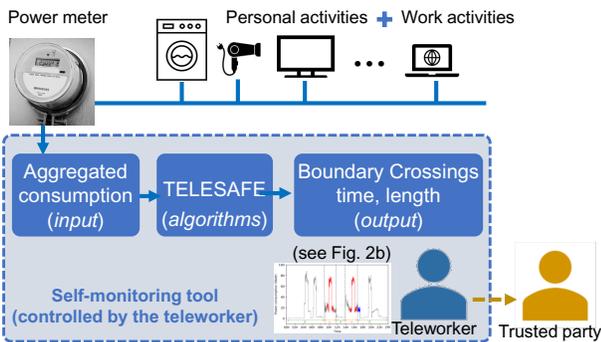


Figure 1: Proposal overview

**Approach.** To overcome these limitations, this work proposes *TELESAFE*, which stands for **TELE**work **S**afety and **A**ctivity **F**eedback **E**valuation, a mechanism to detect work-private boundary crossings by analyzing aggregated, non-annotated time series.

To illustrate its suitability, *TELESAFE* is applied on electric consumption trails in a teleworker home (see Figure 1), an easily accessible data source for all teleworkers via the smart meter, which has penetrated 80% of US households and 14 EU countries [3]. Using other types of time series data, such as network consumption and accelerometer readings, is straightforwardly feasible with the proposal; however, these approaches need the installation of specific software or hardware.

Without loss of generality, *TELESAFE* is devoted to detecting private activities during work periods, or vice versa. These detected events are then used by the employee for self-monitoring.

At their own will, the teleworker *may* share this information with a trusted party (e.g., work inspector). Therefore, *TELESAFE* data remains solely under the control of the teleworker. We also compare *TELESAFE* to supervised AI-based techniques, showing its superiority under some settings. This paper represents the first attempt to formalize the problem and reflect real-world teleworking scenarios, paving the way for future research.

**Technical contributions.** Specifically, this paper makes the following contributions:

- We formalize the issue of detecting boundary crossings of work/private activities using a time-series of numeric data related to the user;
- We propose *TELESAFE*, an unsupervised pattern-aware approach to identify work/private patterns in such time-series;
- We evaluate our proposal using real-world datasets to show the effectiveness and scalability of *TELESAFE*, including its operation on devices with limited computing resources;
- We release our experimental framework code to foster further research.

**Scope and Limitations.** This work focuses on identifying work-private boundary crossings from aggregated, non-intrusive time series data. While our approach is general, we focus on an application using energy consumption patterns collected via smart meters in the context of telework. Unlike traditional methods, our approach avoids intrusive monitoring, does not rely on labeled datasets, and minimizes infrastructure requirements, making it suitable for real-world teleworking scenarios. Nevertheless, our method is inherently limited to detecting patterns that are observable in the available data and may not capture activities unrelated to measurable signals (e.g., non-electronic tasks), or distinguish work and private patterns with very similar traces.

**Relevance to the database community.** The work presented in this article allows extracting new knowledge (work/private activities) by analyzing a time series using data mining techniques related to Matrix Profile (MP) through an unsupervised approach, without the need for manual tagging. We evaluate both the quality of our algorithms and their experimental efficiency and scalability via a real world inspired use case of teleworkers’ electric consumption time series. Our results show that this approach provides a practical solution to this problem.

## 2 PROBLEM ANALYSIS

The teleworking environment stands apart from both the household and the conventional workplace, necessitating a specialized approach to problem analysis. We first introduce the use case in Section 2.1. The required definitions are introduced in Section 2.2. Finally, the objective and the underlying constraints are stated in Section 2.3. Table 1 summarizes the main notations.

### 2.1 Use Case: Telework Self-Inspection

In most countries, a legal framework establishes the boundaries of the employer-employee relationship, outlining the rights and obligations of both parties. Alongside these regulations, governments appoint authorities, often referred to as work inspectors, tasked with ensuring compliance with these principles. Work inspectors

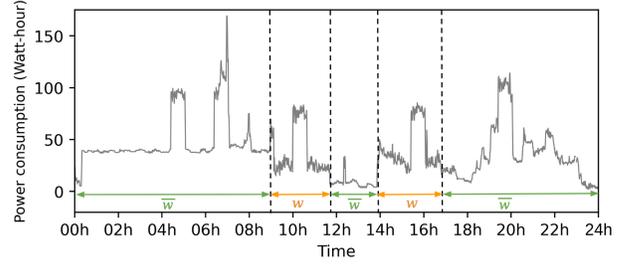
**Table 1: Notations**

Symbol	Definition
$S_u$	Teleworker profile: time series of a user $u$
$(t, s)$	(timestamp, value) pair of a time series
$w$	Work period
$\bar{w}$	Private period
$n$	Length of the time series
$\ell$	Length of a subsequence
$\epsilon$	Distance threshold
$\Delta(S, Q)$	Distance function between time series $S$ and $Q$
$DP(S_i, Q)$	Distance Profile between time series $S_i$ and $Q$
$PWM(S, Q)$	Private/Work Matrix of $\epsilon$ -close subsequences between $S$ and $Q$

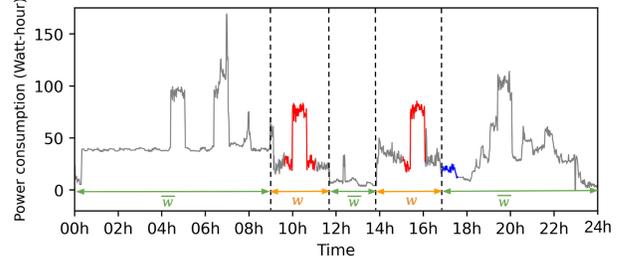
typically ensure that employees take a sufficient amount of breaks as defined by the law. For example, lunch breaks are typically designated as private time because employers are required to provide a 30-minute meal period to nonexempt employees, freeing them from all duties, as mandated by laws like California’s labor code [1]. In the same vein, work inspectors also verify that no work activities are being carried out systematically beyond working hours. In a traditional workplace, work inspectors may conduct inspections on the premises of the company. However, in a telework context, this is legally challenging.

To address this, a self-monitoring system can be provided to employees, enabling them to monitor their own compliance with regulatory breaks and respect for the right to disconnect after work. This system would allow employees to log their work and break times, receiving feedback and reminders to ensure they adhere to legal requirements. Moreover, these detected events could also be voluntarily shared by the teleworker with third parties, such as work inspectors or doctors monitoring employees’ health.

**Ethical and legal considerations.** *TELESAFE* detects boundary crossings without invasive methods like home instrumentation, reducing concerns about privacy intrusion in private spaces. However, using energy consumption data to infer private activities raises ethical and legal issues if accessed or misused without explicit consent. As a self-monitoring tool, *TELESAFE* empowers teleworkers by giving them control over their data, allowing them to evaluate their work-life balance. From a legal perspective, it operates within the data portability right, under privacy regulations like GDPR Article 20. This enables individuals to use their energy consumption data from smart meters (as a typical use case for Personal Data Systems [6, 7, 22]), to evaluate breaks, their right to disconnect, and potential risks of overwork or burnout, promoting a healthier work-life balance. In telework inspections, labor inspectors traditionally ensure compliance with labor laws at company premises, but teleworking complicates this due to home privacy protections. *TELESAFE* addresses this challenge by enabling employees to self-monitor and voluntarily share their data with, e.g., labor inspectors or health professionals, ensuring regulatory compliance while respecting autonomy. Ethical and legal concerns arise if data is accessed without explicit consent, risking privacy violations and surveillance. Privacy-preserving methods like differential privacy



(a) Teleworker profile  $S_u$ .



(b) Boundary crossings during working hours (in red) and non-working hours (in blue).

**Figure 2: Boundary crossings on a teleworker profile.**

for time series [41], should be explored to anonymize data, ensuring ethical and legal compliance across various applications.

## 2.2 Definitions

In order to address the proposed use case, a number of definitions are required. Firstly, we define work and private periods in the teleworking context in Def. 1. Depending on the context, these periods may be determined dynamically by when an employee logs into their employer’s portal, or they may be fixed based on the employment contract or an employee’s statement.

*Definition 1 (Work (resp. Private) Periods).* The work (resp. private) periods  $w$  (resp.  $\bar{w}$ ) of a teleworker, are defined as the time intervals within the teleworker’s schedule that are designated for work activities (resp. personal activities).

(EXAMPLE) *Work and private periods:*

For a typical working day, we may have a work period  $w = [09h; 12h] \cup [14h; 17h]$  and a private period  $\bar{w} = [00h; 9h] \cup [12h; 14h] \cup [17h; 24h]$ , as shown in Figure 2a.

We consider the time series (i.e., collection of values obtained from sequential measurements over time [26]) of the electricity consumption curve for that given day of teleworker  $u$ . Each data point  $s_i$  represents the aggregated energy consumption value of all active devices at that moment. We call *teleworker profile*, the corresponding time series noted  $S_u = \bigcup_{i \in \{1, n\}} (t_i, s_i)$ .

(EXAMPLE) *Teleworker profile:*

Figure 2a shows the example of the power consumption on a given day of teleworker  $u$ .

The specificity of teleworking from home lies in the overlap between professional and personal activities. This overlap typically constitutes what we term a private-work boundary crossing problem, which can be defined as follows:

*Definition 2 (Boundary Crossing Problem).*  $S_u$  exhibits a boundary crossing problem if there exists a subsequence  $S$  during a work period  $w$  that is due to personal activity; or if there exists a subsequence  $S$  during a private period  $\bar{w}$  that is due to work activity.

Our goal in this paper is to detect and resolve the boundary crossings problems in the teleworker profile.

(EXAMPLE) Resolution of the boundary crossings problems:

As shown in Figure 2b, a subsequence due to private activity occurs during the designated work period  $w$  at around 10h and another around 16h, e.g., when the teleworker turns the television on (as part of a personal activity) before finishing work.

### 2.3 Problem, Assumptions, and Constraints

**Problem formulation.** Given a teleworker profile consisting of a time series  $S_u$  of measures corresponding to work and private activities, and a work period  $w$  (the other periods are private periods annotated as  $\bar{w}$ ), our goal is to resolve boundary crossings problems, detect activities and classify them as *WORK* or *PRIVATE*: (1) any time window  $w' \subset w$  where private activities occur and influence work periods patterns; and (2) any time window  $w' \subset \bar{w}$  where work activities occur and influence values in private periods patterns. Note that we are not interested in recognizing the type of activity associated with these patterns, which is different from the state-of-the-art NILM approaches (see Section 3).

The ultimate goal is to ensure that telework complies with the legal work framework. This implies that working hours count on a minimal amount of breaks, as well as leisure time does not include work-related activities.

**Assumptions.** The identification and classification of subsequences of time series is grounded on the assumption that personal activities (resp., work activities) tend to occur more frequently during private hours (resp., work hours). This provides a general solution without assuming activity specifics or device details.

**Constraints.** Considering the use case, our approach must adhere to several constraints:

- *Local:* Processing must be conducted locally, as work inspectors cannot access a teleworker's home.
- *Non-intrusive:* It must use data produced from electric consumption, without additional device-specific insights.
- *Unsupervised:* It should not rely on labelled data from employees for identifying boundary crossings.
- *Lightweight:* It must operate on a smart meter or user device with limited computational capabilities.

Addressing the boundary crossings problem under these constraints is challenging. It requires working in isolation on a resource-limited device, using unannotated time series data without prior knowledge. As shown in the next section, these constraints invalidate classic AI-based supervised techniques.

## 3 RELATED WORK

There are two areas related to the focus of this work – the boundary crossing problem in telework and activity recognition for time series. Table 2 summarizes the comparison. As one can see, our approach is the first to cover the telework context and consider a solution for the boundary violation problem. By construction, it does not require prior training, representing a significant advancement over existing research that is predominantly theoretical or limited in scope regarding telework-specific challenges.

### 3.1 Boundary crossings in Telework

Although telework has increasingly captured the attention of employers and employees alike, the study of boundary crossing within this context remains plagued by a lack of conceptual clarity. As a result, there is growing evidence to suggest that there is a dark side to telework, where the lines between work and personal life become blurred. However, these findings remain scattered and theoretical, missing practical implementations and solutions to detect such crossings.

Much of the existing research uses the term “boundary violation problem” and emphasize the need of new research to manage such violations. However, we think that the term “violation” implies a negative connotation suggesting breaches of personal and professional limits, thus, we consider the term “crossing” to be more appropriate for capturing the boundary dynamics in telework. Hunter *et al.* [31] discuss the conceptual foundations of boundary violations, pointing out how telework can blur traditional work-life boundaries. Sivunen *et al.* [53] studied the links between communication technologies and boundary violations, illustrating how these technologies complicate the management of work-home boundaries thought constant connectivity. Their study highlights the dual-edged nature of such technologies, which enhance flexibility and communication, but also invade personal privacy, due to the difficulty to separate private and work activities.

Büchler *et al.* [13] have found that such boundary violations negatively impact employee's well-being, leading to greater experiences of stress and burnout. This is confirmed by Lambert *et al.* [36], who links these violations directly to mental health challenges such as increased anxiety and decreased job satisfaction. Nevertheless, note that these “violations” generally perceived as negative, can also be considered as positive – opportunities for teleworkers to take breaks during working hours.

While prior research provides valuable theoretical insights into boundary management, there is a significant gap in the development of practical solutions. Our work seeks to bridge this gap by introducing a novel approach that not only recognizes the issues, but also propose a novel model for detecting boundary crossings, moving beyond theory to practical application.

### 3.2 Time Series for Activity Recognition

The use of time series for activity recognition has been largely explored, particularly with the proliferation of human-related sensors. This section presents key contributions in this area, focusing on signal processing, AI-based and database-oriented approaches. Below, we position our research relative to these existing methods to highlight the unique contributions of our approach.

**Table 2: Summary of Related Work**

Ref.	Year	Approach	Telework-based	Training-free	Focus Area	Boundary crossing	Limitations
[48]	2020	Deep Learning with sensor data	✗	✗	User activities	✗	Limited to wearable sensors
[33]	2020	Electricity consumption data analysis	✗	✓	User activities	✗	Sophisticated computation, intrusive data analysis
[28]	2023	Personalized Deep Learning	✗	✗	Sensor data	✗	Requires sophisticated AI, Supervised approach
[57]	2022	Privacy preservation for electricity time series	✗	✗	User data	✗	Supervised approach
[31]	2019	Conceptual analysis	✓	✗	Work-life boundaries	✓	No practical application
[13]	2020	Empirical research	✓	✗	Employee well-being	✓	No practical application
[53]	2023	Conceptual analysis	✓	✗	Communication technologies and telework	✓	No practical application
<b>Ours</b>	2024	Boundary crossing model	✓	✓	Time Series in Telework	✓	See Section 6.3

3.2.1 *Signal processing approaches and NILM.* Research efforts have explored the use of wearable sensor data for activity recognition, as demonstrated by Qin *et al.* [48]. Furthermore, the use of electricity consumption data for activity recognition has also been explored by Ishizu *et al.* [33]. However, from the aggregated consumption data, Non-Intrusive Load Monitoring (NILM) techniques have been proposed to identify individual appliances, also referred to as energy disaggregation [9]. This leads to a (potentially undesired) activity recognition [59]. Similarly, even the presence or absence of the user can be inferred by using appliance electrical consumption, which is dangerous as it may enable robberies [37].

Despite the potential of NILM, it fails to meet our non-intrusive and local processing constraints, as it requires sophisticated computations and potentially invasive data analysis.

3.2.2 *AI-Based solutions.* While NILM techniques are advancing, a major challenge remains: they typically require training in Artificial Intelligence-based systems to perform activity recognition. As such, Ferrari, Li *et al.* [28, 38] propose to opt for Deep Learning approaches, but do not align with our need for unsupervised and computationally light approaches. These approaches typically depend on extensive, annotated datasets and considerable computational resources, making them incompatible with our constraints.

3.2.3 *Database-oriented solutions (Matrix Profile).* Matrix Profile techniques [39, 40, 43, 60–62] and techniques based on motif similarity search [51] offer an efficient way to analyze patterns in time series data with minimal computational resources. However, we demonstrate limitations in accurately identifying boundary crossings under our specified constraints (see Section 6). Existing approaches based on Matrix Profile lack the ability to distinguish between types of activities, such as differentiating between work-related and private activities within a telework context.

In summary, our work presents a new approach to boundary crossing detection by adapting a matrix profile approach to the constraints of the telework context. Unlike previous research, which mainly provides conceptual views, our work proposes a practical solution. In the following, we explore our methodology further by evaluating its effectiveness to identify boundary crossings and comparing its performance against NILM and AI-based solutions.

## 4 FOUNDATIONS OF TELESAFE

Resolving the boundary crossings problem involves detecting and classifying subsequences that occur during inappropriate periods

(e.g., private subsequences during work periods and vice versa). In this section, we introduce methods to detect similar subsequences (Section 4.1) and then classify them as either *WORK* or *PRIVATE* (Section 4.2), forming the foundations of *TELESAFE*. For clarity, we use Figure 3 as a running example to support our explanations.

### 4.1 Detection of Boundary Crossings

The identification phase requires measuring similarity between subsequences. We therefore define a distance function  $\Delta$ :

*Definition 3 (Distance Function).* We denote  $\Delta(S, Q, \Pi)$  as a distance function with codomain  $\mathbb{R}^+$ , which computes the distance between two time series  $S$  and  $Q$ , where  $\Pi$  represents a set of additional parameters specific to the distance function.

*Notation 4 ( $\epsilon$ -closeness).* Let  $\epsilon \in \mathbb{R}^+$ . If  $\Delta(S, Q) \leq \epsilon$  we say that  $S$  and  $Q$  are  $\epsilon$ -close.

(*RUNNING EXAMPLE*) *Distance:*

We use the Dynamic Time Warping (DTW) distance [5, 24, 49]. The DTW distance is defined between two subsequences  $S$  and  $Q$  of length  $\ell$  (i.e.,  $\Pi = \{\ell\}$ ). This distance measure also allows for the detection of patterns longer than  $\ell$ , thanks to its ability to recognize out-of-phase patterns.

Next, we introduce the concept of Distance Profile:

*Definition 5 (Distance Profile).* A Distance Profile  $DP(S_i, S)$ , is a vector of  $|S| - \ell + 1$  distances, between a given subsequence  $S_i$  of length  $\ell$  from index  $i$  to  $i - \ell + 1$ , and all subsequences  $S_j$  of length  $\ell$  of  $S$  starting at index  $j \in [0, |S| - \ell]$ .

(*RUNNING EXAMPLE*) *Distance profile:*

Figure 3(a) shows a time series  $S$  and a subsequence  $S_i$  in red, between 15h and 15h30, starting at index  $i = 920$  (frequency is 1 minute). Figure 3(b) represents  $DP(S_i, S)$ , where each dimension of the vector corresponds to an index value of the time series  $S$ . Note that  $DP(S_i, S)$  reaches a minimal value of 0 at the starting index of  $S_i$ , since it is the distance between  $S_i$  and itself. To find the closest distinct subsequences, we ignore all elements of  $S_i$  (marked in gray in the middle figure). Four  $\epsilon$ -close subsequences are identified (red points in Figure 3(b)): the closest subsequence is at index  $j = 592$  with a value  $DTW(S_{920}, S_{592}) = 1.83$ , the second closest is at index 260 with a value  $DTW(S_{920}, S_{260}) = 1.84$ , etc. To ensure patterns are not counted multiple times, when the distance between subsequences  $S_i$  and  $S_j$  falls below

a threshold  $\epsilon$  (i.e.,  $DTW(S_i, S_j) \leq \epsilon$ ), the  $\ell$  points immediately preceding and following  $S_j$  are disregarded when searching for other nearest subsequences.

All these values will be stored in a so called ‘‘Private/Work Matrix’’ (denoted as  $PWM$ ) to select subsequences that are sufficiently similar to a given one. To achieve this, we introduce  $\epsilon$  as an upper bound on the distances between the similar subsequences. We define this matrix as follows:

*Definition 6 (Private/Work Matrix (PWM)).* A Private/Work Matrix  $PWM(S, Q)$  of two time series  $S$  and  $Q$  is a matrix where the rows are the indices of subsequences of  $S$  and the columns are the indices of subsequences of  $Q$ . Each cell  $PWM(S, Q)[i, j]$  contains 1 iff  $\Delta(S_i, Q_j) \leq \epsilon$  and contains 0 otherwise.

(RUNNING EXAMPLE)  $PWM$  (see Figure 3(c)):

Suppose we choose  $\epsilon$  as indicated by the red horizontal line. Figure 3(c) shows the matrix of all the subsequences of  $S$  that are  $\epsilon$ -close to each other. As shown in the figure, there are four non-zero values for index 920, indicating that these are the only subsequences at a distance less than  $\epsilon$  of  $S_{920}$ . All other indices are set to 0, indicating that they are beyond  $\epsilon$ . Note that this matrix is sparse.

Remarkably,  $PWM$  can be seen as a form of generalization of Matrix Profile [61]. However, the matrix profile only keeps the closest index (i.e., there are no multiple  $\epsilon$ -close values kept per subsequence), regardless of its distance (there is no threshold  $\epsilon$ ).

## 4.2 Classification of subsequences

As stated in our assumptions (see Section 2), we classify each subsequence as  $WORK$  or  $PRIVATE$  based on its similarity to other events that occur predominantly during either  $w$  or  $\bar{w}$  periods. Since we do not have any additional information beyond  $w$  and  $\bar{w}$ , we classify each subsequence (identified by the index of its first element  $i$ ) using an unsupervised approach.

We can therefore formulate our solution to the boundary crossing problem, introduced in Def. 2, using the concepts we have introduced in this section, as follows:

*Proposition 7 (Classification of private (or work) boundary crossings).* Given an index  $i$  of a time series  $S_u$ , we say that  $i$  is a private (resp., work) boundary crossing if (1)  $i$  is an index belonging to the work (resp., private) period and (2)  $S_i$  is classified as  $PRIVATE$  (resp.,  $WORK$ ), using our classification mechanism.

(RUNNING EXAMPLE) *Classification of each subsequence:*

Leveraging the  $PWM$ , the closest subsequences of a given one are obtained. To determine if a subsequence is a private or work subsequence, we use a density-based clustering algorithm, DBSCAN [27], adapted to binary classification. DBSCAN is capable of discovering clusters of similar elements in a metric space (in our case, subsequences). We aim to classify all similar subsequences with the same label ( $WORK$  or  $PRIVATE$ ), considering that a cluster can grow if it has at least one neighbor. The point at time index 920 has four neighbors, three in  $\bar{w}$  (time values in green in Figure 3(c)) and one in  $w$ . Its neighbors have no other

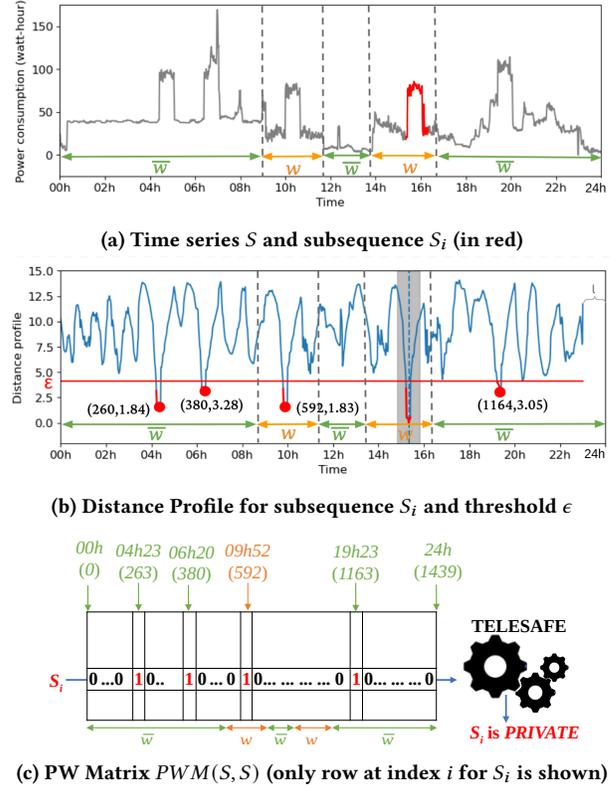


Figure 3: Running Example

neighbors than the ones in the cluster. Thus, this event is considered as  $PRIVATE$  as the majority of elements in the cluster belong to the  $\bar{w}$  period (see the assumption in Section 2.3).

In the next section, we present the algorithms for the notions introduced herein, leading to the proposed technique for detecting boundary crossings, referred to as  $TELESAFE$ .

## 5 TELESAFE ALGORITHMS

In this section, we introduce algorithms tailored for detection and classification of boundary crossings, as analyzed in the previous section. Following the same structure, Section 5.1 shows the computation of  $PWM$ , and the  $TELESAFE$  approach for boundary crossing classification is presented in Section 5.2.

### 5.1 Computing $PWM$

The objective of Algorithms 1 and 2 is to produce a data structure called the Private/Work Matrix ( $PWM$ ) of a teleworker profile  $S_u$ . In our use case,  $S_u$  is an electric consumption trail of a user  $u$  during a working day. The parameters of the algorithms, denoted  $S$  and  $Q$  for generality (and optimizations detailed in the next section), must be set to  $S = S_u$  and  $Q = S_u$ . The subsequence length  $\ell$  is determined by the user, given the minimum time duration desired for a personal/work activity to be considered in the boundary-crossings detection processes. It is important to note that the activities longer than  $\ell$  can also be detected.

**Building PWM (Algorithm 1).** The first phase of the algorithm computes the nearest neighbors (NN1) for the distance profile between each subsequence  $S_i$  of length  $\ell$  in  $S$  and the time series  $Q$  (line 5 in Algorithm 1). This operation is a fundamental task in time series analysis (e.g., all-pairs of similarity search [12] or matrix profile [61]), and numerous implementations are available. The *dists* array thus stores the minimum distance values for each subsequence  $S_i$  in  $S$  when compared to  $Q$ . Next, the threshold  $\epsilon$  is initialized based on *dists*. It is defined<sup>1</sup> as one standard deviation below the mean of *dists*, as  $\epsilon = \text{mean}(\text{dists}) - \text{std}(\text{dists})$ . Note that while determining a threshold in a general context can be challenging, this approach is straightforward, and it helps to filter out patterns that deviate significantly, ensuring that only relevant boundary-crossing patterns are considered in our specific context. The algorithm then iteratively computes all the closest neighbors within the  $\epsilon$  distance for each subsequence in  $S$  or for those specified in *PatternIndexList* (lines 7 to 12 in Algorithm 1). This operation can be executed in parallel by distributing tasks across multiple processors. Finally, the algorithm returns the matrix *PWM*, encapsulating the relationship between the subsequences of  $S$  and  $Q$  based on the defined threshold  $\epsilon$ .

The internally called function *ProcessIndex* computes each column of the *PWM*, as described in Algorithm 2.

---

#### Algorithm 1: Building PWM

---

**Input:** Time series  $S$  and  $Q$ , subsequence length  $\ell$ , *PatternIndexList* (Optional),  $\epsilon$  (Optional)  
**Output:** PW Matrix *PWM* of the time series  $S$  and  $Q$

```

1  $n \leftarrow |S|, m \leftarrow |Q|$  // Lengths of  $S$  and  $Q$ 
2  $PWM \leftarrow$  matrix of zeros of size  $(n - \ell + 1) \times (m - \ell + 1)$ 
3  $\text{dists} \leftarrow []$ 
4 if  $\epsilon = \emptyset$  then // If  $\epsilon$  is not given in the parameter
5 |  $\text{dists} \leftarrow [\text{NN1}(\text{DistanceProfile}(S_i, Q)), i \in (0; n - \ell)]$ 
6 |  $\epsilon \leftarrow \text{Mean}(\text{dists}) - \text{Std}(\text{dists})$ 
7 if  $\text{PatternIndexList} \neq \emptyset$  then
8 | foreach  $i \in \text{PatternIndexList}$  do
9 | | Call ProcessIndex( $S, Q, \ell, \epsilon, i, PWM$ )
10 else
11 | for  $i \leftarrow 1$  to  $n - \ell + 1$  do
12 | | Call ProcessIndex( $S, Q, \ell, \epsilon, i, PWM$ )
13 return PWM
```

---

**ProcessIndex (Algorithm 2).** The function *ProcessIndex*, when called with an index  $i$ , updates the values for  $PWM[i]$  (and the symmetric positions since the distance function is symmetric). The subsequence  $S_i$  consists of the  $\ell$  elements beginning at index  $i$  (line 1). The distance profile is then computed by measuring the distance between  $S_i$  and all subsequences in  $Q$  (line 2) using the selected distance function (e.g., DTW). To prevent self-matching when  $S_i$  appears in both inputs  $S$  and  $Q$ , the  $\ell$  indices around  $i$  are excluded from consideration (line 3). Subsequences where the distance is less than  $\epsilon$  are marked as  $\epsilon$ -close (line 6). To avoid redundant detection of subsequences from the same pattern, the list of indices to exclude

<sup>1</sup>Note that the values of  $\epsilon$  and *PatternIndexList* may optionally be provided as parameters when calling the algorithm. This is for optimization purposes.

---

#### Algorithm 2: ProcessIndex

---

**Input:** Time series  $S, Q$ , subsequence length  $\ell$ ,  $\epsilon$ , index  $i$ , matrix *PWM*  
**Output:** Updated *PWM* for index  $i$

```

1  $S_i \leftarrow S[i \text{ to } (i + \ell - 1)]$  // Extract subsequence  $S_i$  from  $S$ 
2  $DP_i \leftarrow \text{DistanceProfile}(S_i, Q)$ 
3  $\text{IgnoreList} \leftarrow [i - \frac{\ell}{2} \text{ to } i + \frac{\ell}{2}]$  // Indices to ignore to prevent self-matching
4 for  $j \leftarrow 1$  to  $|Q| - \ell + 1$  do
5 | if  $DP_i[j] \leq \epsilon$  &  $j \notin \text{IgnoreList}$  then
6 | |  $PWM[i, j] \leftarrow 1$  // Mark as  $\epsilon$ -close
7 | |  $\text{NewIgnoreList} \leftarrow [j - \frac{\ell}{2} \text{ to } j + \frac{\ell}{2}]$ 
8 | |  $\text{IgnoreList} \leftarrow \text{IgnoreList} \cup \text{NewIgnoreList}$ 
9 return
```

---

is continually updated by appending the  $\ell$  points surrounding each detected subsequence  $i$ , ensuring that these subsequences have not been identified in prior detections (lines 7 and 8).

## 5.2 TELESAFE Boundary Crossings Detection

The *TELESAFE* technique described in Algorithm 3 detects and classifies boundary crossings. This algorithm clusters each subsequence of the *PWM* using *DBSCAN* (hence the name *TELESAFE<sub>DBSCAN</sub>*), then labels each cluster according to the majority of the time periods to which the subsequences in the cluster correspond.

**TELESAFE<sub>DBSCAN</sub> (Algorithm 3).** The process starts by initializing a list  $L$  containing all indices of the subsequences in  $S$ . The algorithm iterates as long as  $L$  is not empty; at each iteration, a new cluster *Cluster* is initialized, and the first element in  $L$  is removed and added to the current cluster. Each cluster is built by exploring the *PWM*: for a current subsequence, every other subsequence that is  $\epsilon$ -close (i.e.,  $PWM[\text{Cluster}[i], j] = 1$ ) is considered a neighbor and added to the cluster if not already processed. This step ensures all subsequences within  $\epsilon$  distance are added to the same cluster.

**Labelling.** After constructing the clusters, the algorithm counts the number of indices in *Cluster* that are within the work period  $w$  and the private period  $\bar{w}$ . If the majority of the indices fall within  $w$ , the cluster is labelled as *WORK* and otherwise, as *PRIVATE*. Note that we label as *PRIVATE* if counts are equal. Once we process all the elements in  $L$  and we label all clusters, the algorithm returns the list of clusters with their label.

**Algorithm termination.** Note that *TELESAFE<sub>DBSCAN</sub>* is both deterministic and has proven convergence, because the clustering mechanism is based on *DBSCAN* (lines 4-19 and 28) which has both these properties. Also, each subsequence is processed exactly once and added to a cluster, ensuring each subsequence is appropriately categorized based on its proximity to others within an  $\epsilon$ -close distance and that the list  $L$  is empty at the end of the algorithm.

The labelling mechanism makes *TELESAFE<sub>DBSCAN</sub>* deterministic as it only depends on periods  $w$  and  $\bar{w}$  and the (deterministic) result of the clustering part. Our main contribution is showing how *PWM* can be used with an unsupervised clustering (and tagging)

mechanism (here DBSCAN and majority class choice) to tag each subsequence as *WORK* or *PRIVATE*, in order to detect boundary crossings. Our general approach could work with different unsupervised classification algorithms, as we show in Section 6, results using a simplified version of DBSCAN also produce very convincing results (while being more adapted to resource-constrained devices).

---

**Algorithm 3:** *TELESAFE*<sub>DBSCAN</sub>

---

**Input** : *PWM* matrix where each cell  $PWM[i, j] = 1$   
if subsequence  $S_i$  is  $\epsilon$ -close to  $Q_j$   
**Output** : Clusters of subsequences from  $S$  tagged  
as *WORK* or *PRIVATE*

```

1  $L \leftarrow$  List of all subsequence indices in  $S$ 
2  $processed \leftarrow \emptyset$ 
3  $Clusters \leftarrow []$ 
4 while  $L \neq \emptyset$  do
5    $current \leftarrow L[0]$ 
6   Remove  $current$  from  $L$ 
7   if  $current \notin processed$  then
8      $Cluster \leftarrow []$ 
9     Append  $current$  to  $Cluster$ 
10    Add  $current$  to  $processed$ 
11     $i \leftarrow 0$ 
12    while  $i < len(Cluster)$  do
13      foreach  $j$  such that  $PWM[Cluster[i], j] = 1$  do
14        if  $j \notin processed$  then
15          Append  $j$  to  $Cluster$ 
16          Add  $j$  to  $processed$ 
17          Remove  $j$  from  $L$ 
18         $i \leftarrow i + 1$ 
19     $count_w \leftarrow$  Count of WORK indices in  $Cluster$ 
20     $count_p \leftarrow$  Count of PRIVATE indices in  $Cluster$ 
21    if  $count_w > count_p$  then
22       $label \leftarrow$  WORK
23    else
24       $label \leftarrow$  PRIVATE
25    Append  $label$  to  $Cluster$ 
26    Append  $Cluster$  to  $Clusters$ 
27 return  $Clusters$ 

```

---

**Complexity.** The overall *TELESAFE* approach is composed by computing the *PWM* containing the indices of the  $\epsilon$ -close subsequences, which is parallelizable with multicore devices, and clustering the subsequences using the *PWM* structure. The bottleneck of the performance of the overall algorithm is the first stage - computing *PWM*. Space complexity does not depend on  $\ell$  and is only due to storing the whole  $n \times n$  matrix, resulting a worst case space complexity of  $O(n^2)$ . As the matrix is sparse, experimental space complexity is in fact much lower. Worst-case time complexity is derived from lines 11 and 12 in Algorithm 1 and line 2 in Algorithm 2 : Algorithm 1 calls  $O(n)$  times the distance calculation function which is  $O(n\ell^2)$ . Lines 4 to 8 of Algorithm 2 are  $O(n)$ , thus the overall worst case temporal complexity is  $O(n^2\ell^2)$ . Note that *DBSCAN*

(with a classical Python implementation) runs with a worst case temporal complexity of  $O(n^2)$  (see [27]).

**Lightweight implementation.** Given our objective of running on a low-power local device (e.g., a classical Raspberry Pi is clocked at 1.8GHz with 1 GB RAM), *TELESAFE*<sub>DBSCAN</sub> has the disadvantage of needing to cluster the whole database. In order to reduce RAM consumption, other classification algorithms can be used, provided they operate in an unsupervised setting.

A simple approach is to consider a  $k$ -NN inspired algorithm, where each subsequence, identified by its first timestamp, can be tagged as a *WORK* (resp., *PRIVATE*) depending on the fact that the  $k$  closest subsequences (using *DistanceProfile*) occur during  $w$  (resp.,  $\bar{w}$ ) or  $\bar{w}$  (resp.,  $w$ ). While respecting the lightweight constraint, this type of approach does not provide good quality results in terms of  $F_{score}$  (see next section Table 4). These bad results are due to the fact that  $k$ -NN approaches fix the number of closest neighbors considered (instead of using *PWM* which dynamically decides how many, if any, neighbors should be considered).

We thus use instead the *TELESAFE - Lite* algorithm, which builds upon the precomputation of *PWM*, while only considering the direct neighbors of a subsequence in order to determine if it is of type *PRIVATE* or *WORK*, instead of computing the whole cluster. The main difference between *TELESAFE*<sub>DBSCAN</sub> and *TELESAFE - Lite*, is that a subsequence is tagged *WORK* (resp., *PRIVATE*) if the majority of its neighbors occur during  $w$  (resp.,  $\bar{w}$ ). To further optimize *PWM* computation, time series subsequences are divided into two categories (work and private) based on the fact that they occur during  $w$  or  $\bar{w}$ , and subsequences without neighbors in the other category are pruned. This pruning is performed before calculating distances within the same category, improving performance while preserving utility.

## 6 EXPERIMENTAL EVALUATION

We now validate the results of *TELESAFE* using three real datasets and compare with state-of-the-art representative of AI-based supervised approaches. Section 6.1 describes the experimental setup and methodology. Results are shown in Section 6.2, while discussion, limitations, and challenges are addressed in Section 6.3.

### 6.1 Experimental Setup

**Datasets.** As no public datasets are specifically built for this matter of telework boundary crossings, we use several energy datasets that include timestamps and power consumption data, that we combine to model daily electricity consumption in teleworking scenarios:

- *Individual Household Electric Power Consumption (IHEPC)* from UCI [30]: We use this dataset as background data, detailing power usage in a household over nearly four years at one-minute intervals, serving as our noise floor.
- *Orange4Home* [19]: This dataset documents teleworking activities within a smart home. It includes 180 hours of recorded daily activities, well-labeled, with working periods, across 4 weeks of workdays, providing 20 workdays in total. We use the energy consumption provided by the office plugs to overlay telework activities, simulating the work consumption during a typical work period from 9 AM to 6 PM with daily variations.

**Table 3: Private activities: devices and durations**

Category	Device	Length (sec.)
Break appliances	Coffee Maker (CM)	{105, 132}
	Water Boiler (WB)	{60, 123}
	Microwave Oven (MO)	{39, 72}
White goods	Cooking Stove (CS)	{552, 603}
	Dishwasher (DW)	{1,685, 3,372}
	Laundry Dryer (LD)	{1,563, 3,521}
Brown goods	Multimedia Center (MC)	{1,253, 2,200}
	Playstation3 (PS3)	{3,621, 8,800}
	TV-LCD (TV)	{2,327, 4,252}

- *Tracebase* [50]: This dataset provides energy data for various household devices (e.g., Dishwasher, Laundry Dryer). To address boundary crossings, we inject the energy consumption of certain devices related to private activities into the daily electricity consumption. We selected devices of different categories inspired by [16] depending on their average usage duration (see Table 3), and pick two different patterns of different lengths for each device for better testing our proposal.

**Datasets combination for evaluating boundary crossings.** The process to evaluate boundary crossings involves overlaying these datasets to simulate the energy consumption of telework days, denoted as  $S_u$ , as follows:

- (1) A specific day is chosen from the IHEPC dataset, and the electricity consumption for  $S_u$  is initialized on that day, using the IHEPC dataset as a baseline.
- (2) From the Orange4Home dataset, a working day is selected, the working period  $w$  is extracted, and the consumption values are injected into  $S_u$  during this working period.
- (3) In the Tracebase dataset, private appliance patterns are selected and injected according to detailed scenarios, during  $w$  and  $\bar{w}$ .

The Tracebase and Orange4Home datasets are aggregated at 5-second intervals (the smallest common denominator), ensuring a cohesive and accurate representation of energy consumption during telework days. We also upsample the UCI dataset to match this frequency data by applying repeated samples. For each measurement scenario, we generate a set  $S = \{S_u\}$ , where each teleworker profile ( $S_u$ ) simulates the times series data of a worker for one given day, created using the process described above. Each set  $S$  contains at least 200 teleworker profiles, and aggregates at least 20 different days of IHEPC (meeting our specific criteria) and 20 different days of Orange4Home. Within these profiles, we perform at least 10 random Tracebase injections corresponding to the target scenario.

**Algorithms.** The goal of the evaluation is to measure and validate the ability of *TELESAFE* in detecting boundary crossings. We compare our proposal to regular AI-based supervised algorithms:

- *TELESAFE*: Algorithm proposed in this paper, which builds upon the computation of *PWM* and the classification of subsequences as *WORK* or *PRIVATE*. Since the algorithms are intended to run on hardware with limited resources, the *Lite* version presented in Section 5.2 is primarily considered. This *Lite* version of *TELESAFE* is also compared to *k*-NN and DBSCAN-based versions. The subsequence length  $\ell$  is set to 60, 30, 5

and 1 minute. Although this parameter is set by the teleworker, different values are used for the sake of completeness.

- Arsenal [42]: Classic Machine Learning algorithm for time series classification (denoted as ML).
- InceptionTime [34]: Deep Learning algorithm for the same task (denoted as DL).

Arsenal and InceptionTime are utilized by training them on the same datasets as *TELESAFE*, and their performance is compared using standard evaluation metrics. For Arsenal, we set the number of kernels to 100, aligning with the parameters used in a referenced NILM study [46] and the time limit of 60 minutes. For InceptionTime, the training is done on 20 epochs, determined to be sufficient after a process of incremental trial and error. We train the models with datasets from a single scenario at a time and subsequently test them with datasets from the same scenario using 5-fold cross-validation method. This allows us to reach optimal results for supervised approaches, which is beneficial for the sake of fairness.

**Evaluation platforms.** We evaluate *TELESAFE* on a Laptop HP, 13th Gen Intel(R) Core(TM) i7-1365U, with 30 GB of RAM, although we impose limits to assess the suitability for resource-constrained devices. On the other hand, the supervised solutions Arsenal and InceptionTime are executed on a server, Intel(R) Xeon(R) Gold 6248R CPU 3GHz with 251GB of RAM.

**Scenarios.** We crafted six scenarios to simulate various behaviors of teleworkers, ranging from the simplest to more complex situations:

- *I1 - Single device injection*: Our initial step involves a basic scenario where we inject a same pattern once into both  $w$  and  $\bar{w}$ . This study aims to assess the impact of work data on disrupting private device usage patterns (in which case *TELESAFE* should classify the pattern as boundary crossings).
- *I2 - Pattern deformation and length variation*: We modify the pattern lengths from I1 by injecting two patterns of different lengths for the same device, as listed in Table 3 (right column).
- *I3 - Multiple device injections without overlap*: We increase the complexity by injecting patterns from two different devices. This stage replicates I1 with two distinct device patterns, each injected once in both  $w$  and  $\bar{w}$ .
- *I4 - Overlapping device injections*: Similar to I3, but with the added complexity of overlapping patterns being injected during (one of) the two periods  $w$  or  $\bar{w}$ .
- *I5 - No injection*: To evaluate our model’s robustness, we created a scenario where standard detection conditions are not met. It uses only work data and background noise, excluding any private activity data. This setup is meant to evaluate the algorithm’s opt-out condition and its ability to correctly detect the absence of private activities.
- *I6 - Multiple occurrences*: In this scenario, we increase the pattern frequencies by injecting each pattern several times during both  $w$  and  $\bar{w}$ . Specifically, if a pattern is injected more times during  $w$  than during  $\bar{w}$ , we are evaluating the opt-out condition of *TELESAFE*. This injection is particularly important for analyzing special jobs such as gamers and journalists, where devices like TVs or gaming consoles as used for work purpose.

These scenarios are representative of a number of teleworker profiles. For instance, I3 illustrates a teleworker who takes regular

breaks (e.g, a coffee break at 11.00 am), which has been shown to have positive effects [10, 55]. I5 represents a non-stop worker who does not take any breaks, whereas I6 depicts a worker who systematically extends their working hours beyond the period stated in their contract, reflecting the growing trend of teleworkers exceeding their working hours [29]. Lastly, I2/I4 represents a worker whose breaks occur at different times each day to manage both private and work-related issues, which is one of the biggest advantages perceived by teleworkers [14].

**Evaluation metrics.** We evaluate *TELESAFE*'s performance using  $F_1$  score (noted simply  $F_{score}$ ) metrics. A prediction is made for every subsequence of length  $\ell$  contained in the work periods. We consider that a time series should be tagged as *PRIVATE* if it intersects some private activity pattern. Thus, positive ground truth (for a private boundary violation in  $w$ ) is composed of all the  $\ell$ -length intervals that intersect with a private activity pattern. More precisely, for a private pattern present in the time series from timestamp  $t_i$  to  $t_j$ , there will be  $(j - i) + 2 \times (\ell - 1)$  positive ground truth intervals, for which the starting timestamp  $k$  is such that  $k \in I_{ij}$  and  $I_{ij} = [t_i - (\ell - 1), t_j + (\ell - 1)]$ . Negative ground truth consists of all the other subsequences of length  $\ell$  during  $w$ . In terms of boundary crossings, True Positives (resp., False Negatives) are thus all the subsequences of the work period such that their starting timestamp is in  $I_{ij}$  and that are detected as *PRIVATE* (resp., *WORK*). True Negatives (resp., False Positives) are all the subsequences of the work period such that their starting timestamp is not in  $I_{ij}$  and that are detected as *WORK* (resp., *PRIVATE*).

## 6.2 Results

**Overview.** We propose and analyze experiments showing the following results. Firstly, we compare several boundaries crossing detection algorithms that we proposed in Section 5.2 (Table 4). We then compare our unsupervised approach to ML and DL approaches, showing that results are close (Figure 4), our approach sometimes even outperforms the supervised approaches (Figures 5). We also show that *TELESAFE - Lite* can be executed on a resource constrained device within a reasonable time (Figure 6). Finally, we give some results concerning the efficiency of our algorithm on a more difficult scenario: detecting boundary crossings with traces from several people (one teleworker and one non-teleworker) in Table 5.

**Comparison of algorithms proposed in Section 5.2.** In terms of boundary-crossing detection efficiency, Table 4 shows comparative  $F_{score}$  performance for various classification algorithms on the hardest classification task (I6), using a basic  $k$ -NN classifier (simply choosing a fixed number of neighbors and tagging the majority class) as lightweight classifier for *TELESAFE*. We see that *TELESAFE - Lite*, the lightweight slightly modified version of *TELESAFE<sub>DBSCAN</sub>* produces the same quality results, far outperforming naïve lightweight classification algorithms (*TELESAFE - Lite*'s  $F_{score}$  is always greater than 0.88, where naïve lightweight algorithms range between 0.44 and 0.57) with the reason in Section 5.2. For these experiments, we averaged the evaluation metrics over only 10 runs, as the only goal here was to provide a first intuition on what algorithm we should use as an unsupervised competitor to the ML and DL approaches. Due to the results shown in Table 4, we focus in the rest of the experimentation on the

*TELESAFE - Lite* version, noted simply *TELESAFE*, that we will compare with state-of-the-art ML and DL (supervised) solutions.

We now show the results of the scenarios described in Section 6.1. In each figure, the results of *TELESAFE*, Arsenal (ML) and InceptionTime (DL) are shown from the left to the right, and the average value is marked above each bar. The X-axis represents the different types of private activities considered, represented by the injection of several traces of one or more category of device (see Table 3 for the notations of the devices considered). Individual values for each device are indicated in the figures. We give the average values and compare them in the descriptive text below.

**Table 4: F-score on I6 varying *TELESAFE* classification**

		1 - NN	5 - NN	10 - NN	DBSCAN	Lite
Device	<b>PS3</b>	0.57	0.57	0.54	0.92	0.95
	<b>TV</b>	0.53	0.48	0.45	0.93	0.91
	<b>CS</b>	0.45	0.44	0.44	0.90	0.88

**Results for I1.** Results are presented in Figures 4a-4c-4e. The injected device lengths used are highlighted in bold in Table 3. The overall average  $F_{score}$  of *TELESAFE* stands at 0.92 which is satisfying. Comparing with supervised methods, *TELESAFE* demonstrates a  $F_{score}$  that is close: it is on average 3.1% lower than the ML approach and 5.6% lower than the DL approach. One can note that the Multimedia Center (MC) device yielded the least favorable results, due to its usage pattern closely resembling that of a typical screen. Detecting the private MC activity remains the most difficult task, regardless of the approach (*TELESAFE*, ML or DL). Nevertheless, on the whole *TELESAFE* provides results that are very close to the ones produced in a supervised setting, while being unsupervised.

**Results for I2.** Results are presented in Figures 4b-4d-4f. Again, *TELESAFE* is very close to its supervised counterparts: *TELESAFE* is only 5.5% and 8.0% lower in  $F_{score}$  than the ML and DL approaches on average. Despite these discrepancies, the overall average  $F_{score}$  of the *TELESAFE* stands at 0.90, which seems satisfactory. The primary limiting factor remains the Multimedia center, for the same reasons outlined in I1.

**Results for I3.** Results are presented in Figure 5a. Again, and despite the increasing difficulty of the task (there are now two different types of private activities that are injected), *TELESAFE* is only 11% and 13% lower in  $F_{score}$  than the ML and DL approaches on average, respectively. The scenario involving the television and Multimedia Center showed the worst results due to the difficulty in detecting the Multimedia Center with the same reason as in I1. In the combined case of CM+MO, the performance shows a slight decrease versus the separate evaluations of these devices in I1. Similarly, the results for CS+DW are inferior to those of the individual assessments in I1 because the durations of device usage differ. Despite the limitations, the overall average  $F_{score}$  is 0.84.

**Results for I4.** As presented in Figure 5b, the effectiveness of *TELESAFE* shows a decline of only 5.3% and 6.0% compared to ML and DL methods for  $F_{score}$ . This decrease in performance, particularly pronounced in the DW+LD scenario, is due to the overlapping usage patterns blurring the distinct patterns, challenging the method to identify pattern similarities. Furthermore, the results

from CM+MO increase slightly compared to those in I3. The overall average  $F_{score}$  is 0.89, which is deemed satisfactory.

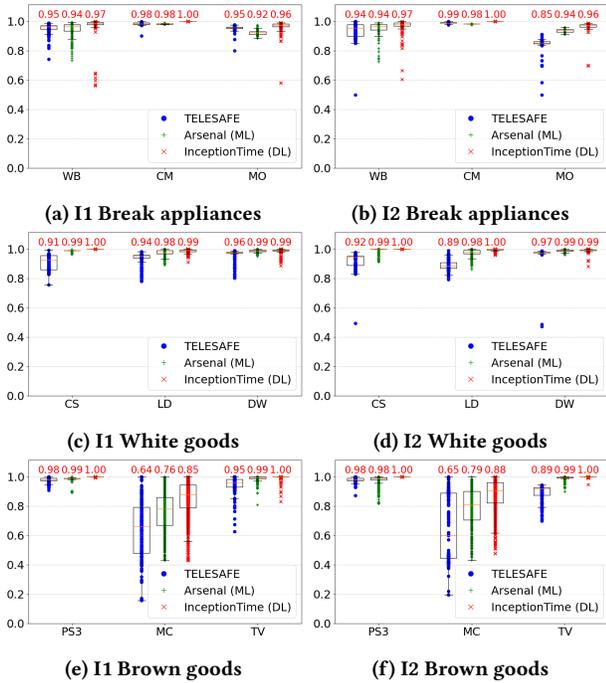


Figure 4: F-score results for I1 and I2.

**Results for I5.** Interestingly, in the next two scenarios (Figure 5), which are more complex, *TELESAFE* outperforms some supervised approaches. Results of I5 are presented in Figure 5c. This scenario, if applied directly, would heavily favor supervised approaches due to the highly unbalanced classes. The *best performing* classifier in this case would be the constant classifier, always outputting “no boundary crossing”. To avoid this, we mix the I5 time series with I1 time series during training but evaluate the results on the I5 instances. As *TELESAFE* does not require training, this has no effect on its results. The two supervised algorithms show similar or slightly worse effectiveness than *TELESAFE*, highlighting its strength. This shortcoming in supervised approaches may be due to the limited amount of data in the training set, underscoring *TELESAFE*’s advantage in preserving privacy without requiring training data. *TELESAFE* achieves  $F_{score}=0.98$ , which is a 0.5% and 1.0% increase for  $F_{score}$  for ML and DL.

**Results for I6.** Results are presented in Figure 5d. We applied the same strategy as in the I5 scenario to avoid the issue of unbalanced classes. *TELESAFE* clearly outperforms the ML and DL algorithms, as their analysis is confined to each individual window, without the capability to track and count pattern occurrences in the whole time series. *TELESAFE*’s average  $F_{score}$  is 0.93 completely outperforming by 20% to 39% the ML and DL techniques, which again highlight the advantage of our approach, particularly its robustness and efficiency in handling complex pattern recognition tasks in the context of teleworking.

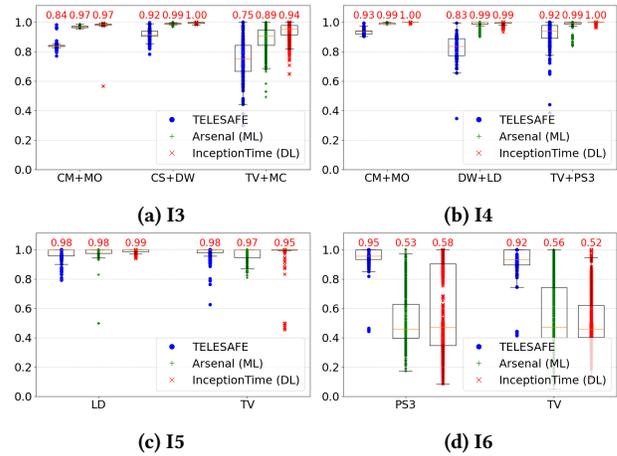
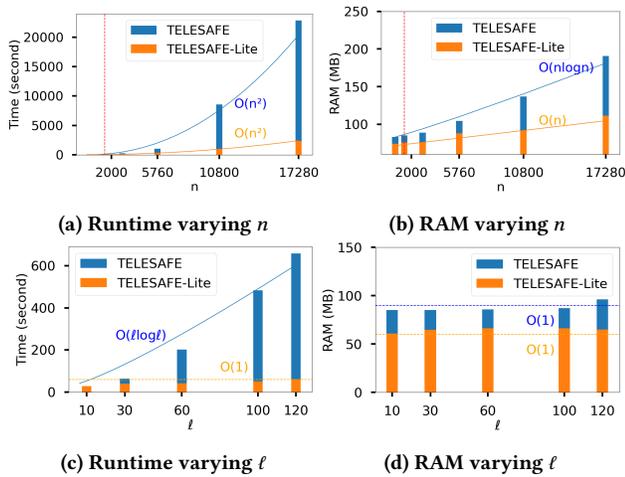


Figure 5: F-score results for I3, I4, I5 and I6.

**Performance and scalability.** To address the constraints outlined in Section 2.3 (requiring minimal computing resources for execution on IoT devices like smart meters or Raspberry Pi), we simulated a real execution environment. Our application ran in a Docker container configured to mirror the settings of a Raspberry Pi [47]. Thus, we built an image with an ARM v8 64-bit architecture, limiting RAM usage to 1GB. Figure 6 illustrates the scalability of *TELESAFE*<sub>DBSCAN</sub> (denoted as *TELESAFE* in the figures for simplicity) and *TELESAFE – Lite* with respect to the parameters  $n$  and  $\ell$ . In figures 6a and 6b, we evaluate the runtime (in seconds) and the RAM (in MB) of the two algorithms on time series of length  $n = 720$  (one day at 2 mins polling) to  $n = 17280$  (at 5 sec polling).  $\ell$  is fixed at 30, and the most commonly used sampling frequency (at 1 min polling, thus  $n = 1440$ ) is indicated by a vertical line. In figures 6c and 6d,  $n$  is fixed at 1440, with a maximum meaningful value  $\ell = 120$ . Indeed, the shortest duration of a  $w$  or  $\bar{w}$  in our dataset is 120 min, thus any  $\ell > 120$  would include more than a whole  $w$  or  $\bar{w}$  period, rendering the approach meaningless. Note that both algorithms remain well within the 1 GB RAM limit (always under 200 MB as shown in figures 6b and 6d). This figure shows that *TELESAFE* scales *experimentally* in  $O(n \log n)$  for RAM usage and  $O(n^2 \ell \log \ell)$  for time. *TELESAFE – Lite* scales in  $O(n)$  for RAM usage and  $O(n^2)$  for time, with a better constant compared to the non-optimized *TELESAFE*, while using very limited computing resources. *TELESAFE – Lite* does not depend on  $\ell$  for runtime, and neither *TELESAFE* nor *TELESAFE – Lite* depend on  $\ell$  for RAM usage (approx 90 MB for *TELESAFE* and 60 MB for *TELESAFE – Lite* with a fixed  $n = 1440$ ).

With more computing resources (e.g., more cores), the *PWM* precomputation can be parallelized, with a theoretical speedup factor of up to the minimum value of  $n$  and the number of cores. Experimental results show that for a time series with 1440 points, the *PWM* computation achieves speedups of 1.66, 1.79, and 3.69 when using 2, 4, and 8 cores, respectively. One can compute the distances on the fly during the clustering phase, which will save memory by not storing the *PWM*. However, as showed in Figure 6, the memory consumption of the algorithm is low (several hundred



**Figure 6: RAM and runtime of *TELESAFE-Lite* v.s. *TELESAFE* (avg. 10 runs). The red vertical bars show the maximum realistic value for  $n$ .**

MB), while time consumption is the constraining factor (but only a few minutes in realistic settings). We thus highlight that the precomputation is an interesting trade-off of memory in favor of execution time.

**Multiple user context.** We also consider the scenario where multiple people are present in the same home, such as a teleworker with a non-working person (child, parent, etc.). The test follows the same experimental framework as experiment I6, but with different background data (IHEPC data). Results are shown in Table 5, where the single-user (S-U) values are from Figure 5d, and multi-user (M-U) values are averaged over 10 runs. To simulate a person doing personal activities during another’s work, we selected days with active devices during work periods instead of constant-load devices. Using 10 random days from the IHEPC dataset (representing the non-working person activity) combined with one trace from the Orange4Home dataset, and considering TV use as a private activity for the teleworker, we achieved an average  $F_{score}$  of 0.84, with an interesting absence of false negatives (all private activities were recognized). False positives arose when the non-working person more performed the same private activities more frequently during private periods, or the same TV pattern appeared during work periods. This highlights a limitation of *TELESAFE*, as it can detect breaks but cannot determine which individual made the break.

**Table 5: Boundary crossings detection with multiple users**

	M-U $F_{score}$	M-U Acc.	S-U $F_{score}$	S-U Acc.
TV	0.835	0.836	0.91	0.95
PS3	0.964	0.962	0.95	0.98

### 6.3 Discussion, Limitations and Challenges

We have demonstrated that *TELESAFE* meets the constraints outlined in Section 2.3 while delivering results comparable to those of

supervised NILM AI-based algorithms. It is lightweight and can run locally on constrained devices, requiring no external communication except result sharing. Additionally, it relies solely on data from smart meters, eliminating the need for home instrumentation, and does not require training. *TELESAFE* shows satisfactory effectiveness (with  $F_{score} > 0.8$ ) in identifying boundary crossings across all tested scenarios, regardless of the number of devices or variations in injection lengths. The adaptability of *TELESAFE* to various operational contexts provides significant advantages, making it a versatile tool for teleworking environments.

**Limitations and challenges.** There are four main limitations of *TELESAFE*. Firstly, the similarity between the patterns of work and non-work devices. This was the case of the Multimedia Center, whose consumption trail is similar to computing devices. Secondly, the concurrent use of devices, as seen in I4 with dishwashers and laundry dryers, becomes a problem when this happens only in work/private periods. Thirdly, *TELESAFE* detects private activities, which are not always equivalent to breaks. When teleworkers operate white goods while working, false positives may arise if these patterns are more frequent in private periods. Lastly, the absence of a real annotated consumption dataset in the telework context limits *TELESAFE*’s ability to generalize across other environments.

*TELESAFE* faces a number of open challenges. For instance, including instantaneous personal activities (e.g., hairdryer, water fountain) and multiple simultaneous devices is necessary. Furthermore, it should be expanded to other numerical time series data, such as sounds, bandwidth consumption or accelerometers. Lastly, dealing with how to identify the user behind each activity is necessary for scenarios with multiple users (e.g., several teleworkers working on different schedules).

## 7 CONCLUSION AND FUTURE WORK

Teleworking has surged in the COVID-19 era, providing benefits like increased flexibility and reduced commuting costs for employees and employers. However, it also presents challenges in evaluating workers’ quality of life, respect for labor laws, and accurate measurements. This paper investigates a key telework issue: detecting boundary crossings between personal and work activities. We introduce *TELESAFE*, a new approach that distinguishes these activities using electricity consumption data while respecting four constraints: non-intrusive, unsupervised, local and suitable for resource-limited devices. Our method utilizes generalized distance profiles, validated through teleworkers’ energy consumption data. Compared with two state-of-the-art machine and deep learning techniques, *TELESAFE* demonstrates satisfactory performance, often outperforming supervised approaches in complex scenarios. As a primer contribution in this direction, a number of limitations and challenges have already been identified. As such, we hope *TELESAFE* will pave the way for future works in this direction, thus addressing more complex scenarios.

## ACKNOWLEDGEMENTS

This work was supported by grants “ANR-22-PECY-0002” IPoP project, “ANR-23-CMAS-0019” CyberINSA, DATAIA and PID2023-150310OB-I00 (MORE4AIO) funded by MCIU/ AEI / 10.13039/501100011033 / FEDER, UE.

## REFERENCES

- [1] California Labor Code, 1937. Available from: <http://leginfo.ca.gov>.
- [2] Right to disconnect: Implementation and impact at company level, 2023. Accessed: 2024-05-17.
- [3] The smart meter rollout in germany and europe, 2023. Accessed: 2024-05-17.
- [4] J. A. Aitken, S. A. Kaplan, and L. Kuykendall. Going with (out) the flow at home: day-to-day variability in flow and performance while teleworking. *European Journal of Work and Organizational Psychology*, 32(5):662–677, 2023.
- [5] S. Alaei, K. Kamgar, and E. Keogh. Matrix profile xxii: exact discovery of time series motifs under dtw. In *2020 IEEE international conference on data mining (ICDM)*, pages 900–905. IEEE, 2020.
- [6] T. Allard, N. Anciaux, L. Bouganim, Y. Guo, L. Le Folgoc, B. Nguyen, P. Pucheral, I. Ray, I. Ray, and S. Yin. Secure personal data servers: a vision paper. 3(1–2):25–35, Sept. 2010.
- [7] N. Anciaux, L. Bouganim, P. Pucheral, I. S. Popa, and G. Scerri. Personal database security and trusted execution environments: A tutorial at the crossroads. *Proc. VLDB Endow.*, 12(12):1994–1997, 2019.
- [8] G.-F. Angelis, C. Timplalexis, S. Krinidis, D. Ioannidis, and D. Tzovaras. Nilm applications: Literature review of learning approaches, recent developments and challenges. *Energy and Buildings*, page 111951, 2022.
- [9] G. F. Angelis, C. Timplalexis, A. I. Salamani, S. Krinidis, D. Ioannidis, D. Kehagias, and D. Tzovaras. Energformer: A new transformer model for energy disaggregation. *IEEE Transactions on Consumer Electronics*, 2023.
- [10] A. P. Association. Give me a break. *American Psychological Association*, 2019.
- [11] N. Batra, J. Kelly, O. Parson, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh, and M. Srivastava. Nilmkt: An open source toolkit for non-intrusive load monitoring. In *Proceedings of the 5th international conference on Future energy systems*, pages 265–276, 2014.
- [12] R. J. Bayardo, Y. Ma, and R. Srikant. Scaling up all pairs similarity search. In *Proceedings of the 16th international conference on World Wide Web*, pages 131–140, 2007.
- [13] N. Büchler, C. L. ter Hoeven, and W. van Zoonen. Understanding constant connectivity to work: How and for whom is constant connectivity related to employee well-being? *Information and Organization*, 30(3):100302, 2020.
- [14] P. R. Center. 35% of workers who can work from home now do this all the time in u.s. *Pew Research Center*, 2023.
- [15] H. S. Cho, T. Yamazaki, and M. Hahn. Aero: extraction of user’s activities from electric power consumption data. *IEEE Transactions on Consumer Electronics*, 56(3):2011–2018, 2010.
- [16] L. Ciabattini, G. Comodi, F. Ferracuti, and G. Foresi. Ai-powered home electrical appliances as enabler of demand-side flexibility. *IEEE Consumer Electronics Magazine*, 9(3):72–78, 2020.
- [17] CNIL. Les questions-réponses de la cnil sur le télétravail, 2020. <https://www.cnil.fr/fr/les-questions-reponses-de-la-cnil-sur-le-teletravail>, Accessed: 2024-09-27.
- [18] Council of the European Communities. Council directive 89/391/eec of 12 June 1989 on the introduction of measures to encourage improvements in the safety and health of workers at work. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A31989L0391>, 1989. Accessed: 2024-05-17.
- [19] J. Cumin, G. Lefebvre, F. Ramparany, and J. L. Crowley. A dataset of routine daily activities in an instrumented home. In *Ubiquitous Computing and Ambient Intelligence: 11th International Conference, UCAmI 2017, Philadelphia, PA, USA, November 7–10, 2017, Proceedings*, pages 413–425. Springer, 2017.
- [20] G. de España. Ley 23/2015, de 21 de julio, ordenadora del sistema de inspección de trabajo y seguridad social, 2015. Accessed: 2024-05-17.
- [21] G. de la République Française. *Code du travail*. 1973. Accessed: 2024-05-17.
- [22] Y.-A. de Montjoye, E. Shmueli, S. Wang, and A. S. Pentland. openpds: Protecting the privacy of metadata through safeanswers. *PLoS ONE*, 9, 2014.
- [23] B. Deutschland. *Gesetz über die Durchführung von Maßnahmen des Arbeitsschutzes zur Verbesserung der Sicherheit und des Gesundheitsschutzes der Beschäftigten bei der Arbeit (Arbeitsschutzgesetz - ArbSchG)*. 1996. Accessed: 2024-05-17.
- [24] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.
- [25] A. Eira. 71 vital remote work and telecommuting statistics: 2024 data analysis predictions, 2024. <https://financesonline.com/remote-work-and-telecommuting-statistics/>, Accessed: 2025-03-02.
- [26] P. Esling and C. Agon. Time-series data mining. *ACM Comput. Surv.*, 45(1), dec 2012.
- [27] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, and U. M. Fayyad, editors, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, Portland, Oregon, USA, pages 226–231. AAAI Press, 1996.
- [28] A. Ferrari, D. Micucci, M. Mobilio, and P. Napolitano. Deep learning and model personalization in sensor-based human activity recognition. *Journal of Reliable Intelligent Environments*, 9(1):27–39, 2023.
- [29] E. F. for the Improvement of Living and W. Conditions. The rise in telework: Impact on working conditions and regulations. *Eurofound*, 2023.
- [30] G. Hebrail and A. Berard. Individual Household Electric Power Consumption. UCI Machine Learning Repository, 2012. DOI: <https://doi.org/10.24432/C58K54>.
- [31] E. M. Hunter, M. A. Clark, and D. S. Carlson. Violating work-family boundaries: Reactions to interruptions at work and home. *Journal of Management*, 45(3):1284–1308, 2019.
- [32] InterGuard. Employee monitoring software remote monitoring of employee computer activity. <https://www.interguardsoftware.com/employee-monitoring-software/>, Accessed: 2024-09-27.
- [33] K. Ishizu, T. Mizumoto, H. Yamaguchi, and T. Higashino. Home activity recognition using aggregated electricity consumption data. In *2020 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 302–307. IEEE, 2020.
- [34] 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*, 34(6):1936–1962, 2020.
- [35] D. G. Korzun, S. I. Balandin, and A. V. Gurtov. Deployment of smart spaces in internet of things: Overview of the design challenges. *Internet of Things, Smart Spaces, and Next Generation Networking*, pages 48–59, 2013.
- [36] E. G. Lambert, R. Worley, and V. B. Worley. The effects of perceptions of staff-inmate boundary violations and willingness to follow rules upon work stress. *Security Journal*, 31:618–644, 2018.
- [37] A. Lentzas and D. Vrakas. Machine learning approaches for non-intrusive home absence detection based on appliance electrical use. *Expert Systems with Applications*, 210:118454, 2022.
- [38] Y. Li, G. Yang, Z. Su, S. Li, and Y. Wang. Human activity recognition based on multi-environment sensor data. *Information Fusion*, 91:47–63, 2023.
- [39] M. Linardi, Y. Zhu, T. Palpanas, and E. Keogh. Matrix profile x: Valmod-scalable discovery of variable-length motifs in data series. In *Proceedings of the 2018 international conference on management of data*, pages 1053–1066, 2018.
- [40] M. Linardi, Y. Zhu, T. Palpanas, and E. Keogh. Valmod: A suite for easy and exact detection of variable length motifs in data series. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1757–1760, 2018.
- [41] Y. Mao, Q. Ye, Q. Wang, and H. Hu. Differential privacy for time series: A survey. *IEEE Data Eng. Bull.*, 47(2):67–92, 2024.
- [42] 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*, 110(11):3211–3243, 2021.
- [43] T. Mondal, R. Akbarinia, and F. Masseglia. knn matrix profile for knowledge discovery from time series. *Data Mining and Knowledge Discovery*, 37(3):1055–1089, 2023.
- [44] C. of Europe. European convention on human rights, 1950. Accessed: 2024-05-17.
- [45] U. S. D. of Labor. *Occupational Safety and Health Act of 1970 (OSH Act)*. 1970. Accessed: 2024-05-17.
- [46] A. Petralia, P. Charpentier, P. Boniol, and T. Palpanas. Appliance detection using very low-frequency smart meter time series. *ACM e-Energy* 2023, 2023.
- [47] R. Pi. Raspberry pi 4. <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>.
- [48] Z. Qin, Y. Zhang, S. Meng, Z. Qin, and K.-K. R. Choo. Imaging and fusing time series for wearable sensor-based human activity recognition. *Information Fusion*, 53:80–87, 2020.
- [49] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(3):1–31, 2013.
- [50] 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)*, pages 1–9. IEEE, 2012.
- [51] P. Schäfer and U. Leser. Motiflets: Simple and accurate detection of motifs in time series. *Proceedings of the VLDB Endowment*, 16(4):725–737, 2022.
- [52] S. Singh and A. Yassine. Mining energy consumption behavior patterns for households in smart grid. *IEEE Transactions on Emerging Topics in Computing*, 7(3):404–419, 2017.
- [53] A. Sivunen, J. L. Gibbs, and J. Leppäkumpu. Managing collapsed boundaries in global work. *Journal of Computer-Mediated Communication*, 28(4):zmad019, 2023.
- [54] C. Tagliaferri. Remote work burnout statistics: Trends and insights to consider in 2024, 2023. <https://thinkremote.com/remote-work-burnout-statistics/>, Accessed: 2024-09-27.
- [55] A. Technical. Surprising working from home productivity statistics. *Apollo Technical LLC*, 2024.
- [56] J. Vitak and M. Zimmer. Surveillance and the future of work: exploring employees’ attitudes toward monitoring in a post-COVID workplace. *Journal of Computer-Mediated Communication*, 28(4):zmad007, 06 2023.
- [57] H. Wang and C. Wu. Privacy preservation for time series data in the electricity sector. *IEEE Transactions on Smart Grid*, 2022.

- [58] 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*, 289:162, 2022.
- [59] S. Wilhelm and J. Kasbauer. Exploiting smart meter power consumption measurements for human activity recognition (har) with a motif-detection-based non-intrusive load monitoring (nilm) approach. *Sensors*, 21(23):8036, 2021.
- [60] C.-C. M. Yeh, N. Kavantzias, and E. Keogh. Matrix profile iv: using weakly labeled time series to predict outcomes. *Proceedings of the VLDB Endowment*, 10(12):1802–1812, 2017.
- [61] 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 *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1317–1322. Ieee, 2016.
- [62] Y. Zhu, S. Gharghabi, D. F. Silva, H. A. Dau, C.-C. M. Yeh, N. Shakibay Senbari, A. Almaslukh, K. Kamgar, Z. Zimmerman, G. Funning, et al. The swiss army knife of time series data mining: ten useful things you can do with the matrix profile and ten lines of code. *Data Mining and Knowledge Discovery*, 34:949–979, 2020.