

Online Binary Models are Promising for Distinguishing Temporally Consistent Computer Usage Profiles

Luiz Giovanini*, Fabrício Ceschin*, Mirela Silva, Aokun Chen, Ramchandra Kulkarni, Sanjay Banda, Madison Lysaght, Heng Qiao, Nikolaos Sapountzis, Ruimin Sun, Brandon Matthews, Dapeng Oliver Wu, André Grégio, and Daniela Oliveira

Abstract—This paper investigates whether computer usage profiles comprised of process-, network-, mouse-, and keystroke-related events are unique and consistent over time in a naturalistic setting, discussing challenges and opportunities of using such profiles in applications of continuous authentication. We collected ecologically-valid computer usage profiles from 31 MS Windows 10 computer users over 8 weeks and submitted this data to comprehensive machine learning analysis involving a diverse set of online and offline classifiers. We found that: (i) profiles were mostly consistent over the 8-week data collection period, with most (83.9%) repeating computer usage habits on a daily basis; (ii) computer usage profiling has the potential to uniquely characterize computer users (with a maximum F-score of 99.90%); (iii) network-related events were the most relevant features to accurately recognize profiles (95.69% of the top features distinguishing users were network-related); and (iv) binary models were the most well-suited for profile recognition, with better results achieved in the online setting compared to the offline setting (maximum F-score of 99.90% vs. 95.50%).

Index Terms—Computer user profiling, continuous authentication, machine learning, time series analysis, user study.

1 INTRODUCTION

COMPUTER user profiling is the procedure of constructing a behavior-based digital identity of a person by leveraging their computer usage data, such as network traffic, process activity, mouse and keyboard dynamics [1], [2]. This procedure has the potential to uniquely characterize computer users by their usage patterns in terms of activity (e.g., process and network events) and temporal consistency (i.e., events repeating on a regular basis) [3], [4], [5]. For example, a middle-aged CEO from a Fortune 500 company uses his computer very differently from a young software developer in a tech startup in California. While the former might use email client, office software, web browser, and customized company management software, the latter will likely use IDE, CAD, and software versioning tools, alongside a web browser, email client, and regularly accesses .com.br websites given their Brazilian descent.

Several factors make computer usage profiles (referred

to as simply *profiles* for the remainder of this paper) well-suited for applications in continuous authentication (CA). The automatic recording of profiles can be implemented in a transparent fashion without requiring user intervention, thus facilitating usability and acceptability [6]. Furthermore, especially with regards to corporate employees, although some might perceive the recording of computer usage as potentially privacy-invasive, most already have a very limited expectation of privacy [7]. In fact, some employee activities (e.g., email and computer usage) are often recorded by their employers in both the public and private sectors [8].

This paper does **not** propose a CA solution targeting personal or corporate computer users. Instead, based on naturalistic data collected from an ecologically-valid user study, we seek to provide initial evidence on whether profiles constitute a *useful* and *feasible* source of data to uniquely identify computer users while they use their devices, discussing challenges and opportunities of leveraging profiles for applications in CA. Towards this end, we also investigated *temporal changes* in profiles (i.e., changes in computer usage habits taking place over time), which may impose additional challenges for CA and inform the design of more robust CA solutions. Our study focused on answering the following research questions:

- **RQ1:** Are computer usage profiles consistent over time (i.e., do computer usage habits repeat periodically)?
- **RQ2:** Do computer users have unique profiles? In other words, are profiles distinguishable from one another?
- **RQ3:** What features (e.g., network events, processes) are most important for constructing a unique profile?

To address these questions, we conducted an Institu-

- L. Giovanini (corresponding author), M. Silva, A. Chen, H. Qiao, D. Oliver Wu, and D. Oliveira are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: lfrancogiovanini@ufl.edu).
- F. Ceschin and A. Grégio are with the Department of Informatics, Federal University of Paraná, Curitiba, PR Brazil.
- S. Banda, M. Lysaght, and R. Kulkarni are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA.
- N. Sapountzis is with Cisco, San Jose, CA 95134 USA.
- R. Sun is with the Khoury College of Computer Science, Northeastern University, Boston, MA 02115 USA.
- B. Matthews is with the Electro-Optical Systems Laboratory, Georgia Tech Research Institute, Atlanta, GA 30318 USA.
- Charles Stark Draper Laboratory, Cambridge, MA 02139 USA.

*The first two authors have equal contribution.

Manuscript received Month dd, yyyy; revised Month dd, yyyy.

tional Review Board (IRB) approved user study to collect profiles from 31 computer users (MS Windows 10) over an 8-week period (taking place asynchronously between September 2020 and March 2021) in an ecologically-valid setting, wherein users interacted with their computers naturally without any interference or probing from the research team. Importantly, our definition of profiles encompasses process-, mouse-, network-, and keystroke-related events on the users' computers. We then used time series analysis techniques to investigate temporal changes in the profiles over the study period (**RQ1**). Our dataset was also submitted to a comprehensive machine learning analysis involving a diverse set of classifiers (one-class and binary models in both offline and online settings) aimed to assess profile uniqueness (**RQ2**) and identify the most important features in distinguishing among profiles (**RQ3**). All artifacts created throughout the course of this study, including our module for extracting profiles and our dataset of ecologically-valid computer usage data itself, will be made publicly available for vetted research usage.¹

The main takeaways of our analyses are: (1) profiles were mostly consistent over time and repeating on a daily basis, yet exhibited some level of irregularity related to factors such as days off, and change of habits due to the COVID-19 pandemic; (2) computer usage profiles have the potential to uniquely identify computer users; (3) network features were the most relevant to accurately distinguish users; and (4) online binary models are preferred over offline binary and one-class models due to better performance in distinguishing profiles and robustness to temporal changes in computer usage.

The remainder of the paper is organized as follows. Section 2 discusses the threat model faced by CA approaches and the assumptions of our analysis. Section 3 reviews related work. Section 4 presents our user study methodology. Section 5 describes the design and implementation of the profile extractor used to gather users' computer usage data. Section 6 describes the methodology of our machine learning analyses. Section 7 presents our experimental results. Section 8 discusses study's findings, limitations, and suggestions for future work. Section 9 concludes this paper.

2 THREAT MODEL AND ASSUMPTIONS

This section discusses the main assumptions of this work and the threat model faced by CA approaches. First, we assume that CA methods are better suited to be employed in corporate environments, which are disproportionately targeted by internal and external adversaries, and consists of large groups of employees performing their primary tasks with a institutional computer device. Second, although our discussion is focused on CA, this paper does not propose yet a new CA solution. Third, we consider that corporate employees already face low expectations of privacy in their work environment [8]. For example, in many organizations, network traffic, files and emails are monitored, and the devices and applications employees can access are restricted.

1. We are discussing with our IRB and legal counsel about options of data release based on University-mediated agreements with vetted researchers to protect the de-identified dataset to ensure that the data is used for ethical and responsible research only.

The development of a CA solution leveraging computer usage profiles involves the recording of activity. Despite that we do not advocate the development of privacy-invasive CA solutions without the users being aware of their low expectations of privacy.

Threat Model. As discussed in Sec. 1, the main appeal of a CA system is not to replace traditional, point-of-entry authentication schemes (e.g., passwords or security keys), but to complement them and address their limitations—mainly that, after a user is authenticated into the system, their identity is not subsequently verified. A CA approach can potentially flag events that do not fit a learned user profile, e.g., connections to certain IP addresses or subnets, and patterns of exfiltration of information even after the user is authenticated, including insider attacks based on activity. For example, it is plausible that employees working on the same project will have similarities in profiles (e.g., same applications, files, schedules). Thus, in constructing a group profile, the malicious employee's outlier behavior (i.e., an insider attack) could be flagged as unusual activity. Given the challenges and privacy, security, and legal implications of conducting such study in a private corporation or government organization, we conducted our study with mainly university students and considering this large, diverse university environment as a proxy of an organization.

3 RELATED WORK

There is a vast literature on leveraging user profiling for CA using a variety of data, such as network traffic (e.g., [2], [9]), use of applications (e.g., [4], [10]), and keystroke or mouse dynamics [11], [12], [13], [14]. User profiling has also been used for intrusion detection, where deviations from a "normal behavior" are considered as intrusions [15]. There is, nonetheless, a scarcity of publicly available and diverse datasets containing users' application data [10], [16]. Some large-scale micro-longitudinal datasets collected from several user study participants are available (e.g., [17], [18]), but primarily focus on smartphone data instead of ecologically valid data from a personal laptop or desktop computer.

A notable exception is the dataset by Murphy et al. [16], wherein the keystroke data, mouse movements and clicks, and background processes were collected from 103 users over 2.5 years on each participant's private computer in an ecologically valid setting. The authors found that authentication performance degraded significantly when data was collected in less controlled environments. Though this dataset is larger than ours, it did not include network metadata, which in our analysis, was the most relevant type of data to distinguish user profiles; we additionally did not collect keystrokes from users. Moreover, Murphy et al. did not analyze the temporal consistency of the profiles.

One line of research that closely resembles ours is that of Payne et al. [5], where the authors collected system events from seven volunteers for two hours in a controlled environment. However, unlike our user study, time information was not associated with user activities. Their experiment was also not ecologically valid because the users were not using their own devices nor their computers naturalistically. López et al. [19] used Self-Organizing Maps (SOM) to enhance insight and interpretability about user behavior data

by untangling hidden relationships between variables. The authors generated the SOM visualizations of survey results that analyzed user behavior, security incidents, fraud, and data from a malware scanning tool to contrast Internet users' digital confidence with the level of malware infection. Their approach focused on drawing qualitative conclusions from their self-reported dataset. In contrast, we go beyond by exploring a naturalistic dataset quantitatively using multiple time series analysis methods and machine learning models.

In sum, our review of related work on user profiling for CA showed that none of the existing works evaluated the feasibility of desktop-based computer usage profiling in a naturalistic way, as proposed here. Instead, many data collection procedures restricted user behaviors to specific tasks [20], [21] or devices [22]. Some CA proposals operate by building profiles based on short-term user activity records [5], while others did not explore changes in user behavior over time [21], [23]. Thus, though these studies provide valuable insight into what technologies can be employed for computer usage profiling, this procedure's feasibility and temporal robustness remains understudied. Our findings shed light on such aspects, providing an actionable recommendation for future research and designing of effective, micro-longitudinal, behavior-based CA systems.

4 USER STUDY METHODOLOGY

This IRB-approved study requested that participants installed our extractor module on their personal computers and used their devices naturally for 8 weeks. The entire study took place asynchronously from September 2020 to March 2021, wherein we successfully captured 8 weeks of computer usage data from 31 of the total 63 participants enrolled. In this section, we detail our methodology.

Participants. The study was originally comprised of 63 participants who were recruited via SONA, flyers, Internet advertising (e.g., the UF Facebook page that advertises studies), and word-of-mouth. Interested individuals were guided towards an online survey to determine eligibility which, if cleared to participate, would ask the participant for informed consent along with demographic information. After study completion, the participants who were recruited via SONA were compensated with two course credits, while the remainder with a \$50 Amazon Gift Card. For inclusion in the data analysis, participants were required to complete the entire 8-week study period. This excluded 32 of the original 63 enrolled participants (who experienced technical issues as discussed below), thus reducing our sample size to 31 participants ranging from 18–53 years ($M = 27.32$ years, $SD = 8.27$; 64.52% female, 32.26% male, 3.23% gender non-conforming). Table 1 in Appendix A summarizes the demographics of our participants. On average, we recorded 272 hours of computer usage data from each participant ($SD = 189.9$ hours, range: 31 – 859), which corresponds to approximately 4.9 hours per study day ($SD = 3.4$ hr/day, range: 0.6 – 15.3). We found some instances of high computer usage (e.g., 15.3 hr/day) that we hypothesize may be due to the participant leaving their device turned on for extended periods of time.

Procedure. At the beginning of the study, enrolled participants first were asked to complete an online survey to

determine their eligibility: (i) above 18 years of age, (ii) own a personal computer (desktop or laptop) that is not shared and (iii) is used regularly, (iv) have regular access to the Internet, (v) have Windows 10 Operating System installed, and (vi) reside in the United States (as per UF IRB regulations for compensation). Once the participants were deemed eligible, they received a consent form disclosing the study procedures, minimal study risks, and data protection measures. The study did not involve deception in that participants were informed of the actual research purpose: continuously record their computer activities for 8 weeks to increase understanding of computer usage profiles. Participants were informed that keyboard keys and data sent over the network such as file contents **would not** be recorded, but that the following **would** be collected throughout the duration of the study: metadata related to software and network activity. Having read and electronically signed the informed consent form, participants were asked a series of demographic questions (e.g., age, ethnicity) and to install our extractor on their personal computer. Of note, our informed consent was approved with fairly general language that made it clear that we would be installing monitoring software on the participants' computers and collecting various kinds of computer usage information. This language was deemed clear and sufficient by UF IRB for purposes of informed consent because it is common to use language that omits technical terms/jargon pertaining to the study focused research area. Certain jargon would not only be meaningless to most subjects, but also possibly give them a false sense of security towards the power of these procedures or engender a false sense of protection against potential risks the study entails. This extractor would record logs of system-level events and uploads them to our lab servers in a secure fashion. For each participant, the 8-week study period began on the day successful installation could be verified within our systems. Participants were instructed to use their computers naturally and were reminded not to share their personal computers with anyone while in the study. Only IRB-trained research assistants who were part of the project interacted with participants to assist them with any questions or concerns they might have. Upon completion of the study and after uninstalling the extractor, participants were asked to complete a final debriefing questionnaire comprised of seven questions pertaining to the consistency of their computer usage during the study period. After completing the debrief, participants were compensated.

Data Attrition. A few issues occurred during the initial run of the study. First, because international participants were ineligible as per UF IRB, we discarded three participants that were located outside the U.S. We compensated them nonetheless (with UF IRB approval) and discarded their collected data. Second, our extractor only collected the destination IP address from 41 participants, thus we were unable to resolve which domains they accessed. To remedy this situation, we began capturing DNS Queries to resolve the destination IP address to its respective domain. All 41 participants who encountered this issue were communicated about an issue with data collected and invited to restart the study, receiving extra compensation for their extended time.

The 19 participants who accepted were then asked to sign an addendum and promptly restarted the study once the new software update was available; the remaining 22 who declined were compensated upon study completion and had their data discarded from the study. All of these issues were properly reported to the UF IRB and approval was received to restart the study. Lastly, after study completion, we noticed that seven participants had technical issues with the extractor; we therefore discarded them from our sample. These issues, coupled with participant attrition, decreased our sample from 63 to 31 participants.

5 COMPUTER USAGE PROFILE EXTRACTION

In this section, we describe the design and implementation of the MS Windows 10 profile extractor our team developed to collect computer usage data from the study participants. We first considered using existing user-level and system event extractor tools (e.g., [24], [25], [26], [27]) in a standalone fashion, but decided against these options either because (1) the tool was intended for diagnostic tasks, thus incurring prohibitive performance overheads for continuous usage, such as required for our user study [25]; (2) the tool required nodes where data was collected to be on the same network, which would make our user study infeasible [24]; or (3) the tool did not provide fine-grained process activities, such creation or termination of process/thread, the suspension/resumption of processes/threads, process memory access [26], [27], which we deemed crucial in our design for the construction of computer usage-based profiles.

In our extractor, there are two main components installed on the users' systems: one for collecting data and one for uploading the data. The data collection component was based on a Sysmon-based logger to collect system level events, and a user-level application to collect keyboard and mouse related events. To collect system-level events (process creation and termination, network connection), we developed a logger based on Sysmon [27], a suite of tools to debug, manage, diagnose Windows systems. To collect mouse and keyboard events, we developed a Windows application to collect three types of events: mouse clicks, keyboard clicks, and process information, complementing information collected by the Sysmon-based logger. Whenever a mouse click is made on any application, the timestamp is recorded and associated with the application where the click occurred. After the initial click, if the user continues clicking on the same application, no further entries for that application are made over the next five minutes. However, during this 5-minute window, if the user switches to another application and makes a mouse click, that click timestamp is indeed recorded (see log entry examples below). Timestamps for keyboard events are recorded using this same method.

- 1288|C:/Program Files/Google/Chrome/Application/chrome.exe|132162206271021146|
- 8056|C:/Program Files/AVAST/Application/AvastBrowser.exe|132162207043754581|

The motivations behind this 5-minute window were two-fold. First, we wanted to capture the user's profile of software usage as reflected by user activity in windows of five minutes. There is a tradeoff in deciding the window size: the shorter, the more accurate the profile is, the longer,

the better the performance of the capturing module and less interference with user activities. Our design relies on windows of 5-minute of software activity as a proxy for software usage on that window. We considered three types of software activity: system-level events (e.g., process creation and termination, network connection) and mouse and keystroke events and the occurrence of either of them was considered software activity on the 5-minute window. Second, the mouse and keyboard activity added diversity to the type of data collected during the 5-minute window, which helps enriching the uniqueness of the user profiles. Thus, although the window size brings tradeoffs and limitations, it captures whether there as user software activity on a given interval.

This logging application also records complementary process information. It records the amount of time the process stays in user vs. kernel mode in one-minute granularities. The second component installed in the user's computers is an uploader, which is a python script that runs every 5 minutes. All logs captured by the extractor module are securely transferred to our lab server.

6 DATA ANALYSIS

This section goes over our data pre-processing steps and analyzes, which involved a diverse set of time series analysis methods and machine learning (ML) models.

6.1 Data Pre-processing

We pre-processed the raw data of each participant to generate an $N \times 6$ matrix where each line represented one minute of computer activity on a given study day (the number of lines therefore varied among users according to their computer usage). The columns contained: (i) a timestamp, (ii) a list of all active processes, (iii) a list of all domains accessed, (iv) the number of clicks associated with the timestamp, (v) the number of keystrokes associated with the timestamp, and (vi) an indicator (binary) of background processes activity that was detected via the occurrence of network traffic coinciding with lack of keyboard/mouse events, indicating a network activity that was generated by the process independently and without the interaction of the user (e.g., automatic updates running without the user's knowledge, or the user listening to music on YouTube without interacting with the browser for a long time period).

6.2 Profile Temporal Consistency Analysis (RQ1)

We aimed to investigate whether the profiles were *consistent over time* or *periodic* (RQ1), that is, whether study participants repeated computer usage habits periodically. Towards this goal, from the pre-processed data of each user, we extracted a time series of the number of active minutes (y -axis) per hour within the 8-week study period (x -axis) in two conditions: (i) considering the entire activity exhibited by the user, which included background process activity, and (ii) discarding background process activity. We included the second condition because the portion of background process activity generated without user knowledge (e.g., automatic software updates) may add a periodic component to the profile that is not related to the user behavior. In

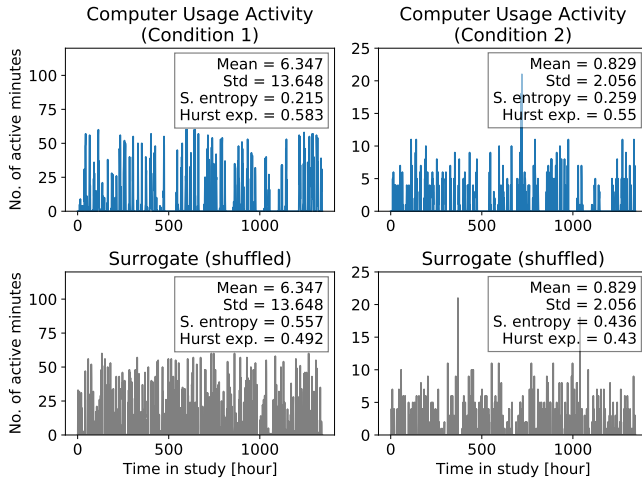


Fig. 1: Time series of computer usage data and its respective surrogate counterpart for User 1 in Conditions 1 (left) and 2 (right).

other words, each of the 31 users was represented by two time series of 1,344 data points each (56 study days \times 24 hours/day) ranging from 0 to 60.

In a nutshell, our analysis was comprised of two steps. First, we used quantitative methods to investigate whether the time series describing the profiles were periodic (i.e., structured, correlated in time) or random/stochastic (i.e., unstructured, uncorrelated in time). Next, for all time series differing from unstructured data, we used standard techniques to check for periodicity and identify the period.

In the first step of our analysis, we used the surrogate data testing method [28] to check for stochastic behavior in our time series. This method consists of comparing a particular nonlinear measure obtained from the time series with the distribution of the same measure obtained from a set of constructed time series—also known as *surrogates*—with the same statistical properties (e.g., mean, variance) but completely uncorrelated in time. If the result from the time series deviates significantly from the surrogate distribution, one can conclude that the time series is unlikely to be purely stochastic and therefore possesses some level of correlation in time. We chose two well-known nonlinear measures able to capture temporal correlations in time series data:

- *Sample entropy* corresponds to the negative natural logarithm of the conditional probability wherein sequences similar for m points remain similar after adding one more data point ($m + 1$) within a tolerance threshold r [29]. The more unstructured (random) the time series, the larger the sample entropy.
- *Hurst exponent* is a measure of long-term memory (or long-range correlations) of a time series, ranging between 0 and 1. A Hurst exponent of 0.5 suggests a stochastic behavior. Results lower than 0.5 suggest an anti-persistent behavior (a high value is followed by a low value and vice-versa) while results higher than 0.5 suggest a trending behavior (a high value is followed by a higher one). The closer the Hurst exponent is to 0 or 1, the stronger the anti-persistent or trending behavior of the time series, respectively.

We created 100 unique surrogates for each time series by shuffling its data points, thus destroying any existing temporal correlation—which we measured via sample entropy and Hurst exponent—while preserving the statistical properties, as illustrated in Figure 1. We computed the sample entropy and Hurst exponent using the `nolds` Python library with default parameters ($m = 2$ and $r = 0.2$ in the former). Before calculating sample entropy, we normalized the time series and surrogates to zero mean and unit variance to remove any non-stationary components that may confound results [30]. We did not normalize the time series nor the surrogates to extract the Hurst exponent since we are interested in capturing any existing trends in the data. We computed both metrics 100 times for each time series and then for its 100 unique surrogates. Next, after checking the normality of the distribution of the results with the Shapiro-Wilk test ($\alpha = 0.05$), we compared the distribution of each metric between time series vs. surrogates using either the paired t-test or the Wilcoxon signed-rank test ($\alpha = 0.001$).

After verifying that our times series (profiles) differed from stochastic data, we attempted to identify their period using two popular signal processing techniques: the *periodogram* and the *autocorrelation function* [31]. The former is obtained through the discrete Fourier transform, a method for expressing a time series as a sum of periodic components, and contains an approximation of the power spectral density (PSD, i.e., the power over different frequency values) of that time series. As illustrated in Fig. 2, the periodogram of a stochastic time series contains many components spread over different frequency values, thus not clearly exhibiting a main component. Conversely, the periodogram of a structured time series exhibits clear peaks for the main frequency (0.01Hz in the given example) and its multiples (or harmonics). The frequency value containing the highest spectral power (aka peak frequency, f_p) is associated with the strongest periodic component of the signal through the relation $T = 1/f_p$. We therefore estimated the period of each of our time series as the inverse of the peak frequency in its periodogram [31].

The second technique we used to estimate the period of our time series and validate the results obtained with the periodogram was the autocorrelation. The autocorrelation measures the self-similarity of a given time series over different delay times or lags (τ), ranging from -1 (perfectly negative autocorrelation) to 1 (perfectly positive autocorrelation) [31]. As illustrated in Fig. 2, for a stochastic (unstructured) time series, the autocorrelation is nearly zero for all values of lags because the data points are uncorrelated in time. On the other hand, for a structured time series with period T ($T = 100$ hours in the given example), the autocorrelation exhibits peaks for $\tau = k.T$ where $k = \{1, 2, 3, \dots\}$, and its amplitude decreases as the lag increases. A common way to estimate the period of a given time series is through the examination of N consecutive peaks in the autocorrelation function. In our analysis, we examined whether the five first peaks of each time series autocorrelation matched the period found through the periodogram. In other words, we verified whether the autocorrelation exhibited peaks for $\tau = k.T$ where $k = \{1, 2, \dots, 5\}$.

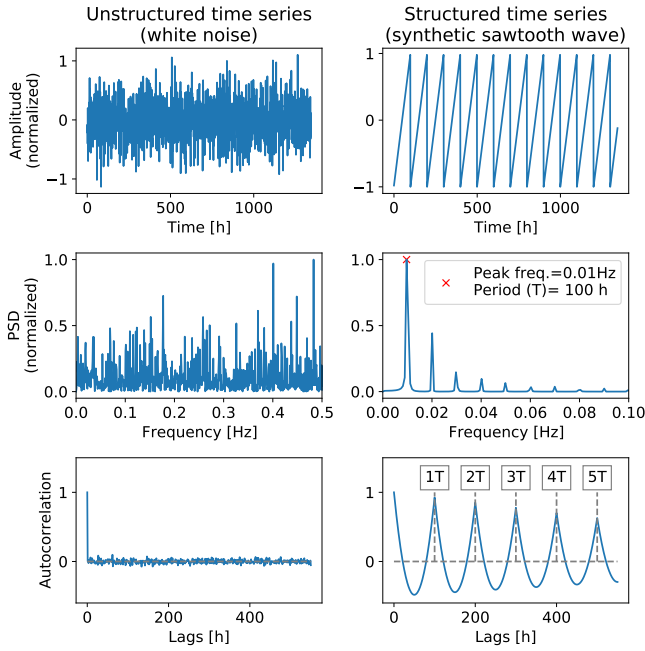


Fig. 2: Periodogram and autocorrelation for an unstructured (left panel) and structured (right panel) time series.

6.3 Machine Learning Analysis

Our ML experiments aimed to assess profile uniqueness (**RQ2**) and identify top features in distinguishing among profiles (**RQ3**).

6.3.1 Feature Extraction

An important aspect to be considered when analyzing profiles with ML models is the window of profile data needed for deciding whether the current behavior actually belongs to a certain user. We define this period as a *sliding window of time*. Since our logs were collected on a minute-basis, our window size t can assume any integer corresponding to the number of minutes (i.e., $t = 1, 2, 3, \dots$). In our strategy, we summed the number of clicks, keystrokes, and background traffic activity (all integer numbers), and concatenated the strings of the processes and domains used within consecutive t -minute windows, keeping the last timestamp of the window (Fig. 6 in Appendix B illustrates this process for a window size of 3 minutes). Therefore, every minute, a newly acquired window size of t minutes is analyzed by the ML classifier, which makes a prediction about whether such window of activity belongs to a certain user. A disadvantage of this technique is that larger t values are more susceptible to the cold-start problem (i.e., when there is a lack of initial information to start the recognition task, common in recommendation systems [32]). In our experiments, we opted to test several windows sizes to analyze the impact of small and large windows in predicting user identity based on the profiles. We tested $t = \{1, 2, 5, 10, 30, 60\}$ minutes.

Given that the numerical features (# of clicks, keystrokes, and background traffic activity) only need to be normalized before being used by a classifier, our feature extraction process focused on the textual attributes of the matrix (list of processes and domains). For this purpose, we used TF-IDF, a

statistical measure that evaluates how important a word (in our case, a process or domain name) is to a text in relation to a collection of texts [33]. Each text is represented by a sparse array that contains their TF-IDF values for each word in the vocabulary. Thus, each textual attribute from a window was transformed into its corresponding sparse array containing the TF-IDF values of each process and domain list (this process is also illustrated in Fig. 6, Appendix B). We envisioned that TF-IDF brings many advantages for applications in CA, given that (i) the more a word (e.g., process) appears in an instance, the larger its feature weight is; (ii) the less a word appears in all files, the higher its importance is to distinguish instances; and (iii) it can be periodically updated to accommodate new processes and websites using evolving feature sets or retraining its vocabulary when, for instance, a concept drift is detected [34], [35].

By combining the sliding window technique and TF-IDF features, the classifier receives as input an array that represents the window, containing the numerical features (sum of number of clicks, keystrokes, and background traffic activity) and the TF-IDF values for the processes and domains in the window, all normalized using maximum absolute scaler [36]. Thus, the textual attributes are used as a “document” that represents the user for a given moment.

6.3.2 Profile Uniqueness Analysis (RQ2)

The goal of the first set of machine learning experiments was to examine whether the profiles are unique (**RQ2**). For this purpose, we considered two types of ML models: (i) *offline*, where the classifier is trained only once using an initial portion of the data and then performs predictions for the remaining data without retraining; and (ii) *online*, where the classifier is periodically updated with more recent data. Offline classifiers are more popular in the ML field, mainly in stationary distribution problems where data does not change over time (e.g., object recognition). On the other hand, online models are conceptually more suitable to handle non-stationary distribution problems where data change over time [37]. Since we do not know which of these scenarios best describe our profiles. For comprehensiveness, we considered both types of classifiers in our analyses.

In both cases (offline and online learning), we conducted *binary* and *one-class classification* experiments. In the former, multi-class classifiers were used as binary discriminant functions, using the data from a given class (user) as positive, and the data from all the other classes (the remaining users) as negative [38]. In the latter, one predictive model was created for *each* user based on their data only (i.e., regardless of other users’ data) [39]. Though popular in the ML field, multi-class classifiers were left out of our experiments due to their lack of feasibility for profile-based CA. This is because, in the multi-class setting, a single predictive model is created based on the data of *all* users and then used to distinguish them among each other, thus requiring retraining every time a new user is added or removed from the group. Considering all variations of classifiers, parameters, number of runs, and window sizes, we trained and tested 20,460 models in our experiments.

Offline Classification. For the offline experiments, we chose four multi-class classifiers used in the binary classification

(Random Forest, Stochastic Gradient Descent (SGD), Multi-Layer Perceptron (MLP) and LinearSVC [36]) and two one-class (Isolation Forest, and One-Class SVM with both RBF and linear kernel). For the binary classifiers, we used the default parameters in Scikit Learn [36] for Random Forest (100 estimators), Stochastic Gradient Descent (hinge loss function), Multi-Layer Perceptron (100 hidden layers and relu activation function), and LinearSVC (Support Vector Machine linear kernel) for the SVM classifier. To evaluate these models, we split the data (ordered by timestamps) in two sets: the first seven days of data (of each user) were used for training and the remaining for testing. For the binary models, we created 31 binary models (one for each user, where the task is to detect if the current behavior window belongs to a given user) with each multi-class classifier cited before. To evaluate the binary models we used different negative users in the training and test sets, representing a typical continuous authentication system. For the one-class models, we also adopted the default settings in Scikit Learn [36] (Isolation Forest with 100 estimators and two one-class SVM, the first with RBF kernel and the second with linear kernel). For each user, we created an outlier set containing the instances of all the other 30 users, excluding the first seven days—that is, the same instances taken as test set in the binary experiments, for a fairer comparison among one-class and binary results.

Online Classification. For the online experiments, we selected four classifiers that support online learning: three multi-class in the binary classification (Adaptive Random Forest; Stochastic Gradient Descent, SGD; and Perceptron [36]) and one one-class (Half-Space Trees [40]). We leveraged the default parameters in Scikit Multiflow [41] and River [42] for both Adaptive Random Forest and Half-Space Trees [40] (which was the only online one-class classifier available at the time of this writing), and the default parameters in Scikit Learn for SGD and Perceptron. The same training and test sets from the offline experiments were used. The training set was used to train the initial model and the test set to create a data stream, which was considered in a test-then-train evaluation, where each sample is tested by the model (generating a prediction used to compute the F-score) and then used to update it [37].

Evaluation. As per the evaluation metrics, we included recall, precision, and F-score (the harmonic mean of precision and recall) in both offline and online classification experiments. These metrics provide a more realistic measure of a model's performance for unbalanced datasets than more popular metrics, such as accuracy [39]. For each classifier, we obtained the recall, precision, and F-score values per user using the average of ten runs with different random states (for the classifiers and users in the training and test set in the binary classification experiments) [43]. The only exception was the Adaptive Random Forest, in which we calculated the results for a single run due to time constraints (as it exhibited an excessively high training time in our experiments). We then computed the average recall, precision, and F-score results among the 31 users for each classifier, along with the respective 95% confidence intervals.

6.3.3 Top Features Analysis (RQ3)

To analyze the most important features in identifying each user, we first divided their data in weeks to observe which are the most important over time. We trained a binary Random Forest classifier for each week of each user with a time window of 10 minutes (the optimal value found in RQ1), resulting in 248 models trained. As we found that most of the profiles repeat on a daily basis (see Sec. 7), we performed weekly training to consider several repetitions of computer usage habits to identify the most relevant features. Moreover, we chose the Random Forest for two reasons: it provides the importance of each feature in the classification process, and it outperformed the other tested binary models in both offline and online settings. We then used the Gini importance property from scikit-learn's Random Forest implementation [36] to explain our models' predictions for each instance of the corresponding week and user. Finally, we computed these values (the higher the value, the more important the feature is) for each feature and obtained their average, where we collected the top 10 for each user.

7 EXPERIMENTAL RESULTS

Profile Temporal Consistency (RQ1). In our surrogate data testing analysis, the sample entropy results obtained from the time series representing the computer usage profiles were statistically higher ($p < .001$) than the results obtained from the surrogates for all 31 users in the two assessed conditions (with and without background activity). The difference in the sample entropy results between time series vs. surrogates was more noticeable in the first condition (0.222 ± 0 vs. 0.714 ± 0.025) compared to the second (0.277 ± 0 vs. 0.499 ± 0.021). Regarding the Hurst exponent results, statistically different ($p < .001$) values for time series vs. surrogates were observed for 29 profiles (all users except User 5 and User 22) in the first condition (with background activity), and for all 31 profiles in the second condition (without background activity).

All 31 profiles differed from unstructured data in terms of sample entropy in the two assessed conditions ($p < .001$). In terms of Hurst exponent results, the number of profiles that differed from unstructured data was 29 and 31 in the first and second conditions.

We identified a period for 28 profiles (90.3% of our sample) in the first condition (with background activity) and for 29 profiles (93.5% of our sample) in the second condition (without background activity). In both conditions, 26 profiles exhibited a period equal to 24 hours, as exemplified in Fig. 3 for User 9. In other words, 83.9% of the study participants repeated computer usage patterns daily. The periodogram and autocorrelation for all users and conditions are exhibited in Appendix D (Fig. 7).

Figure 3 contrasts results between the periodogram and autocorrelation for Users 2, 19, and 30 in the first condition, and Users 2 and 13 in the second condition, preventing us from identifying a period in these cases. User 2's periodogram and autocorrelation were similar to the expected for unstructured data, especially in the second condition.

For User 19, the periodogram exhibited a period of 192 hrs while the autocorrelation suggested 750 hrs. In the remaining cases, the autocorrelation did not exhibit peaks consistent with the period found with the periodogram.

Twenty-five profiles exhibited results consistent between the two conditions, with the period being equal to 24 hours in all cases. In the cases where the results changed between conditions (Users 5 and 8), the period was found to be lower in the second condition (i.e., after removing background activity), as illustrated in Fig. 3 for User 5.

A period was found for 90% of the profiles in Condition 1, and for 94% in Condition 2. 84% of the profiles exhibited a period of 24 hours.

Profile Uniqueness (RQ2). Figure 4 shows the top results obtained to distinguish among the profiles. The complete results are exhibited in Appendix E (Tables 3, 4, and 5). The online one-class classifier yielded the poorest results: 0% precision, recall, and F-score; the Half-Space Trees model, the only model available, works better when anomalous data are rare, which is the opposite of our scenario [40].

In summary, the best results were reached with the online binary models (top F-score of 99.90%), followed by the offline binary models (top F-score of 95.00%). We observed that evaluation metrics improved with the increase of the window size. Most of the binary classifiers (our top performers) achieved optimal results for $t = 10$ min given that the increase of t to 30 min and 60 min did not cause the evaluation metrics to improve substantially.

Profile Top Features (RQ3). For 29/31 users, the top 10 features were all web domains. After unifying the top 10 features from each user and removing duplicates, we obtained a set of 186 unique features in which 177 were *domains* and only five were *processes*. The four remaining features refer to *background traffic* (which only appeared for two users), *number of mouse clicks*, and *number of keystrokes* (both appearing for a single user). From these 186 top features, 45 were observed for two or more users.

8 DISCUSSION

Computer Usage Profiles Consistency. Our analysis of the temporal consistency of the profiles (RQ1) revealed that the majority of the users enrolled in our study did exhibit computer usage patterns consistent over time in the two assessed conditions (with and without background process activity). Interestingly, most of the users repeated computer usage on a daily basis over the 8-week study period. As the profiles also repeated for the integer multiples of their main period (as observed in the autocorrelation results), it is plausible to assume that such profiles also repeated every j days where $j = \{1, 2, \dots, 56\}$, every k weeks where $k = \{1, 2, \dots, 8\}$, and every l months where $l = \{1, 2\}$, within the study period. In other words, the profiles repeated not only on a daily basis, but also on a weekly and monthly basis, for example.

We acknowledge that these results are likely to be sensitive to changes in the user's routine that alter the way they use the computer (e.g., change of role, vacation, and travel in

corporate environments), which might be the reason for the lack of periodicity observed for User 2. We therefore assume that most of our users did not have significant changes in their main occupation/tasks while in study.

Our analysis involving the surrogate data revealed that all profiles differed from completely unstructured data in the two tested conditions, i.e., they are unlikely to be purely stochastic. However, our periodogram and autocorrelation results (Fig. 7) were not identical to those observed for purely structured data (Fig. 2). Our results look similar to the expected for structured time series contaminated with random noise. In both conditions, most profiles exhibited a periodogram and autocorrelation more similar to the expected for structured data (e.g., User 1), suggesting lower presence of noise. On the other hand, results more similar to the expected for unstructured data were also observed for some users (e.g., User 2), suggesting higher contamination by noise (Fig. 5). In other words, users exhibited different levels of correlation in time in their use patterns.

These findings suggest that profiles can be described as a combination of two components, one *deterministic* (most likely nonlinear) and one *stochastic*, with the latter being stronger for some users than others. In other words, it seems reasonable to infer that profiles are consistent over time, but yet subject to random factors that decrease their temporal correlations and make them less structured.

Another factor that may have caused random drifts in the computer usage behaviors of our users is the current COVID-19 pandemic (our data collection happened during the months of the pandemic). The pandemic has greatly altered how users in general work, learn, socialize, and entertain themselves using their personal devices [44], likely causing them to use the same computer for personal and work tasks. Several activities of daily life started to take place virtually due to safety concerns, such as grocery shopping, bank services, classes in general (e.g., schools, universities, gyms), watching movies or concerts etc., and many of them may not happen on a regular basis (e.g., someone might buy groceries twice a week for two weeks and then spend three weeks without buying anything). This has the potential to strengthen the stochastic component in the profiles.

Importantly, when moving from the first condition (with background activity) to the second one (without background activity), we noticed a decrease in the peaks of the autocorrelation function (Fig. 7). We also observed an increase in the average sample entropy result (from 0.222 ± 0.114 to 0.277 ± 0.124), as well as a decrease in the average Hurst exponent result towards 0.5 (from 0.608 ± 0.104 to 0.564 ± 0.084). All these results indicate that the removal of the background activity caused the profiles to be less correlated in time. This finding is aligned with our hypothesis that the background activity contains periodic events that can strengthen the deterministic component in the profiles. However, some of these events may be produced by the OS without the knowledge of the user and according to a predefined schedule (e.g., automatic software update), which might make them easy to predict/mimic. Thus, it might be possible to increase the robustness of CA tools by leaving background process activity out of the computer usage profiles.

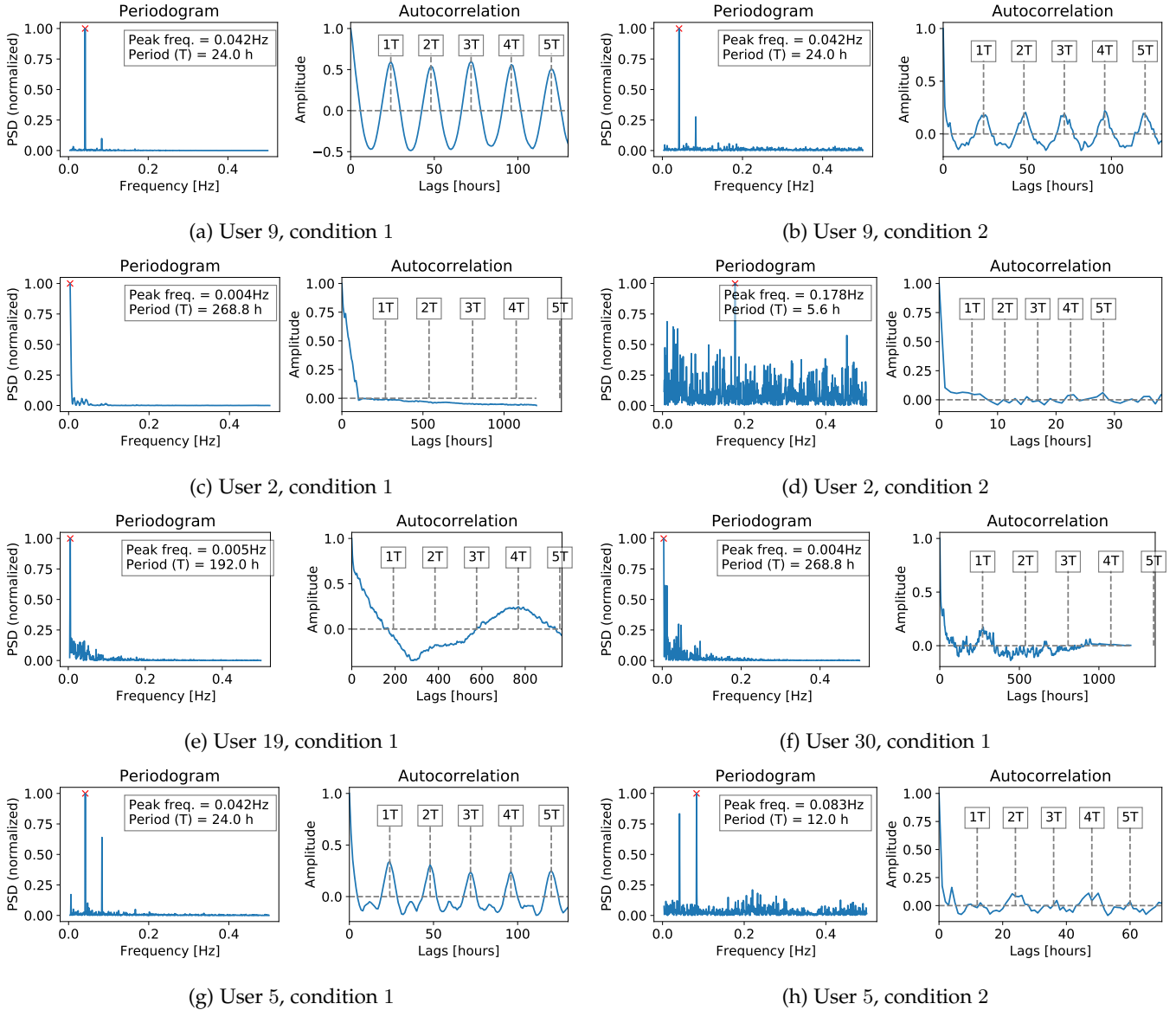


Fig. 3: Periodogram and autocorrelation results for different users and conditions.

The profiles were mostly consistent, with users repeating habits on a daily basis, but subject to random factors decreasing their temporal consistency.

Profiles Uniqueness. Our results from **RQ2** partially supported the hypothesis that profiles comprised of process-, network-, mouse-, and keystrokes-related events can uniquely characterize computer users. While most of the tested binary classifiers yielded promising results, the one-class models achieved poor results (in both offline and on-line settings). In other words, our profiles can be considered unique depending on the learning model used. Importantly, we trained our classifiers using one week of data, which potentially contained seven repetitions of computer usage habits given that we found most of the profiles to repeat on a daily basis (**RQ1**). This may explain the promising results achieved with the binary classifiers. Using less training data can allow the CA solution to be implemented in a more

timely manner, but may also affect its performance since less usage patterns would be available for the learning model.

Our top feature analysis (**RQ2**) revealed that the set of domains accessed by the users was significantly more relevant to distinguish profiles, than other process- and keystrokes-related features. This finding suggests that collecting only network-related events can potentially lead to unique profiles. However, it is important to remark that our data was collected with users who experienced no restrictions on network activity, contrary to some corporate environment that restrict network activity, such as social media sites. Restricting the list of domains that users can access will potentially decrease profile uniqueness and impose challenges to profile-based CA systems. In this case, enriching a profile with process-, mouse-, and/or keystroke-related events may help to increase profile uniqueness and improve user recognition with CA.

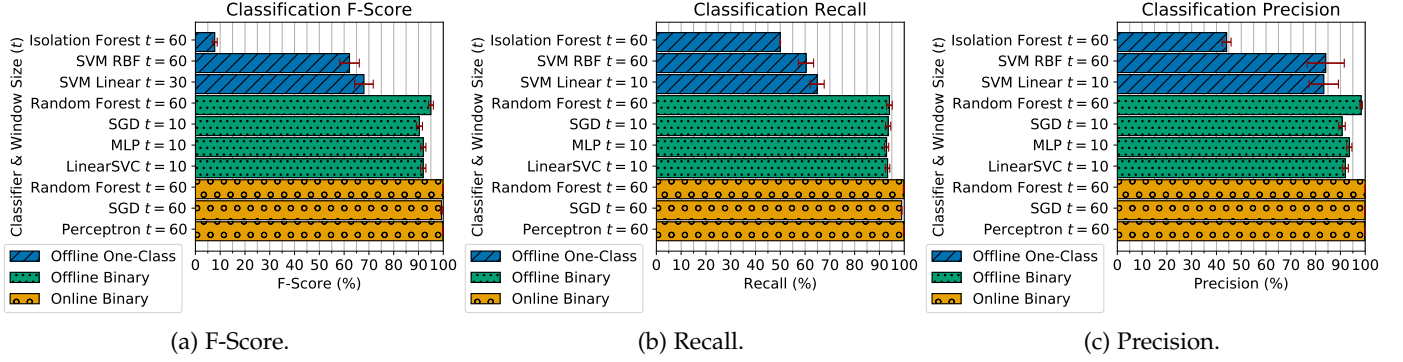


Fig. 4: Top average F-score (a), recall (b), and precision (c) values in distinguishing among the computer usage profiles. The error bars denote the 95% confidence intervals.

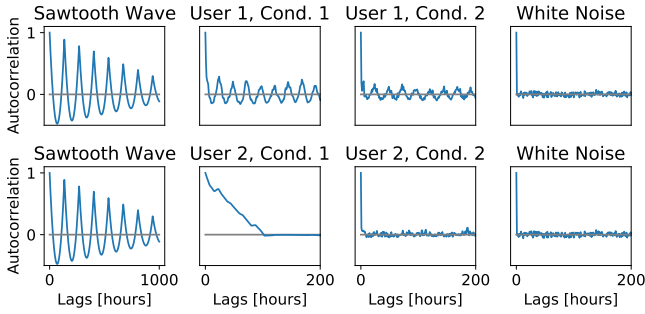


Fig. 5: Comparison between the autocorrelation for purely structured data (sawtooth wave) and purely stochastic data (white noise); results from User 1 (top) and User 2 (bottom).

Computer usage profiles can uniquely characterize computer users when analyzed with binary classifiers. Network-related events constituted the most important feature to construct unique profiles.

Challenges and Opportunities of Using Computer Usage Profiling for Applications in CA. Regarding the feasibility of leveraging computer usage profiles for CA applications, our data suggests that binary models are better suited for profile-based CA tools than one-class models; while both allow the training on a per-user basis, which is advantageous for corporations due to employee turnover/hiring, the binary models were able to identify the profiles more accurately. Moreover, our results suggest that online binary models should be preferred over offline ones due to better performance and robustness to temporal changes in profiles.

One limitation of leveraging profiles for CA is that a certain amount of data is necessary to start recognizing users (a.k.a. cold-start problem). We found that a time window of 10 minutes was the optimal trade-off to recognize our dataset of profiles with the binary models (our top performers), but this optimal value might change for other datasets. This may be a challenge for profile-based CA tools given that the longer this optimal time window, the more susceptible the system is to a cold-start problem, causing the device to be initially vulnerable for a longer period of time. Moreover, the longer it takes for the system to recognize

an anomaly, the less value the CA system might have for corporate security, because the anomaly could correspond to a malicious activity whose effects persist after detection.

8.1 Limitations and Future Work

One limitation of our study lies in our sample, which is small ($N = 31$) and comprised mainly of young adults from a large university (i.e., not representative of all possible environments). Although a large university of nearly 50K students may be somewhat similar to an organization, corporate users likely present different computer usage profiles compared to university students. These factors thus limit the generalizability of our findings. Future studies with more diverse and representative samples are warranted.

Importantly, we collected our data during the COVID-19 pandemic, which might have affected user behavior and, consequently, influenced our conclusions towards profiles' uniqueness and consistency. The yet unforeseeable changes to how our society may shift after the pandemic is unresolved and how this may impact the user profiling analyses presented in this paper will require investigations from future work (e.g., will there be more sparse personal computer usage as users return to work in offices and classrooms?).

Another limitation of our study concerns the integrity of the systems in which computer usage data was recorded: we assumed that the users' systems were not compromised by malware during the study period. The presence of malware could have adversely affected the learning models, as malicious behaviors would be considered part of the usage profile. In addition, although all participants confirmed that they did not share their computers while in study, some self-reported responses might not be accurate.

Regarding our analysis of profiles' temporal consistency (RQ1), we focused on finding a period for each profile using well known techniques. Future research is needed to investigate whether profiles indeed possess a deterministic component combined with a stochastic one, as we concluded based on our results. Given the evidence of nonlinear dynamics in the profiles that we found via surrogate data testing, future studies might benefit from using other nonlinear metrics/techniques to better understand the behavior of computer usage patterns. For instance, one may try to reconstruct the phase space and investigate its properties.

In our classification experiments (RQ2 and RQ3), we employed a time window. This may constitute a limitation

because, in some cases, when the behavior is about to change, the classifier may have difficulty in detecting the shift because the window will gradually change with time, especially when its size is large (60 minutes, for instance). Future work addressing this issue is needed to improve classification performance. Moreover, we only considered traditional metrics (precision, recall, and F-score) to evaluate the performance of the tested classifiers. Future studies focusing on CA systems are advised to consider other evaluation metrics that are better suited for this case, such as receiver operating characteristics (ROC) curve [45].

9 CONCLUSIONS

We conducted an ecologically-valid user study with 31 computer users to systematically investigate whether computer usage profiles comprised of process, mouse, keystrokes, and network events are consistent over time and unique. Though we found evidence of temporal consistency for most of the profiles within the study period—with most of them reoccurring every 24 hours—our results suggest that these profiles experienced variations over time, due to factors such as days off. This suggests that online ML models (which allow for periodical retraining) might be better suited for a profile-based CA tool than offline models (where training occurs only once). In our comprehensive set of experiments, binary classifiers (online and offline) were able to accurately recognize the profiles, indicating that computer usage profiling can uniquely characterize computer users. Network domains accessed by users were more relevant in recognizing them than the keyboard and mouse activity.

ACKNOWLEDGMENTS

This work was supported by NSF award 1815557. This material is based upon work supported by (while serving at) the National Science Foundation. We would like to acknowledge Arpitha Nagaraj Hegde for contributing to the development of our extractor module, Shlok Gilda for his advice on how to encode the computer usage data into time series, and Natalie Ebner for discussions related to an earlier version of our planned user study.

REFERENCES

- [1] D. Kim, P. Dunphy, P. Briggs, J. Hook, J. W. Nicholson, J. Nicholson, and P. Olivier, "Multi-touch authentication on tabletops," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 1093–1102.
- [2] J. Yang, Y. Qiao, X. Zhang, H. He, F. Liu, and G. Cheng, "Characterizing user behavior in mobile internet," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 1, pp. 95–106, March 2015. [Online]. Available: <https://doi.org/10.1109/TETC.2014.2381512>
- [3] C. Tossell, P. Kortum, A. Rahmati, C. Shepard, and L. Zhong, "Characterizing web use on smartphones," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 2769–2778. [Online]. Available: <http://doi.acm.org/10.1145/2207676.2208676>
- [4] L. Fridman, S. Weber, R. Greenstadt, and M. Kam, "Active authentication on mobile devices via stylometry, application usage, web browsing, and gps location," *IEEE Systems Journal*, vol. 11, no. 2, pp. 513–521, June 2017.
- [5] J. Payne, M. Fenner, and J. Kauffman, "System event monitoring for active authentication," *IT Professional*, vol. 15, no. 4, pp. 34–37, July 2013. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/MITP.2013.57>
- [6] J. Chuang, H. Nguyen, C. Wang, and B. Johnson, "I think, therefore i am: Usability and security of authentication using brainwaves," in *International conference on financial cryptography and data security*. Springer, 2013, pp. 1–16. [Online]. Available: https://doi.org/10.1007/978-3-642-41320-9_1
- [7] P. Emami-Naeini, T. Francisco, T. Kohno, and F. Roesner, "Understanding privacy attitudes and concerns towards remote communications during the covid-19 pandemic," in *Seventeenth Symposium on Usable Privacy and Security (SOUPS)*, 2021, pp. 695–714.
- [8] FindLaw, "Email privacy concerns," Jun. 2016. [Online]. Available: <https://www.findlaw.com/consumer/online-scams/email-privacy-concerns.html>
- [9] H. Shi and Y. Li, "Discovering periodic patterns for large scale mobile traffic data: method and applications," *IEEE Transactions on Mobile Computing*, vol. 17, no. 10, pp. 2266–2278, 2018. [Online]. Available: <https://doi.org/10.1109/TMC.2018.2799945>
- [10] U. Mahbub, J. Komulainen, D. Ferreira, and R. Chellappa, "Continuous Authentication of Smartphones Based on Application Usage," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 1, no. 3, pp. 165–180, Jul. 2019.
- [11] A. A. Ahmed and I. Traore, "Biometric recognition based on free-text keystroke dynamics," *IEEE Transactions on Cybernetics*, vol. 44, no. 4, pp. 458–472, April 2014.
- [12] P. Kang and S. Cho, "Keystroke dynamics-based user authentication using long and free text strings from various input devices," *Information Sciences*, vol. 308, pp. 72 – 93, 2015.
- [13] B. Sayed, I. Traore, I. Woungang, and M. S. Obaidat, "Biometric authentication using mouse gesture dynamics," *IEEE Systems Journal*, vol. 7, no. 2, pp. 262–274, 2013. [Online]. Available: <https://doi.org/10.1109/JSYST.2012.2221932>
- [14] Y. Sun, H. Ceker, and S. Upadhyaya, "Shared keystroke dataset for continuous authentication," in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec. 2016, pp. 1–6. [Online]. Available: <https://doi.org/10.1109/WIFS.2016.7823894>
- [15] A. Lazarevic, V. Kumar, and J. Srivastava, "Intrusion detection: A survey," in *Managing Cyber Threats*. Springer, 2005, pp. 19–78.
- [16] C. Murphy, J. Huang, D. Hou, and S. Schuckers, "Shared dataset on natural human-computer interaction to support continuous authentication research," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, Oct. 2017, pp. 525–530. [Online]. Available: <https://doi.org/10.1109/BTAS.2017.8272738>
- [17] U. Mahbub, S. Sarkar, V. M. Patel, and R. Chellappa, "Active User Authentication for Smartphones: A Challenge Data Set and Benchmark Results," *IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2016. [Online]. Available: <https://doi.org/10.1109/BTAS.2016.7791155>
- [18] D. Ferreira, V. Kostakos, A. R. Beresford, J. Lindqvist, and A. K. Dey, "Securacy: an empirical investigation of android applications' network usage, privacy and security," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2015, pp. 1–11. [Online]. Available: <http://dx.doi.org/10.1145/2766498.2766506>
- [19] A. U. López, F. Mateo, J. Navío-Marco, J. M. Martínez-Martínez, J. Gómez-Sanchís, J. Vila-Francés, and A. J. Serrano-López, "Analysis of computer user behavior, security incidents and fraud using self-organizing maps," *Computers & Security*, vol. 83, pp. 38–51, 2019. [Online]. Available: <https://doi.org/10.1016/j.cose.2019.01.009>
- [20] B. Ayotte, M. K. Banavar, D. Hou, and S. Schuckers, "Study of intra-and inter-user variance in password keystroke dynamics," in *Proceedings of the 7th International Conference on Information Systems Security and Privacy (ICISSP)*, 2021, pp. 467–474. [Online]. Available: <https://doi.org/10.5220/0010235504670474>
- [21] H. Zhang, V. M. Patel, M. Fathy, and R. Chellappa, "Touch gesture-based active user authentication using dictionaries," in *2015 IEEE Winter Conference on Applications of Computer Vision (WACV)*, vol. 00, Jan. 2015, pp. 207–214. [Online]. Available: <https://doi.org/10.1109/WACV.2015.35>
- [22] B. Ribeiro, H. Pereira, R. Almeida, A. Ferreira, L. Martins, C. Quaresma, and P. Vieira, "Optimization of sitting posture classification based on user identification," in *2015 IEEE 4th Portuguese Meeting on Bioengineering (ENBENG)*. IEEE, 2015, pp.

- 1–6. [Online]. Available: <https://doi.org/10.1109/ENBENG.2015.7088853>
- [23] J. Huang, D. Hou, and S. Schuckers, "A practical evaluation of free-text keystroke dynamics," in *2017 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, Feb 2017, pp. 1–8.
- [24] T. Inc, "Tanium," Feb. 2020. [Online]. Available: <https://www.tanium.com/products/core-platform>
- [25] M. Russinovich, "Process monitor - windows sysinternals," Dec. 2019. [Online]. Available: <https://docs.microsoft.com/en-us/sysinternals/downloads/procmon>
- [26] Microsoft, "Event tracing for windows (etw)," Apr. 2017. [Online]. Available: <https://docs.microsoft.com/en-us/windows-hardware/drivers/devtest/event-tracing-for-windows--etw->
- [27] M. Russinovich and T. Garnier, "Sysmon - windows sysinternals," Dec. 2019. [Online]. Available: <https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon>
- [28] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. D. Farmer, "Testing for nonlinearity in time series: the method of surrogate data," *Physica D: Nonlinear Phenomena*, vol. 58, no. 1-4, pp. 77–94, 1992. [Online]. Available: [https://doi.org/10.1016/0167-2789\(92\)90102-S](https://doi.org/10.1016/0167-2789(92)90102-S)
- [29] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 278, no. 6, pp. H2039–H2049, 2000.
- [30] M. Costa, A. Priplata, L. Lipsitz, Z. Wu, N. Huang, A. L. Goldberger, and C.-K. Peng, "Noise and poise: enhancement of postural complexity in the elderly with a stochastic-resonance-based therapy," *Europhysics Letters*, vol. 77, no. 6, p. 68008, 2007.
- [31] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [32] J. Yuan, W. Shalaby, M. Korayem, D. Lin, K. AlJadda, and J. Luo, "Solving cold-start problem in large-scale recommendation engines: A deep learning approach," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1901–1910.
- [33] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. USA: Cambridge University Press, 2008.
- [34] F. Ceschin, F. Pinage, M. Castilho, D. Menotti, L. S. Oliveira, and A. Gregio, "The need for speed: An analysis of brazilian malware classifiers," *IEEE Security Privacy*, vol. 16, no. 6, pp. 31–41, Nov 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8636415>
- [35] K. Xu, Y. Li, R. Deng, K. Chen, and J. Xu, "Droidelvolver: Self-evolving android malware detection system," in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2019, pp. 47–62.
- [36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [37] A. Bifet, R. Gavaldà, G. Holmes, and B. Pfahringer, *Machine Learning for Data Streams with Practical Examples in MOA*. MIT Press, 2018.
- [38] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.
- [39] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
- [40] S. C. Tan, K. M. Ting, and T. F. Liu, "Fast anomaly detection for streaming data," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
- [41] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, "Scikit-multiflow: A multi-output streaming framework," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 2915–2914, 2018. [Online]. Available: <https://github.com/scikit-multiflow/scikit-multiflow>
- [42] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem *et al.*, "River: machine learning for streaming data in python," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 1–8, 2021.
- [43] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: <https://doi.org/10.1023/A:1010933404324>
- [44] E. Koeze and N. Popper, "The virus changed the way we internet," Apr. 7 2020. [Online]. Available: <https://www.nytimes.com/interactive/2020/04/07/technology/coronavirus-internet-use.html>
- [45] S. Sugrim, C. Liu, M. McLean, and J. Lindqvist, "Robust performance metrics for authentication systems," in *Network and Distributed Systems Security (NDSS) Symposium 2019*, 2019. [Online]. Available: <https://dx.doi.org/10.14722/ndss.2019.23351>

Luiz Giovanini is a Postdoc Researcher at the Department of Electrical and Computer Engineering (ECE) at the University of Florida (UF). His current research interests include machine learning for applications in the cyber security and biomedical fields.

Fabício Ceschin is a PhD candidate at the Department of Informatics at the Federal University of Parana, Brazil. His research interests include machine learning, adversarial machine learning, and data streams applied to cyber security. He has received support from the Google Research Awards for the Latin America program.

Mirela Silva is a PhD candidate at the Department of ECE at the UF. Her research interests include interdisciplinary computer privacy, focused heavily on the intersection of cyber deception and abuse through the lens of interindividual differences.

Aokun Chen is a Postdoctoral Researcher at the Department of Health Outcomes & Biomedical Informatics at UF. He received his Ph.D. degree from the ECE Department at UF. His current research interests include machine learning for bioinformatics and cybersecurity.

Sanjay Banda is an MSc. Student at the Department of Computer and Information Science and Engineering (CISE) at UF. His current research interests Distributed Systems and Content Delivery Networks.

Madison Lysaght received her Bachelor's degree from UF. Her research interests includes human computer interaction focused around design thinking for cyber deception.

Heng Qiao is a Ph.D. candidate at the Department of ECE at UF. His research interests include machine learning and anomaly detection.

Nikolaos Sapountzis is a Senior Software Engineer at the Network Function Virtualization team at Cisco. He received his PhD from the EURECOM with focus on 5G networks. His research interests include networks, optimization, and privacy and security.

Ramchandra Kulkarni is a Software Development Engineer with Query Processing team at Redshift, AWS. He received his MSc degree in Computer Science Department of CISE at UF.

Ruimin Sun is a Postdoc Researcher in Northeastern University. She received her Ph.D from University of Florida. Her research interests include cyber physical system security, machine learning model privacy, and malware detection.

Brandon Matthews is a Senior Researcher for the Electro-Optical Systems Laboratory at GTRI. He has more than 13 years of experience, largely focused on innovative modern electronic warfare applications.

Dapeng Oliver Wu is a professor at the Department of ECE at the UF. His research interests are in the areas of networking, communications, signal processing, computer vision, machine learning, smart grid, and information and network security. He has served as Editor in Chief of IEEE Transactions on Network Science and Engineering.

André Grégio is an Assistant Professor of the Department of Informatics at the UFPR, Brazil. He received his PhD in Computer Engineering from Unicamp, Brazil. His research interests include malware countermeasures and machine learning techniques applied to security data.

Daniela Oliveira is an Associate Professor in the Department of ECE at UF. Her current research interests are human factors in cyber security, phishing, and disinformation.

APPENDIX A

STUDY PARTICIPANTS' DEMOGRAPHICS

Table 1 summarizes the demographics of our 31 study participants.

TABLE 1: Summary of study participants' demographics.

Category	Metric	Total (N = 31)
Gender	Female	20 (64.52%)
	Male	10 (32.26%)
	Gender Variant/ Non-Conforming	1 (3.23%)
Age	18–25 years	16 (46.43%)
	26–35 years	12 (39.28%)
	36–45 years	2 (7.14%)
	≥46 years	2 (7.14%)
	Associate	3 (9.68%)
Highest Formal Degree	Bachelor's	11 (35.48%)
	Master's	0 (0.00%)
	PhD/Doctorate	5 (16.13%)
	Other	12 (38.71%)
	Single	7 (22.58%)
Marital Status	Married	11 (35.48%)
	Divorced	3 (9.68%)
	In a relationship	10 (32.26%)
Living Condition	Alone	5 (16.13%)
	With spouse/S.O.*	13 (41.94%)
	With child(ren)	3 (9.68%)
	Assisted living	0 (0.00%)
	Other	10 (32.26%)
Employment Status	Employed	17 (54.84%)
	Unemployed	14 (45.16%)
	Retired	0 (0.00%)
Yearly Household Income	<\$10,000	9 (29.03%)
	\$10,000 to <\$50,000	10 (32.26%)
	\$50,000 to <\$100,000	6 (19.35%)
	≥\$100,000	6 (19.35%)
Latino Ethnicity	Not Hispanic/Latino	23 (74.19%)
	Hispanic/Latino	8 (25.81%)
	English	24 (77.42%)
Primary Language	Portuguese	4 (12.90%)
	Sinhala	2 (6.45%)
	Chinese	1 (3.23%)

* S.O. = significant other

time series of computer usage profiles vs. their surrogate counterparts in the two assessed conditions: with and without background process activity.

APPENDIX D

PROFILES' TEMPORAL CONSISTENCY RESULTS

Figure 7 exhibits the periodogram and autocorrelation results for all 31 study participants in the two assessed conditions: with and without background process activity.

APPENDIX E

COMPLETE CLASSIFICATION RESULTS

Tables 3, 4, and 5 exhibit the classification results for all window sizes obtained with the offline binary models, offline one-class models, and online binary models, respectively.

APPENDIX B

FEATURE EXTRACTION DETAILS

Figure 6 illustrates the sliding window strategy used in our ML experiments for a window size of 3 minutes. As shown, we summed the number of clicks, keystrokes, and background traffic activity (all integer numbers), and concatenated the strings of the processes and domains used within consecutive t -minute windows ($t = 3$ in the given example), keeping the last timestamp of the window. For an original matrix containing N lines of active minutes, this process groups computer usage in t -minute windows.

APPENDIX C

SURROGATE DATA TESTING RESULTS

Table 2 exhibits the results of the statistical comparison for the sample entropy and Hurst exponent obtained from the

TABLE 2: Statistical comparison for the sample entropy and Hurst exponent, obtained by leveraging the time series of computer usage profiles against their surrogate counterparts. *Condition 1* denotes that background process activities were assessed, and *Condition 2* indicates that these background activities were *not* assessed. All users have been de-identified.

User	Sample Entropy									Hurst Exponent										
	Condition 1				Condition 2					Condition 1				Condition 2						
	Time Series		Surrogates		p	Time Series		Surrogates		p	Time Series		Surrogates		p	Time Series		Surrogates		p
μ	σ	μ	σ	μ		σ	μ	σ	μ		σ	μ	σ	μ		σ	μ	σ		
1	0.215	0.000	0.568	0.026	***	0.259	0.000	0.467	0.022	***	0.583	0.022	0.500	0.040	***	0.538	0.020	0.496	0.035	***
2	0.146	0.000	0.295	0.011	***	0.428	0.000	0.470	0.019	***	0.639	0.008	0.488	0.039	***	0.546	0.000	0.489	0.031	***
3	0.263	0.000	0.951	0.014	***	0.469	0.000	0.743	0.029	***	0.765	0.013	0.499	0.035	***	0.644	0.002	0.498	0.034	***
4	0.214	0.000	0.901	0.033	***	0.309	0.000	0.488	0.021	***	0.778	0.000	0.504	0.038	***	0.642	0.006	0.487	0.035	***
5	0.539	0.000	1.237	0.044	***	0.559	0.000	0.975	0.040	***	0.496	0.016	0.496	0.037	0.989	0.517	0.004	0.493	0.034	***
6	0.126	0.000	0.370	0.015	***	0.152	0.000	0.247	0.010	***	0.737	0.000	0.495	0.039	***	0.630	0.002	0.487	0.030	***
7	0.228	0.000	0.934	0.032	***	0.285	0.000	0.410	0.016	***	0.679	0.011	0.493	0.041	***	0.509	0.008	0.486	0.036	***
8	0.092	0.000	0.159	0.007	***	0.098	0.000	0.154	0.005	***	0.635	0.018	0.469	0.023	***	0.649	0.012	0.477	0.030	***
9	0.258	0.000	0.899	0.009	***	0.403	0.000	1.050	0.043	***	0.326	0.045	0.495	0.040	***	0.348	0.030	0.502	0.037	***
10	0.234	0.000	0.648	0.027	***	0.294	0.000	0.444	0.020	***	0.591	0.033	0.498	0.042	***	0.383	0.026	0.496	0.037	***
11	0.181	0.000	0.646	0.027	***	0.205	0.000	0.415	0.017	***	0.578	0.019	0.493	0.034	***	0.424	0.007	0.485	0.033	***
12	0.087	0.000	0.432	0.016	***	0.100	0.000	0.279	0.011	***	0.552	0.014	0.495	0.036	***	0.628	0.013	0.482	0.032	***
13	0.378	0.000	0.991	0.044	***	0.494	0.000	0.846	0.038	***	0.617	0.049	0.503	0.037	***	0.546	0.019	0.488	0.034	***
14	0.226	0.000	0.668	0.027	***	0.329	0.000	0.552	0.025	***	0.548	0.039	0.500	0.036	***	0.543	0.051	0.489	0.032	***
15	0.173	0.000	0.552	0.023	***	0.226	0.000	0.496	0.019	***	0.646	0.023	0.502	0.037	***	0.585	0.028	0.492	0.036	***
16	0.147	0.000	0.784	0.040	***	0.218	0.000	0.464	0.016	***	0.462	0.003	0.495	0.037	***	0.628	0.001	0.483	0.035	***
17	0.065	0.000	0.281	0.012	***	0.085	0.000	0.198	0.007	***	0.682	0.002	0.494	0.036	***	0.675	0.001	0.484	0.036	***
18	0.143	0.000	0.216	0.009	***	0.177	0.000	0.212	0.009	***	0.565	0.052	0.484	0.029	***	0.437	0.017	0.476	0.030	***
19	0.162	0.000	0.951	0.008	***	0.321	0.000	0.443	0.019	***	0.791	0.000	0.491	0.040	***	0.626	0.007	0.484	0.033	***
20	0.230	0.000	0.807	0.029	***	0.265	0.000	0.579	0.026	***	0.686	0.012	0.493	0.038	***	0.535	0.033	0.488	0.033	***
21	0.123	0.000	0.283	0.012	***	0.097	0.000	0.153	0.006	***	0.564	0.003	0.488	0.034	***	0.566	0.006	0.477	0.029	***
22	0.113	0.000	0.610	0.026	***	0.156	0.000	0.388	0.017	***	0.485	0.004	0.496	0.034	0.0033	0.627	0.005	0.493	0.031	***
23	0.175	0.000	0.621	0.026	***	0.256	0.000	0.495	0.022	***	0.538	0.073	0.494	0.037	***	0.558	0.010	0.500	0.040	***
24	0.392	0.000	1.444	0.026	***	0.329	0.000	0.424	0.017	***	0.657	0.041	0.487	0.043	***	0.568	0.011	0.490	0.035	***
25	0.228	0.000	0.512	0.023	***	0.223	0.000	0.415	0.018	***	0.669	0.098	0.497	0.035	***	0.557	0.031	0.493	0.035	***
26	0.320	0.000	0.832	0.033	***	0.337	0.000	0.671	0.027	***	0.635	0.006	0.488	0.035	***	0.526	0.011	0.488	0.035	***
27	0.338	0.000	1.092	0.040	***	0.417	0.000	0.674	0.027	***	0.439	0.042	0.495	0.039	***	0.551	0.049	0.488	0.034	***
28	0.471	0.000	1.382	0.041	***	0.372	0.000	0.869	0.031	***	0.568	0.025	0.496	0.041	***	0.716	0.000	0.499	0.035	***
29	0.286	0.000	0.914	0.039	***	0.311	0.000	0.611	0.025	***	0.593	0.008	0.493	0.037	***	0.530	0.026	0.494	0.034	***
30	0.060	0.000	0.383	0.017	***	0.108	0.000	0.192	0.008	***	0.733	0.026	0.503	0.036	***	0.646	0.005	0.481	0.036	***
31	0.271	0.000	0.784	0.034	***	0.304	0.000	0.654	0.032	***	0.625	0.006	0.493	0.042	***	0.599	0.009	0.490	0.039	***
Average	0.222	0.000	0.714	0.025		0.277	0.000	0.499	0.021		0.608	0.023	0.494	0.037		0.564	0.014	0.489	0.034	

*** = p-value < .001

TABLE 3: Offline Binary Classification Results.

Classifier	Window Size	F-Score		Recall		Precision	
		Mean	95% CI	Mean	95% CI	Mean	95% CI
Random Forest	1	75.94%	[74.82%, 77.05%]	75.10%	[73.96%, 76.24%]	79.58%	[78.35%, 80.82%]
	2	83.08%	[81.98%, 84.18%]	83.68%	[82.6%, 84.75%]	84.94%	[83.73%, 86.16%]
	5	90.66%	[89.73%, 91.6%]	90.12%	[89.05%, 91.18%]	93.11%	[92.16%, 94.07%]
	10	92.34%	[91.24%, 93.45%]	92.00%	[90.81%, 93.2%]	94.50%	[93.44%, 95.57%]
	30	94.37%	[93.31%, 95.42%]	93.51%	[92.35%, 94.66%]	97.35%	[96.61%, 98.08%]
	60	95.00%	[94%, 96%]	94.03%	[92.9%, 95.16%]	98.36%	[97.87%, 98.85%]
SGD	1	76.03%	[74.95%, 77.11%]	73.45%	[72.32%, 74.59%]	82.71%	[81.5%, 83.92%]
	2	83.85%	[82.71%, 84.99%]	83.37%	[82.27%, 84.47%]	86.44%	[85.14%, 87.74%]
	5	89.88%	[88.7%, 91.05%]	90.54%	[89.43%, 91.64%]	90.66%	[89.37%, 91.95%]
	10	90.38%	[89.17%, 91.59%]	92.33%	[91.21%, 93.45%]	90.10%	[88.77%, 91.43%]
	30	89.58%	[88.33%, 90.84%]	93.32%	[92.23%, 94.41%]	88.04%	[86.66%, 89.41%]
	60	88.13%	[86.88%, 89.38%]	93.52%	[92.46%, 94.58%]	85.75%	[84.35%, 87.15%]
MLP	1	77.07%	[76.01%, 78.12%]	78.04%	[77.02%, 79.06%]	77.60%	[76.43%, 78.77%]
	2	83.00%	[81.92%, 84.07%]	86.06%	[85.08%, 87.03%]	81.81%	[80.61%, 83%]
	5	89.76%	[88.73%, 90.8%]	91.72%	[90.91%, 92.54%]	89.29%	[88.12%, 90.45%]
	10	91.94%	[90.89%, 92.99%]	92.88%	[92.02%, 93.73%]	92.37%	[91.2%, 93.54%]
	30	91.85%	[90.65%, 93.04%]	92.53%	[91.39%, 93.68%]	92.49%	[91.24%, 93.74%]
	60	91.76%	[90.62%, 92.91%]	91.70%	[90.46%, 92.93%]	93.62%	[92.59%, 94.65%]
LinearSVC	1	77.19%	[76.15%, 78.23%]	76.67%	[75.61%, 77.73%]	79.47%	[78.31%, 80.63%]
	2	84.52%	[83.4%, 85.63%]	85.56%	[84.52%, 86.6%]	84.93%	[83.69%, 86.17%]
	5	90.10%	[88.97%, 91.22%]	91.19%	[90.17%, 92.21%]	90.40%	[89.17%, 91.63%]
	10	91.92%	[90.86%, 92.98%]	93.30%	[92.43%, 94.17%]	92.00%	[90.82%, 93.19%]
	30	90.66%	[89.42%, 91.9%]	93.18%	[92.04%, 94.33%]	90.01%	[88.68%, 91.33%]
	60	90.79%	[89.56%, 92.02%]	93.24%	[92.07%, 94.42%]	90.09%	[88.8%, 91.39%]

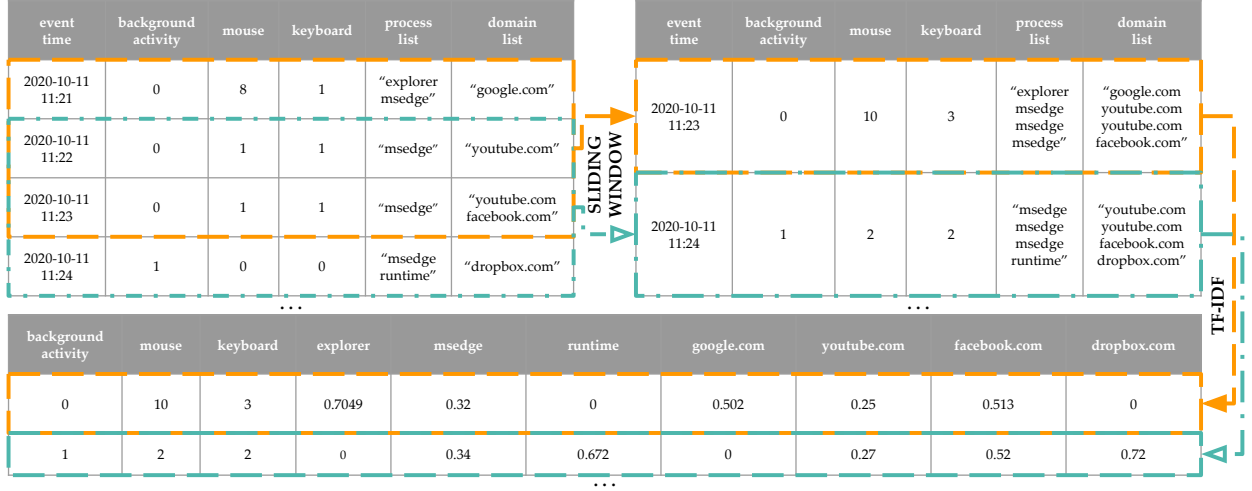
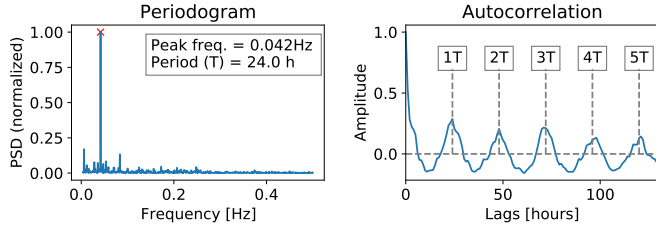
Fig. 6: Example of sliding window feature extraction with window size $t = 3$ minutes (illustrative TF-IDF values).

TABLE 4: Offline One-Class Classification Results.

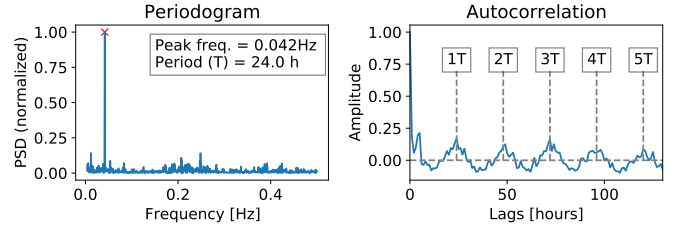
Classifier	Window Size	F-Score		Recall		Precision	
		Mean	95% CI	Mean	95% CI	Mean	95% CI
Isolation Forest	1	3.08%	[2.84%, 3.32%]	49.99%	[49.99%, 49.99%]	4.44%	[3.31%, 5.56%]
	2	3.08%	[2.84%, 3.32%]	49.98%	[49.97%, 49.98%]	11.15%	[9.29%, 13.01%]
	5	3.08%	[2.28%, 3.88%]	49.92%	[49.87%, 49.97%]	23.51%	[15.72%, 31.3%]
	10	3.13%	[2.34%, 3.92%]	49.83%	[49.72%, 49.95%]	23.79%	[16.43%, 31.15%]
	30	3.93%	[3.68%, 4.19%]	48.96%	[48.59%, 49.34%]	37.51%	[35.49%, 39.52%]
	60	7.80%	[6.8%, 8.8%]	48.62%	[48.02%, 49.22%]	44.10%	[42.3%, 45.9%]
One-Class SVM (Kernel = RBF)	1	44.97%	[43.62%, 46.31%]	55.09%	[54.03%, 56.15%]	52.70%	[52.25%, 53.15%]
	2	45.36%	[40.15%, 50.56%]	53.69%	[50.31%, 57.07%]	53.82%	[50.89%, 56.76%]
	5	46.20%	[45.04%, 47.36%]	56.88%	[55.81%, 57.95%]	53.44%	[52.77%, 54.1%]
	10	51.64%	[47.56%, 55.72%]	58.07%	[55.18%, 60.96%]	57.37%	[52.68%, 62.06%]
	30	60.20%	[56.14%, 64.25%]	60.37%	[57.32%, 63.42%]	74.30%	[66.69%, 81.92%]
	60	62.17%	[58.23%, 66.12%]	59.64%	[56.7%, 62.59%]	84.10%	[76.64%, 91.56%]
One-Class SVM (Kernel = Linear)	1	60.48%	[59.49%, 61.48%]	64.11%	[63.26%, 64.97%]	62.38%	[61.33%, 63.42%]
	2	60.92%	[59.79%, 62.04%]	63.55%	[62.63%, 64.48%]	65.07%	[63.7%, 66.44%]
	5	63.74%	[62.44%, 65.05%]	63.70%	[62.69%, 64.71%]	71.84%	[70.32%, 73.36%]
	10	67.48%	[63.86%, 71.1%]	64.88%	[62.05%, 67.71%]	80.29%	[75.04%, 85.54%]
	30	68.00%	[64.25%, 71.75%]	64.64%	[61.62%, 67.66%]	83.31%	[77.41%, 89.2%]
	60	65.95%	[62.1%, 69.8%]	63.73%	[60.63%, 66.83%]	82.09%	[75.32%, 88.87%]

TABLE 5: Online Binary Classification Results.

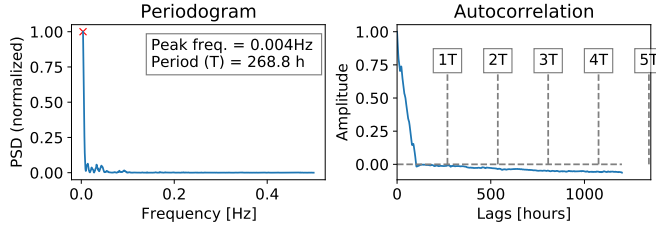
Classifier	Window Size	F-Score		Recall		Precision	
		Mean	95% CI	Mean	95% CI	Mean	95% CI
Adaptive Random Forest (No Drift Detection)	1	73.39%	[66.46%, 80.33%]	63.23%	[54.63%, 71.83%]	94.56%	[93.08%, 96.05%]
	2	81.66%	[75.24%, 88.07%]	73.11%	[64.97%, 81.26%]	97.96%	[97.26%, 98.65%]
	5	93.16%	[90.11%, 96.2%]	88.36%	[83.71%, 93.01%]	99.60%	[99.26%, 99.95%]
	10	97.07%	[95.4%, 98.73%]	94.70%	[91.82%, 97.58%]	99.92%	[99.88%, 99.96%]
	30	99.60%	[99.37%, 99.84%]	99.25%	[98.8%, 99.7%]	99.97%	[99.95%, 99.99%]
	60	99.90%	[99.81%, 99.98%]	99.81%	[99.65%, 99.97%]	99.99%	[99.97%, 100%]
SGD	1	85.90%	[84.26%, 87.54%]	80.76%	[78.86%, 82.65%]	93.64%	[92.92%, 94.35%]
	2	93.98%	[92.9%, 95.07%]	91.45%	[90.11%, 92.79%]	97.29%	[96.77%, 97.82%]
	5	98.45%	[98.07%, 98.83%]	97.80%	[97.36%, 98.25%]	99.13%	[98.81%, 99.44%]
	10	99.28%	[99.15%, 99.42%]	98.95%	[98.79%, 99.11%]	99.63%	[99.51%, 99.74%]
	30	99.22%	[99.02%, 99.43%]	98.69%	[98.35%, 99.04%]	99.81%	[99.77%, 99.84%]
	60	99.29%	[98.99%, 99.59%]	98.82%	[98.33%, 99.32%]	99.87%	[99.84%, 99.9%]
Perceptron	1	87.83%	[86.92%, 88.74%]	87.43%	[86.5%, 88.36%]	88.23%	[87.34%, 89.13%]
	2	94.50%	[93.92%, 95.08%]	94.53%	[93.95%, 95.11%]	94.48%	[93.9%, 95.06%]
	5	98.12%	[97.84%, 98.4%]	98.16%	[97.88%, 98.44%]	98.08%	[97.8%, 98.37%]
	10	99.17%	[99.07%, 99.27%]	99.22%	[99.12%, 99.31%]	99.12%	[99.01%, 99.23%]
	30	99.69%	[99.65%, 99.74%]	99.72%	[99.68%, 99.77%]	99.66%	[99.62%, 99.7%]
	60	99.80%	[99.77%, 99.84%]	99.85%	[99.81%, 99.88%]	99.76%	[99.73%, 99.8%]



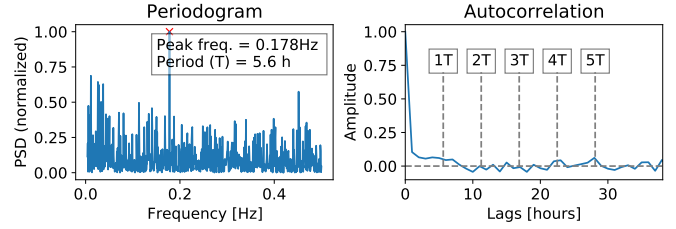
(i) User 1, condition 1



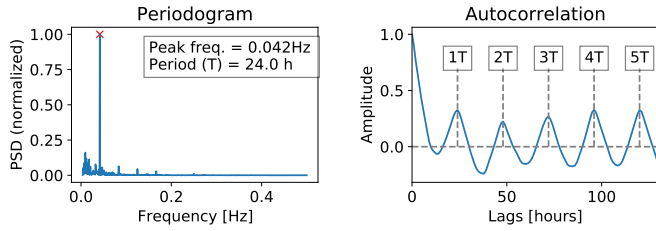
(ii) User 1, condition 2



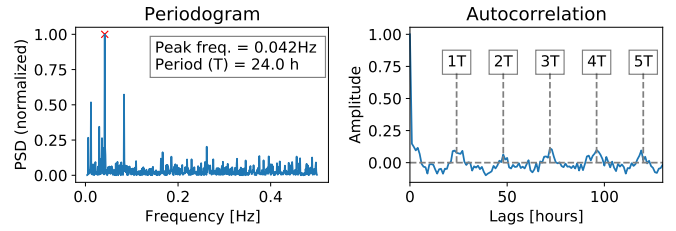
(iii) User 2, condition 1



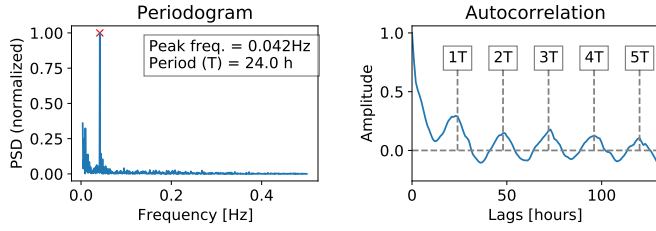
(iv) User 2, condition 2



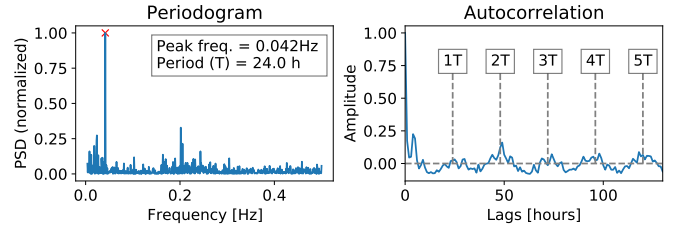
(v) User 3, condition 1



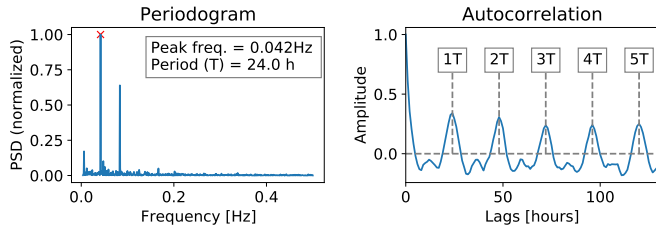
(vi) User 3, condition 2



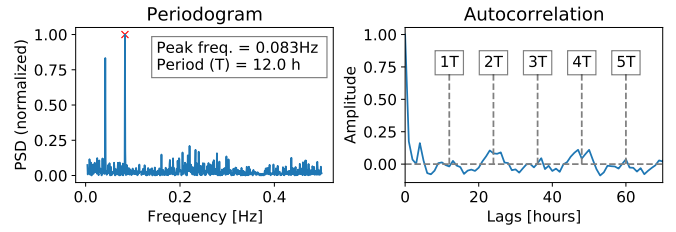
(vii) User 4, condition 1



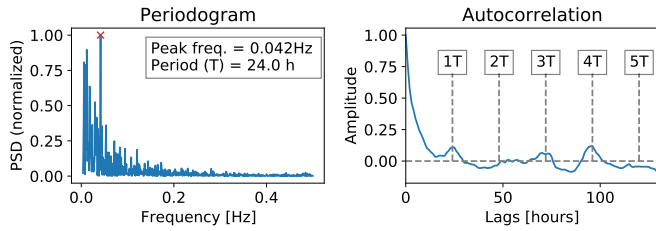
(viii) User 4, condition 2



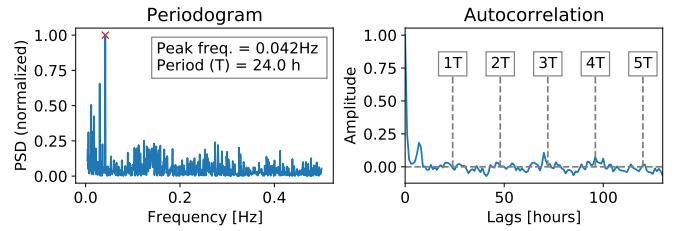
(ix) User 5, condition 1



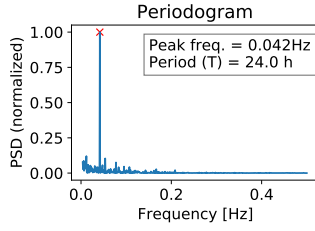
(x) User 5, condition 2



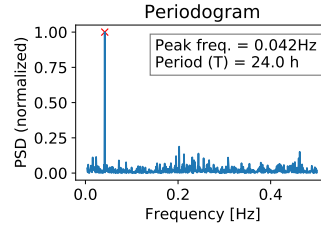
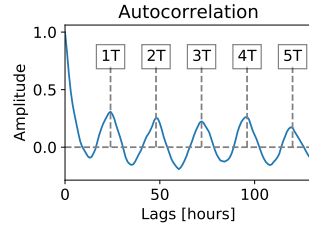
(xi) User 6, condition 1



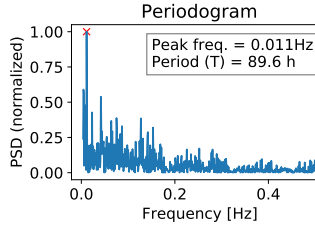
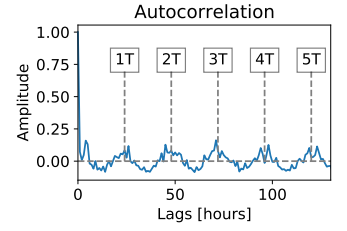
(xii) User 6, condition 2



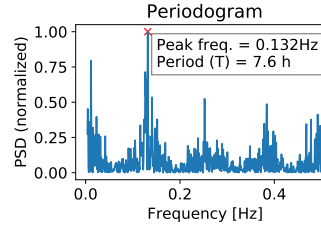
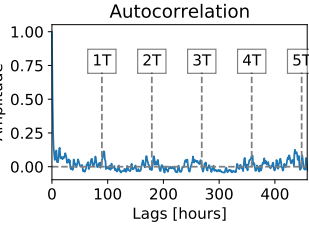
(xiii) User 7, condition 1



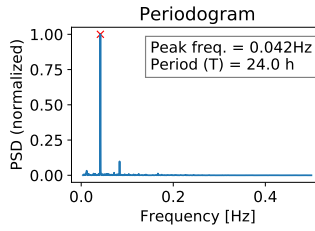
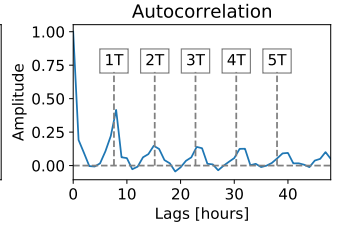
(xiv) User 7, condition 2



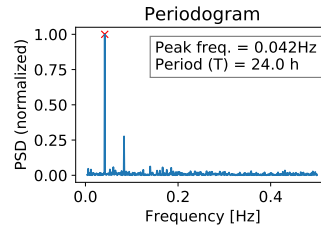
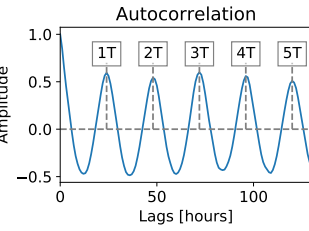
(xv) User 8, condition 1



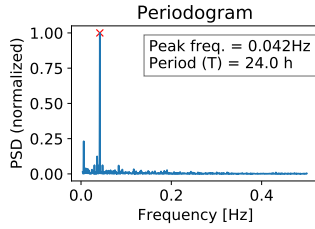
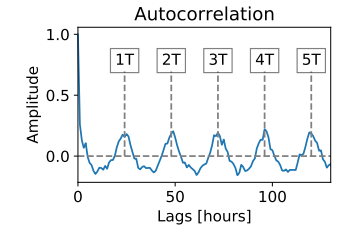
(xvi) User 8, condition 2



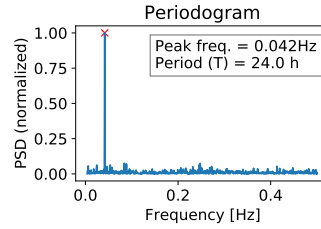
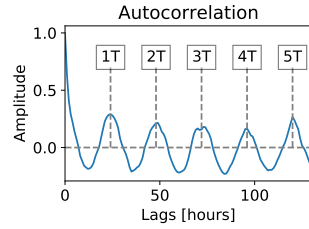
(xvii) User 9, condition 1



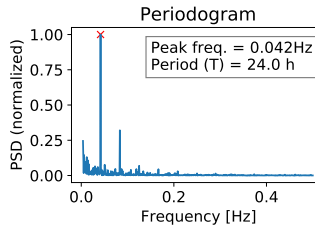
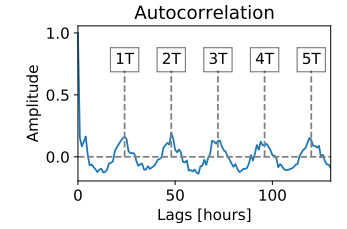
(xviii) User 9, condition 2



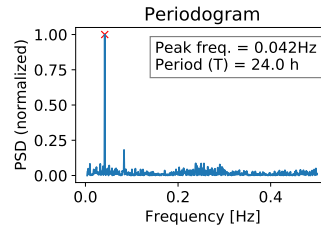
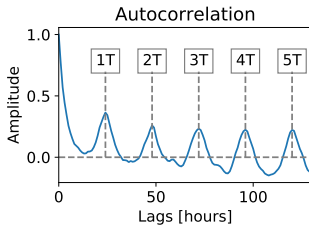
(xix) User 10, condition 1



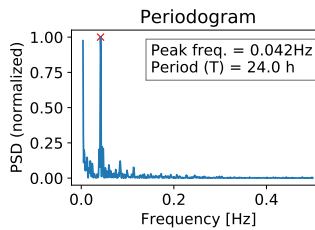
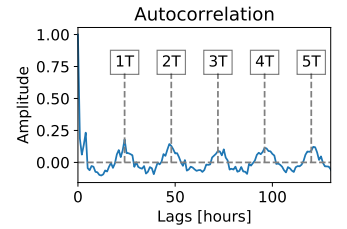
(xx) User 10, condition 2



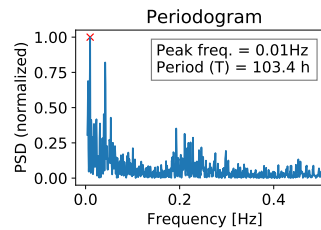
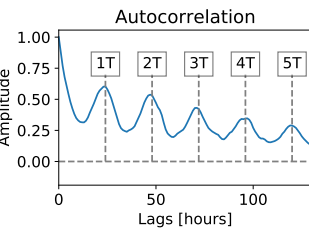
(xxi) User 11, condition 1



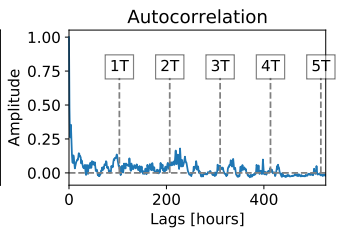
(xxii) User 11, condition 2

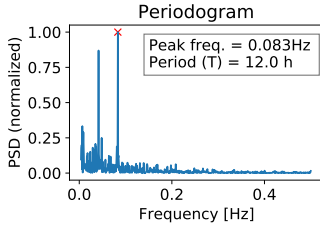


(xxiii) User 12, condition 1

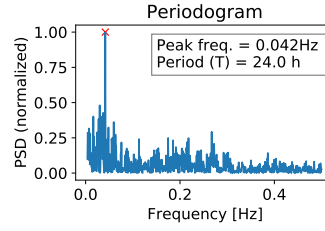
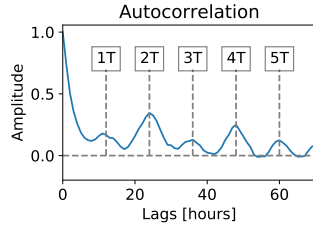


(xxiv) User 12, condition 2

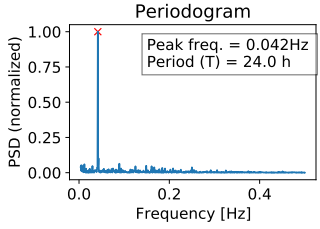
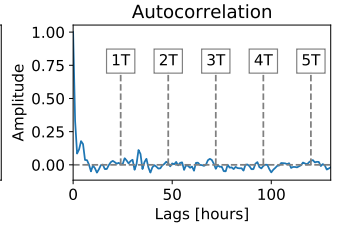




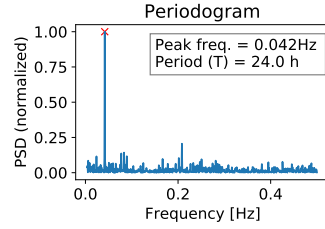
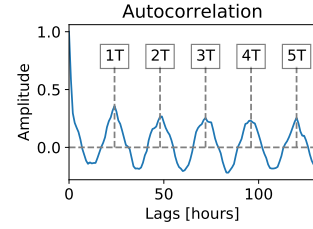
(xxv) User 13, condition 1



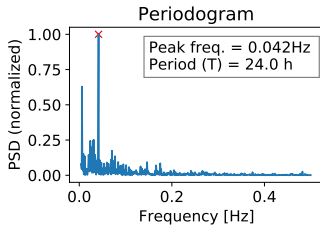
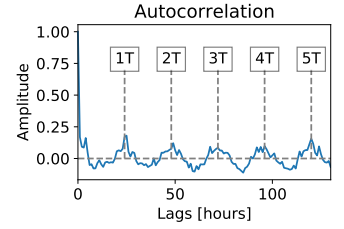
(xxvi) User 13, condition 2



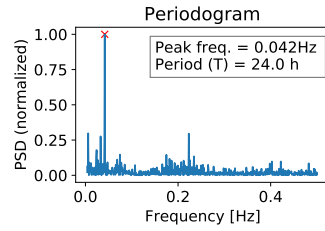
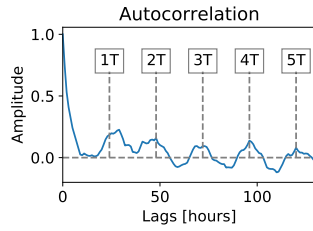
(xxvii) User 14, condition 1



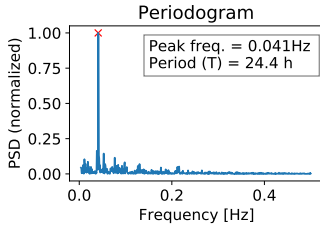
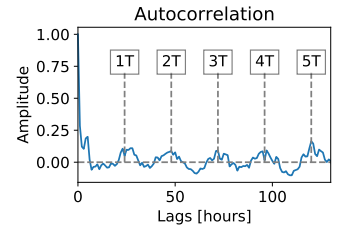
(xxviii) User 14, condition 2



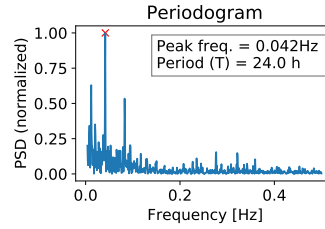
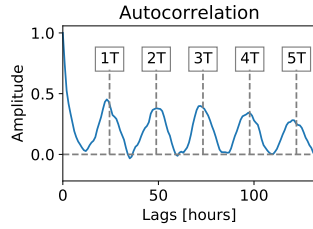
(xxix) User 15, condition 1



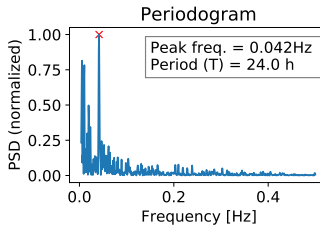
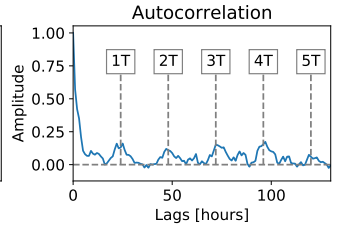
(xxx) User 15, condition 2



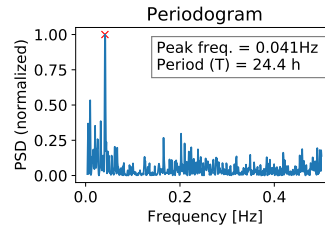
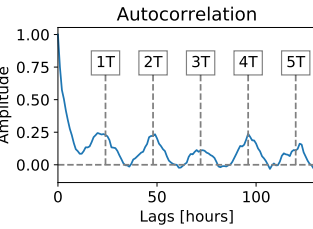
(xxxii) User 16, condition 1



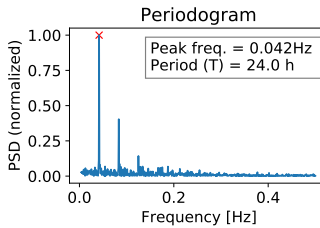
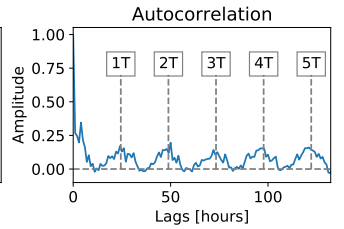
(xxxiii) User 16, condition 2



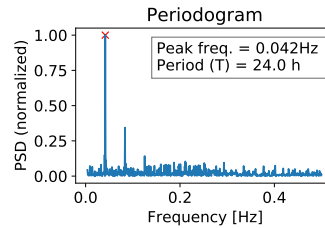
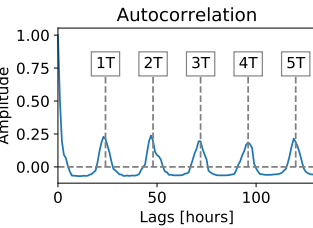
(xxxiv) User 17, condition 1



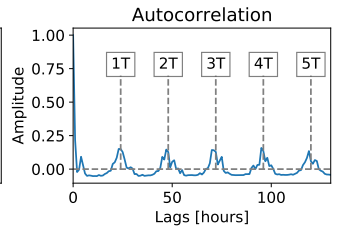
(xxxv) User 17, condition 2

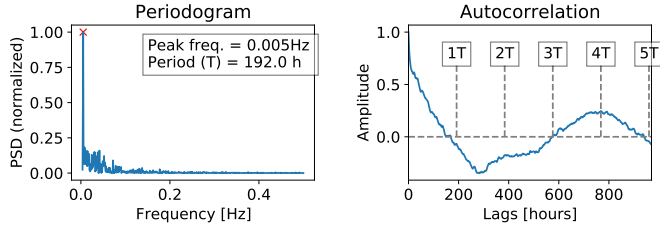


(xxxvi) User 18, condition 1

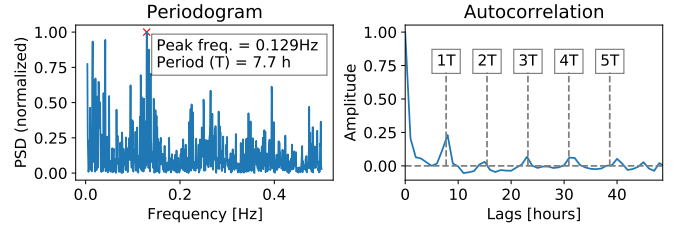


(xxxvii) User 18, condition 2

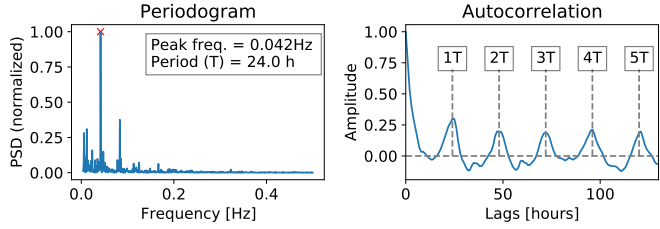




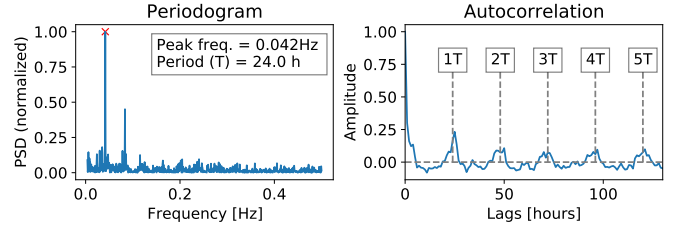
(xxxvii) User 19, condition 1



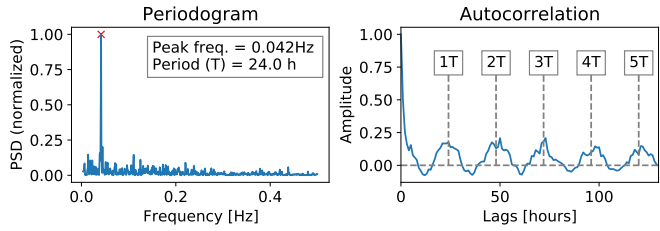
(xxxviii) User 19, condition 2



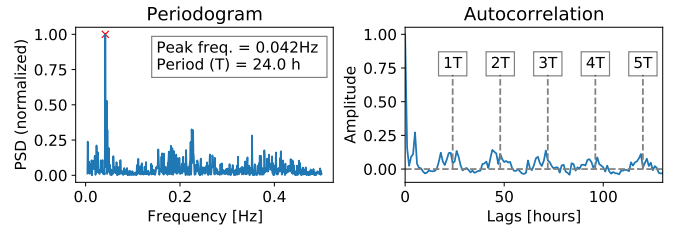
(xxxix) User 20, condition 1



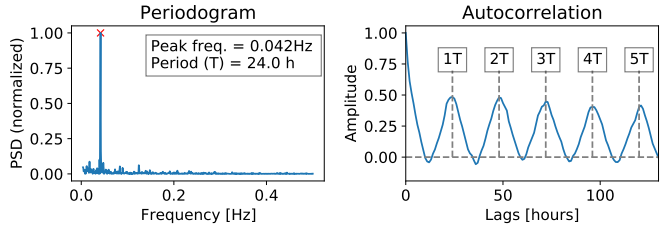
(xl) User 20, condition 2



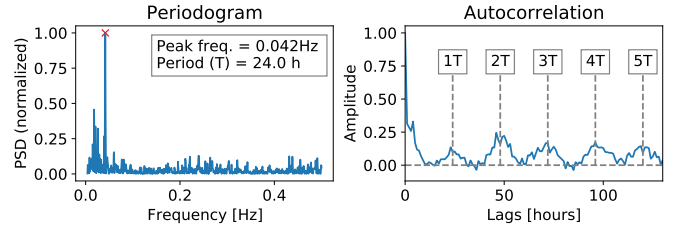
(xli) User 21, condition 1



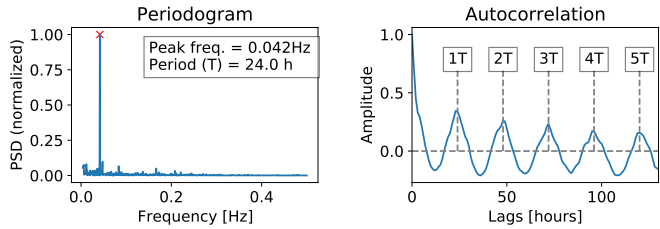
(xlii) User 21, condition 2



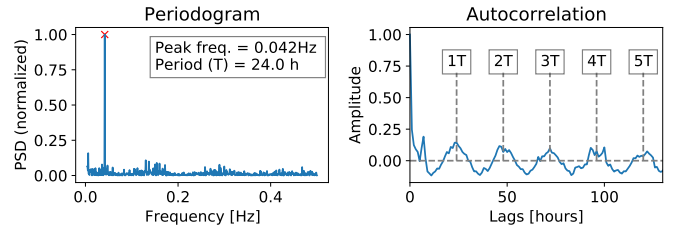
(xliii) User 22, condition 1



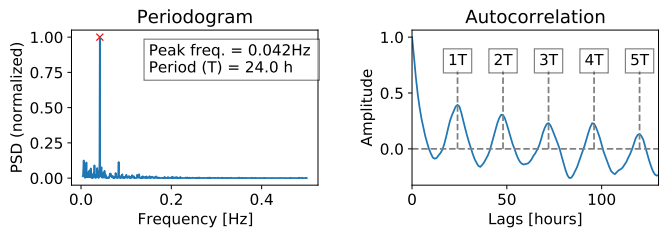
(xliv) User 22, condition 2



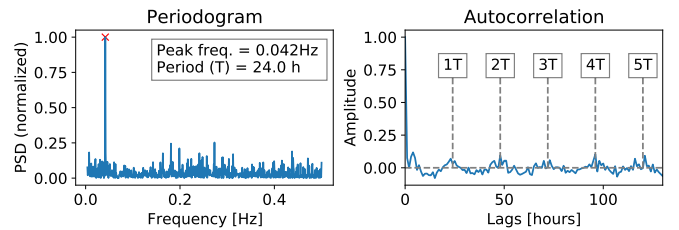
(xlv) User 23, condition 1



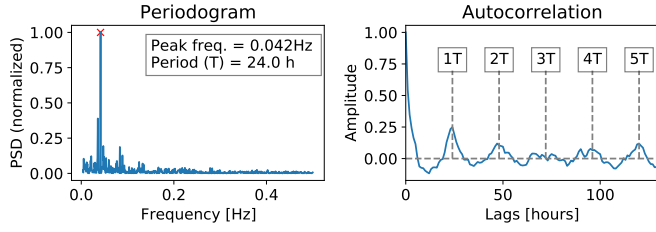
(xlvi) User 23, condition 2



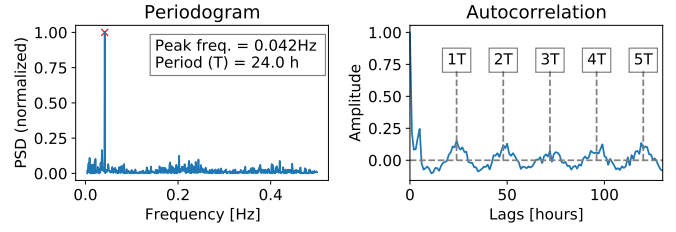
(xlvii) User 24, condition 1



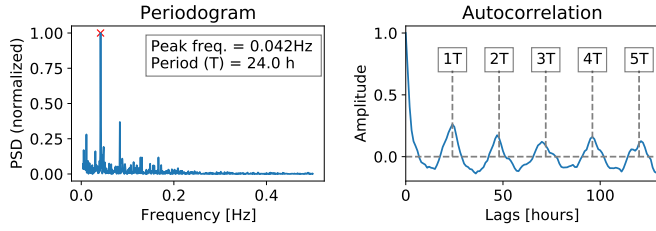
(xlviii) User 24, condition 2



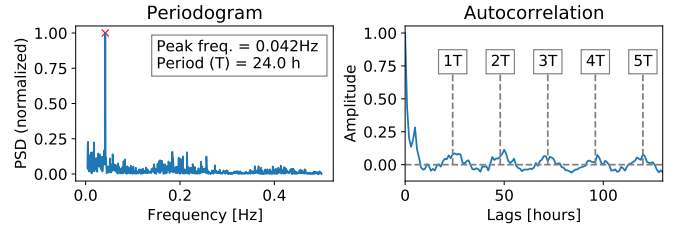
(xlix) User 25, condition 1



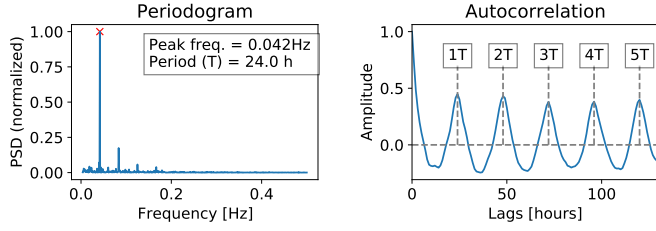
(l) User 25, condition 2



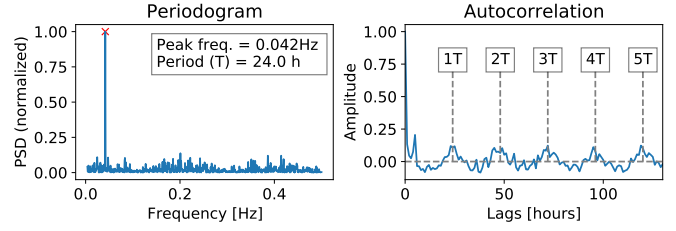
(li) User 26, condition 1



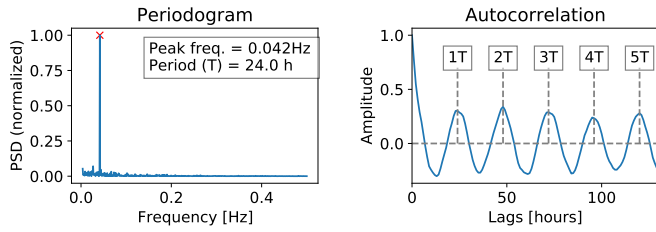
(lii) User 26, condition 2



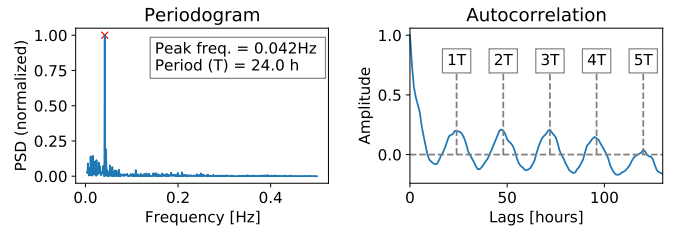
(liii) User 27, condition 1



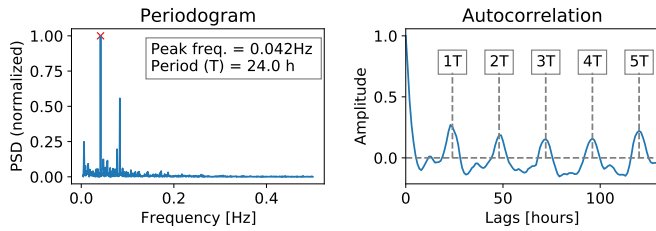
(liv) User 27, condition 2



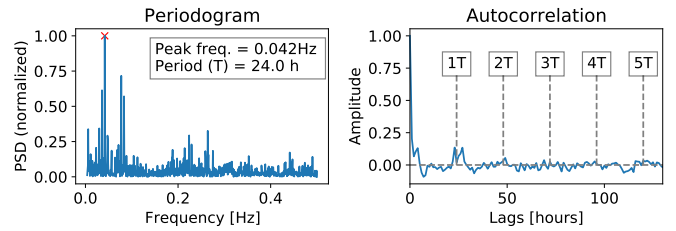
(lv) User 28, condition 1



(lvi) User 28, condition 2



(lvii) User 29, condition 1



(lviii) User 29, condition 2

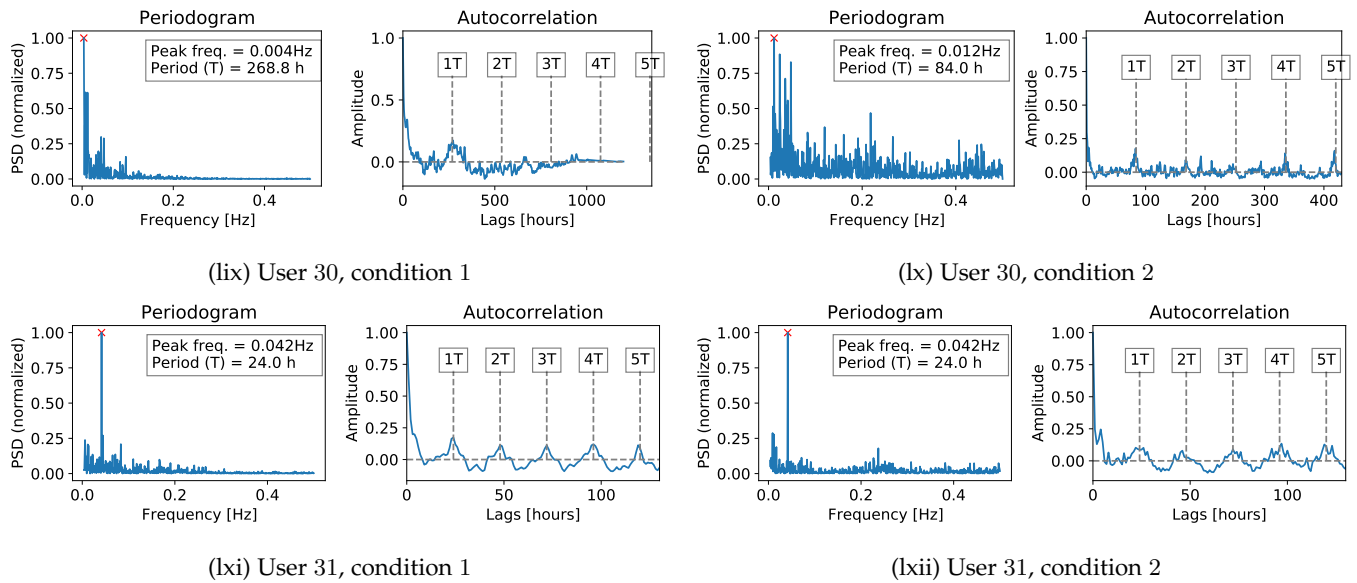


Fig. 7: Periodogram and autocorrelation results for all 31 participants in condition 1 (with background process activity) and condition 2 (without background process activity).