Not Quite Yourself Today: Behaviour-Based Continuous Authentication in IoT Environments

ANDRAŽ KRAŠOVEC*, Faculty of Computer and Information Science, University of Ljubljana, Slovenia
DANIEL PELLARINI, Faculty of Computer and Information Science, University of Ljubljana, Slovenia
DIMITRIOS GENEIATAKIS, European Commission, Joint Research Centre (JRC), Ispra, Italy
GIANMARCO BALDINI, European Commission, Joint Research Centre (JRC), Ispra, Italy
VELJKO PEJOVIĆ, Faculty of Computer and Information Science, University of Ljubljana, Slovenia

The shortcomings of the traditional password-based authentication mechanism are becoming increasingly apparent as we transition from "one user – one device" to a richer "multiple users – multiple devices" computing paradigm. The currently dominant research direction focuses on on-device biometrics, which require sensitive information, such as images of a user's face, to be constantly streamed from a single recording source, often the device on which a user is getting authenticated. Instead, in this work we explore the possibilities offered by heterogeneous devices that opportunistically collect non-sensitive data in smart environments. We construct an IoT testbed in which we gather data pertaining to a person's movement in space, interaction with certain physical objects, PC terminal usage, and keyboard typing, and construct machine learning models capturing the person's behaviour traits. We commence our examination with models constructed from data sensed during a previously-completed task run and with such models we achieve up to 68% user identification accuracy (c.f. 7% baseline) among up to 20 individuals. Taking into account the limits of behaviour persistence we then revise our approach to continuously refine the model with the most recently sampled sensor data. This method allows us to achieve 99.3% user verification accuracy and successfully prevent a session takeover attack within 12 seconds with less than 1% of false attack detection.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Security and privacy** → *Authentication*; • **Computing methodologies** → *Machine learning approaches.*

Additional Key Words and Phrases: continuous authentication, Internet of Things (IoT), machine learning

ACM Reference Format:
Andraž Krašovec, Daniel Pellarini, Dimitrios Geneiatakis, Gianmarco Baldini, and Veljko Pejović. 2020. Not Quite Yourself Today: Behaviour-Based Continuous Authentication in IoT Environments. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 136 (December 2020), 29 pages. https://doi.org/10.1145/3432206

*The author was a trainee at the European Commission, Joint Research Centre (JRC) during the course of this work.

Authors' addresses: Andraž Krašovec, ak6688@student.uni-lj.si, Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, Ljubljana, Slovenia, 1000; Daniel Pellarini, dp2317@student.uni-lj.si, Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, Ljubljana, Slovenia, 1000; Dimitrios Geneiatakis, dimitrios.geneiatakis@ec.europa.eu, European Commission, Joint Research Centre (JRC), Ispra, Via Enrico Fermi 2749, Ispra (VA), Italy, 21027; Gianmarco Baldini, gianmarco.baldini@ec.europa.eu, European Commission, Joint Research Centre (JRC), Ispra, Via Enrico Fermi 2749, Ispra (VA), Italy, 21027; Veljko Pejović, veljko.pejovic@fri.uni-lj.si, Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, Ljubljana, Slovenia, 1000.

2474-9567/2020/12-ART136 $15.00
https://doi.org/10.1145/3432206

1 INTRODUCTION

Data privacy and access control are becoming increasingly important as the growth of ubiquitous computing leads to the expansion of both the number of points where the data is collected, stored, and accessed from, as well as the range of domains in which the data is used. Authentication is the staple method for ensuring access control, yet, traditional password-based authentication remains plagued with security and usability issues [18, 22]. A recent survey finds that privileged access credential abuse represents 74% of all security breaches in US and UK companies [13]; consequently, more than $100 billion are spent every year on corporate cybersecurity [13]. The failure of traditional authentication methods is even more prominent in service-based multi-user environments – the clash of a burdensome password-based (re)authentication and the need for a timely reaction is most clearly observed in hospital emergency rooms. In one hospital the authors of an aptly named “You want my password or a dead patient" article find the code for accessing emergency supply room written on the room doors, since preventing authentication due to a forgotten code can lead to fatal consequences [29].

Moving from “what you know" to “what you are", biometric information enables authentication based on a user’s fingerprint, iris scan, voice, gait, and other properties. While relieving the user from the burden of remembering a password, current commodity biometric solutions still rely on explicit user authentication at the beginning of an access session and later assume that the same user keeps the privileges until explicit de-authentication or a time-out. Despite the benefits, imposing such a one-off authentication solution in a dynamic environment shared by users of different access privileges, such as a hospital or a hotel reception desk, would unlikely improve the overall security. Thus, a surgeon authenticated by a fingerprint might be rushed to an emergency room, meanwhile an unintended user might snitch the terminal. Another issue arises due to the perceived intrusiveness of biometric-based solutions – even in more private environments, such as smart homes, users are uncomfortable with the collection of biometric data [42].

Behavioural biometrics can address these limitations by creating a more complete user profile, through the analysis of certain user traits during everyday tasks over a time dimension [60]. Furthermore, by shifting authentication from a one-off occurrence to a continuous activity, behavioural biometrics also addresses the problem of keeping a device secure even after it has been unlocked: continuous authentication (CA) ensures that, throughout the usage session, the authenticated user indeed remains the one accessing the resource. Traditional biometric information, e.g. face recognition, can be harnessed for continuous authentication. Nevertheless, due to the data sensitivity and the need for high security protection, CA solutions based on “soft” digital identities inferred from data that are not considered personal or sensitive emerge as a more attractive alternative. Such soft identities are often created by service providers in order to distinguish, identify, and monitor users as unique digital entities, e.g. while they navigate online [31, 47]. For instance, HTTP cookies are used for session management, personalisation, and user tracking, as HTTP is a stateless protocol, while web beacons are used mainly to track users when navigating from one web site to another. Yet, neither of these, nor approaches that rely solely on device and web browser fingerprinting [2, 3, 7] can provide strong indicators for user identification, as another entity might use exact the same device or browser, or follow the same sequence of URLs.

Projected to surpass the 75 billion deployed devices in 2025 [53], the Internet of Things (IoT) represents an attractive source of data upon which future continuous authentication solutions could be based. Data sensed by a personal computing device, including the user’s GPS location, acceleration, and other modalities sensed by e.g. a mobile phone, already contain enough information to detect when a rogue user gets into the possession of the mobile phone [49]. On the other hand, the informativeness of data collected by sensor-enabled devices associated with a particular environment, rather than devices carried by the intended user, is yet to be assessed. These data, however, could prove immensely important for CA in multiuser environments. Non-sensitive data originating from heterogeneous IoT devices, such as accelerometers, gyroscopes, force sensors, passive infra-red (PIR) sensors, could potentially reveal enough information on user behaviour to make constructing digital identities for CA

feasible. Furthermore, IoT environments alleviate the need to equip each user with their own device (e.g. a smartphone) and avoid a single point of failure. Finally, a large number of cheap heterogeneous sensors could easily be deployed in almost any space and could provide a multifaceted robust view of a user's behaviour.

In this work we advance beyond a single location/device and a single-point-in-time authentication and develop an approach for continuous authentication in an IoT environment. We commence with the analysis of the informativeness of different IoT sensors and their corresponding features when it comes to user authentication. We develop an experimental IoT testbed architecture consisting of different types of sensors such as accelerometer, gyroscope, force, PIR, etc., and computational systems (i.e. a PC, a WiFi network), similar to a range of modern workplaces. In the above testbed, we collect data generated by up to twenty users pertaining to their movement in space, interaction with certain physical objects, PC terminal usage and keyboard typing, and construct machine learning models capturing the participants' behaviour traits. We are aware of the relatively low number of participants for a CA research paper and plan on extending the dataset with new participants at a later date. Using our random forest-based model these behaviour patterns are sufficient for independent user authentication with above 60% accuracy (c.f. 7% baseline). However, when harnessed as a means to confirm the identity of an already authenticated user, behaviour models extracted from IoT data provide more than 99% accuracy and successfully prevent an attack scenario in which a rogue user takes over a previously authenticated session.

The main contributions of our work can be summarised as follows:

(1) We identify the most informative sensors and features for IoT-based authentication in office-like scenarios. Namely, those related to a PC terminal and work-desk interaction;
(2) We develop an authentication mechanism that relies solely on non-sensitive data gathered from the environment;
(3) We identify limited temporal persistence of the sensed behaviour traits and advocate for continually updated authentication models;
(4) We devise a CA solution that confirms an already-authenticated user's identity with up to 99.3% accuracy;
(5) Harnessing the classifier confidence we develop an approach that successfully detects credentials abuse in an Iot environment after only 12 seconds with less than 1% false rejection rate;
(6) We publicly release our dataset as well as the accompanying analysis and models[1], thus facilitate further efforts in IoT-based continuous authentication research.

Our study is conducted in a carefully designed office-like IoT testbed; nevertheless, we believe that the tremendous growth of IoT device popularity will soon make our findings applicable to a wide range of environments, from hospitals, over enterprises, to factories. Furthermore, while in this work we focus on authentication, behaviour modelling demonstrated in this paper could also be used for service personalisation, for example. In summary, we believe that sensor-based behaviour modelling represents the next step in making our IoT environments genuinely smart.

2 RELATED WORK

Biometrics-based behavioural authentication approaches can help address the usability issues of traditional authentications systems, which are typically based on something that the user knows. In fact, even in case of multi-factor authentication, something that the user knows (e.g. in the form of passwords) is often used as the first factor of an authentication system. Biometrics-based behavioural authentication can also help address the security problems of traditional authentication systems, by implementing techniques that allow devices to keep authenticating the user in a transparent manner even after a device has been unlocked.

Touch dynamics is one of the most common methods used to perform behaviour-based authentication on a mobile device [44]. In touch dynamics, the authentication data comes from the way the user interacts with the

[1] https://gitlab.fri.uni-lj.si/lrk/ca-iot

touch screen of the device. It has been shown in [20] that touch dynamics data can vary significantly among users, and Shen et al. [48] have also studied how data coming from mobile phones' sensors, such as accelerometers and magnetometers, can be used for inferring input information in a data set compiled based on 30 users. Accuracy rates in their experiment reached 85%; similar solutions have been proposed by other research [6, 40, 59, 61] as well. However, this authentication method cannot be applied in cases where users primarily interact with a device using something other than a touchscreen, like a hardware keyboard (for example on a shared computer) or via voice commands (for example with a smart speaker).

Face recognition is a CA mechanism that is gaining traction thanks in large measure to its ubiquity and to the improving quality of smartphone front-facing cameras. Compared to touch dynamics, face recognition presents new challenges related to the environment in which it is being used: authenticating a user in low-light, for example, or recognising the user even when in a different pose, requires additional attention. Furthermore, face masks that became ubiquitous during the COVID-19 pandemics render face recognition-based authentication all but impossible in shared spaces. An approach tailored to smart home security applications has been proposed in [57]. Unfortunately, spoofing attacks through impersonation are still an open challenge when it comes to face recognition applications [58].

Gait dynamics continuous authentication mechanisms analyse how the user walks, using sensor data coming from the device's accelerometer and gyroscope. Authentication systems based on gait dynamics have been proposed in [25, 55]. Hybrid approaches, in which gait dynamics has been used together with other features, such as face recognition [21] and Electroencephalography (EEG) [63], have been proposed. This approach shows its limitations when applied to users with impaired mobility, even with temporary limitations caused by an accident, and cannot therefore cover the entirety of the user base.

One specific form of authentication, which is also used in this paper, is related to keystroke dynamics, i.e. the method of analysing the way users type on a computer keyboard, and of classifying users on the basis of their regular typing rhythm. People can be recognised by their typing style. A recent review is provided in [46] while previous studies have evaluated different distance metrics to identify typing users, such as in [8, 24, 41, 45]. A study focused specifically on mobile keystroke biometrics has been presented in [9], while a wearable-assisted sensor-based typing dynamics CA system has been proposed in [1]. In addition to monitoring keystrokes, as in traditional typing-based CA systems, this approach also uses a wearable device, such as a smartwatch, to capture the typing rhythm of the user through the device's accelerometer and gyroscope sensors. Data collected from the wearable and the keyboard, including key-pressing pressure, hand rotation, and hand displacement during typing are then used for continuous authentication. While error rates of this approach were low in the researchers' testing (up to 1% EER - Equal Error Rate), the approach relies on yet another device a user needs to have on them – something we aim to avoid in our work. In addition to this, the proposed approach requires a regular first-factor authentication system, and is therefore not meant to be a complete replacement of the existing authentication system.

However, neither touch dynamics, nor face recognition, nor gait dynamics, nor wearable-based mechanisms (nor indeed any monomodal approach to authentication) can currently present a holistic solution for CA. Additionally, all current approaches tend to work, or at least offer their best results, only in a limited set of circumstances, or with some clearly-predefined requirements, which do not necessarily reflect a real-world scenario. A promising approach involves using multi-modal behaviour-based mechanisms instead (such as the framework presented by Crawford et al. in [16]): by monitoring different aspects of device usage over a certain period of time and looking for significant deviations in the patterns observed, it is possible to notice unexpected behaviour and lock the device for preventative security. This approach has therefore the potential to cater to every device owner, no matter how they use the device, and regardless of the type of device used: all that is necessary is for them to use it as they normally would. In this area of authentication, several approaches have been suggested, from application usage or communication information such as phone calls and text messages [32, 35, 56] to profiling

users based on web browsing activity [62] or network traffic meta data [14]. But such methods might not be as successful in scenarios where users do not use their devices for web surfing, or more in general when there is no active network usage. Authentication systems based on motion patterns and device interaction have been proposed in [12, 51], although they both focus on smartphone usage. Other approaches based on general system features like memory, CPU and network data, as provided by the underlying operating system (OS) [36], process creation, registry key changes and file system actions [52], or mouse movements, keystrokes and application usage [17] have also been proposed. But the accuracy of approaches based on system features risks becoming lower when more than one user can utilize the same device (e.g. in the case of a shared device in an office or laboratory). Our own research presented here hints that these features might indeed become less informative when a device is shared by multiple users for the same task.

Mobile devices and wearables are not the only means to perform biometrics-based behavioural authentication: an approach based on RFID sensors has been proposed in [23]. In their experiment, Huang et al. used a multi-modal Convolutional Neural Network (CNN) for feature extraction, and a Support Vector Machine classifier for user authentication. The accuracy of human motion monitored through RFID reached 97% during testing, but it is possible that such an approach might not be able to achieve similar results in environments other than the meeting room where the original test was performed, because such approaches might learn environment-specific features and therefore overfit. In addition to this, it is possible that some very informative features for user authentication are being left out, by focusing exclusively on human motion captured by RFID sensors. Another approach, this time based on IoT smart devices such as Amazon Echos and Google Home voice assistants, has been proposed in [43]. IoT smart devices, much like smartphones, also come equipped with a series of sensors which allow them to sense their environment and react to users' interactions. Behavioural features extracted from network traffic could therefore also be collected from IoT devices in a smart home environment. Metadata such as port numbers, packet sizes, and device types have been analysed in this experiment. Over a period of three weeks, Ongun et al. looked at network traffic generated by 10 users who interacted with 15 IoT devices in their lab. Subsequently, they evaluated the performance of Logistic Regression, Random Forest, and Gradient Boosting models, achieving an 86% accuracy rate when classifying six users, and 97% accuracy rate when classifying five users. Nevertheless, in smart home environments, the number of users expected to be authenticated is usually very limited and clearly defined, and it is also possible that some interesting features are being lost when looking exclusively at network traffic. We therefore believe that a multi-modal approach, such as the one we propose in this paper, has the potential to cater to a wider variety of scenarios.

3 THREAT MODEL AND ASSUMPTIONS

End-user authentication is one of the foremost security measures that is essential to ensure protection of private and personal sensitive data. Nowadays, static authentication mechanism require end-users to identify themselves when first signing up for a service and then such mechanisms assume that the same end-user is active up until he/she explicitly signs outs [26]. While this procedure is end-user friendly and reduces the burden of repeating authentication efforts on behalf of end-users, it nonetheless demotes system security. This is because malicious applications and adversaries can gain access to a system where a legitimate end-user has already authenticated or after the initial sign on and subsequently perform specific attacks (e.g., masquerading).

We believe that *continuous authentication (CA)* where a user's behaviour represents a basis for ongoing re-evaluation of the user's identity, mitigates the above issues and perfectly complements static forms of authentication (including password-based and multi-factor authentication). Our goal is, thus, to use the above approach to alleviate the following threat:

A malicious agent takes over a pre-authenticated session from a legitimate user;

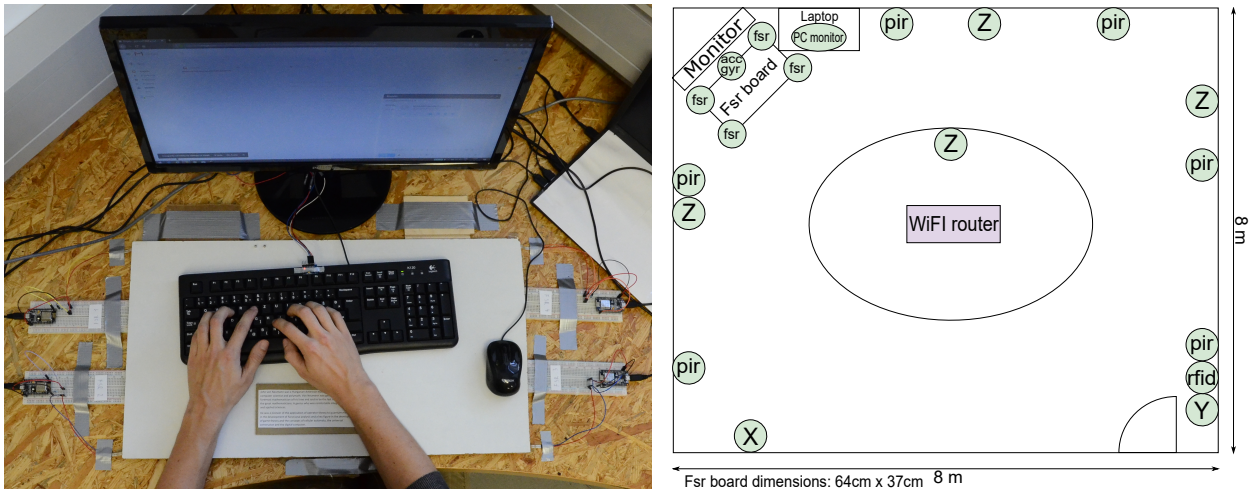

Fig. 1. LEFT: A user performing Task 1, with the force sensors on the corners of the plate, the accelerometer and the gyroscope on the keyboard, and the PC along with all the peripherals visible. RIGHT: Testbed room layout with sensor locations. Task 3 requires the user to find the box marked *X*, then *Y* and at the end find further instructions in one of the boxes marked *Z*. All boxes are equipped with hall sensors. Instructions are presented in Appendix B.

We assume that the above attack is happening in a smart environment hosting a range of sensors, and it is exactly the reflection of a user's behaviour provided by these sensors that we use as a foundation of our machine learning (ML) based CA approach. So the work at hand studies to what extent ML can be used for user authentication and can enforce additional authentication mechanism whenever mis-behaviour is identified in a smart environment. We *do not* study whether a ML classifier may resist to targeted attacks, i.e. where a malicious agent has the ability to observe and mimic the legitimate user. Instead, we leave such considerations for future work.

As a preliminary to our CA approach, we first examine the ability of the smart space to provide the necessary differentiation among behaviours. This would increase the security of such spaces and enable seamless secure human-smart space interaction. Humans are different in their behaviour both within and between subjects. Therefore, we first examine whether the human behaviour can be captured at one moment and then used as a template for matching the behaviour in another moment (Section 6). We show a limited differentiability of the behaviour, thus we proceed to examine the ability of a smart environment to continuously track the behaviour of a person (Section 7). We demonstrate that with-a-single-task-run behaviour is much more persistent and that we can provide 99% authentication accuracy in case data sensed from the user at a current moment is matched against recently-sensed data.

4 IOT TESTBED ARCHITECTURE

IoT devices are deployed in a wide range of environments, from smart homes, over public infrastructure, to factories. Nevertheless, IoT-based authentication is likely to bring the most benefits to close-space environments accessed by a relatively modest number of users. These environments include shared offices, factory control rooms, hospital terminal rooms, reception lobbies in hotels, and similar. Thus, in this work we design a testbed that reflects the general characteristics of the above environments.

We instrument a furnished, quiet air-conditioned $8m \times 8m$ room on the third floor of our department building with IoT devices, such as sensors and computing equipment (see Figure 1). During our study the room was accessed exclusively by a single study participant at a time. In particular, our architecture consists of the following sensing and computing components:

- **A Personal Computer (PC)**: a Lenovo laptop with a 4-core Intel CPU and 8 GB of memory running Microsoft Windows 10 OS, connected to an external display, mouse and keyboard, thus recreating a configuration of a desktop computer or a shared terminal;
- **An accelerometer and a gyroscope** attached to the keyboard, installed to detect users' typing behaviour, as well as keyboard positioning;
- **Four force sensors** positioned at each corner of a rigid rectangular plate under a keyboard and a mouse, installed to detect users' posture behind the computer;
- **Six passive infrared (PIR) sensors**, capturing users' movement patterns around the room;
- **Six hall sensors** placed in boxes positioned around the room; these sensors detect interaction with the boxes, which are part of one of the predetermined tasks that the participants were asked to complete;
- **A software-based PC resources monitor** collecting CPU, memory and network usage, installed in order to capture users' computer usage behaviour;
- **A WiFi router** providing Internet connectivity for the sensed data transfer;
- **A server** hosting an MQTT broker and back-end components, including an MQTT subscriber, an object-relation mapping (ORM) manager and a PostgreSQL database.

On the data collection side, each hardware sensor is connected to a dedicated microcontroller, as our preliminary analysis has demonstrated the need for such a connection in order to achieve the fastest possible data sampling rates. We opted for NodeMCU microcontrollers[2] as they have an integrated Wi-Fi module that simplifies sensor control and data collection via an IP network. The microcontrollers' firmware driving the data collection and transport is written in the Arduino embedded programming platform.

The message queue telemetry transport protocol[3] (MQTT) is used for data collection. The sensors act as MQTT publishers, while the server consumes the data. On the server side, a Mosquitto MQTT broker[4] receives the data sent by the microcontrollers and the PC, and forwards them to the MQTT subscriber (implemented in Python's Django framework[5]), which inserts it in a queue for the RabbitMQ Message broker[6]. The ORM manager (also implemented in Django) takes messages from the queue of the Message broker and inserts the data into a PostgreSQL database[7]. From there on the data are ready for visualisation and further processing.

The microcontrollers are instructed to collect the data from the moment a user enters to the moment the user exits the room; these events are detected via an RFID key card read at the entrance of the room. Raw data are collected with different granularities for different sensors. Accelerometer, gyroscope, force, and PC resources monitoring sensors are sampled periodically, whereas PIR and hall sensors report "on-change" events. The summary of the data collected at each sensor is shown in Table 1.

5 DATA SET

In this section, we explain the approach for compiling our behavioural biometrics dataset. We describe the tasks that the users perform in the IoT environment presented in Section 4, examine the properties of the collected

[2]https://www.nodemcu.com/
[3]https://www.mqtt.org/
[4]https://www.mosquitto.org/
[5]https://www.djangoproject.com/
[6]https://www.rabbitmq.com/
[7]https://www.postgresql.org/

Table 1. Data collected from each of the sensors together with the sampling rate.

Sensor	Collected data	Sampling Rate
Accelerometer	16 bit value of x, y and z axes	200 Hz
Gyroscope	16 bit value of x, y and z axes	200 Hz
Force	8 bit value of force	200 Hz
PC CPU utilisation	per core usage in %	1 Hz
PC mem. consumption	total memory usage in %	1 Hz
PC network data	cumulative of sent and received packages	1 Hz
PIR	motion/no motion	on change
Hall	motion/no motion	on change

Table 2. Task running time statistics, including the mean, standard deviation, minimum and maximum running time of the task of the filtered data in seconds. Last column represents the average amount of recorded data points of a given task in thousands (*k*).

Task	Mean	Std	Min	Max	Average amount of data
Task 1 (email)	367 s	118 s	213 s	603 s	803 k
Task 2 (Web browsing)	491 s	209 s	202 s	1090 s	1004 k
Task 3 (treasure hunt)	496 s	247 s	255 s	1325 s	986 k

data, and analyse the features extracted for the purpose of further exploration of (continuous) authentication in IoT environments (Sections 6 and 7). Recognising that a multimodal dataset reflecting user behaviour in a smart environment could facilitate further research on authentication, service personalisation, and context-based adaptation, producing a publicly available dataset is one of the goals of our work.

5.1 Experimental Tasks and Data Collection Protocol

We devise the following experimental tasks to simulate common user actions in a smart environment, and collect data pertaining to these actions and sampled by sensors that are a part of the architecture defined in Section 4. Table 2 lists tasks length statistics and the amount of data gathered on average.

(1) The first task instructs a user to send an email with predetermined text written on a card found under the terminal's keyboard. The text is about 600 characters long; to prevent shortcuts (i.e. copy-pasting) between the first and the second run of the task, we use different text for the two runs. The content of the two texts can be found in Appendix A. The main focus of this task is to stimulate varied responses from the accelerometer and the gyroscope attached to the keyboard, as most of the time in this task is spent typing.
(2) The second task is intentionally less linear to encourage individual computer usage patterns. The test subjects are instructed to check the weather forecast for a certain geographical region or country for the following few days and, based on the result, suggest three tourist attractions to visit and send their recommendations via email. The choice of the region or the country is left to the subjects, but it has to differ between the two runs of the tasks. The sensor we are targeting in this task is the PC resource usage monitor, as users are not instructed which programs, tools, or websites to use to accomplish this task.
(3) The third task does not only include interaction with the PC, but also introduces indoor mobility. In particular, in a form of a "treasure hunt". The test subjects have to search for a series of labelled boxes in the room and follow the instructions (presented in Appendix B) in each box. The instructions require the subjects to use the PC to compile a data table, generate a graph, and send the graph via email. In addition,

before leaving the room, the test subjects are asked to reset the experiment. With respect to user movement, an additional degree of freedom is introduced with the presence of a large obstacle (a row of tables) in the middle of the room, forcing the subjects to navigate around it by going either to the left or the right of it. Due to extensive movement, in this task we obtain richer PIR and hall sensor data.

The experimental process starts when a test user taps her RFID key card for the first time at a reader installed at the room entrance. The subject completes the above three tasks in the presented order, with a brief rest period between the consecutive tasks. The subject then performs *the second run* of the three tasks in the same order, immediately after *the first run* is completed. The subject taps out the RFID card after each task and taps it in before starting the next one. The final tap out signalises the end of the experiment session. It should be noted that during the experiment no other entities, except for the subject under test, are present in the room.

The participants are free regarding their choice of applications to complete the assignments. Nevertheless post-experiment discussion with participants revealed that the all of them used Google Chrome for Internet browsing and email activities, despite other popular modern browsers (Google Chrome, Microsoft Edge, Mozilla Firefox and Opera) also being installed and appended to the taskbar. To produce graphs in task 3, our users relied on either Google Sheets or Microsoft Excel.

5.2 Data Collection and Filtering Approach

We recruited 21 volunteers to participate in our study. The participants' age ranges from 23 to 58 years old, 9 of them are female and 12 are male, 12 are students and 9 are employed. All of the participants are healthy adults without disabilities or any other characteristics that could introduce obvious variability of the collected data. The participants were not compensated, monetarily or in any other way, for their participation.

From the collected data we first filter out all the data corresponding to task runs in which the subjects failed to follow the instructions or there were technical difficulties which resulted in the loss or corruption of data. If one task run of a user is deemed compromised, both runs of those task of that user are discarded. The failure to follow instructions is monitored by the control of the provided content by the participants via email and a post-experiment discussion with each of the participants. For example, a subject did not copy the text from the provided card during first task's session, but only sent an email without any or with some random content. Fortunately, those occurrences were rare. To ensure that quality data is collected we prepare a real time report generator, with the purpose of reporting the amount of data each sensor provided over a period of time. With it we are able to detect faulty sensors or disturbances in WiFi connectivity. Any glitches in data collection or users' failure to full comply with the instructions may introduce artificial differences among subjects' behaviour and result in an unreasonably high evaluation accuracy of even the simplest behaviour-based authentication mechanism. Thus, after rigorously applying the filtering, we retain 17, 20, and 18 subjects for the first, the second and the third task, respectively.

In total, we gather fifteen hours and forty minutes of data in the form of 115,366,524 data points collected by our smart environment's sensors. Per run, the first task provides about 800,000 data points on average with the standard deviation of 25,000 points among users, where the second and the third task average out at roughly one million data points each with the standard deviation of 35,000 points.

5.3 Feature Extraction

We extract both time and frequency domain features from the collected data. With respect to time-domain, we extract features such as the mean, standard deviation, mean crossing rate, and others, over a time-segment. These features are commonly used in pervasive sensing, e.g. for activity recognition [39]. In addition, we engineer a new "signal" by combining dimensions from different data sources, e.g. for the accelerometer the data comes along three dimensions (x,y,z); by using the square root of the sum of squares at every time step within every

Table 3. Features extracted from each of the sensor types. The amount of data points varies per sensor because of the different sampling rates, with the infrared data only recorded on change. Range of values of cumulative amount of data at PIR sensor indicate the on-trigger nature of the sensor, i.e. some PIR sensors are triggered more often than others.

Data Source	Extracted Features	Cumulative amount of data
Accelerometer	mean, sd, mean crossing rate, mean derivative value, 10 Hz frequency bins	39,000,000
Gyroscope	mean, sd, mean crossing rate, 10 Hz frequency bins	39,000,000
Force sensor	mean, sd, mean crossing rate, 10 Hz frequency bins	39,000,000
CPU utilisation	mean, sd, minimum, maximum, mean crossing rate	57,600
Memory consumption	mean, sd, minimum, maximum, jump-related features	57,600
Network data (num. of packets)	mean, sd, start-end delta, jump-related features	57,600
PIR sensor	activity class	10 – 1,000

time segment, where n represents the count of the sensor readings of a given segmented time chunk, we obtain:

$$s_i = \sqrt{x_i^2 + y_i^2 + z_i^2},\ i = 1, \ldots, n$$

Such an aggregated (intensity) signal is also calculated for the gyroscope (three dimensional data) and CPU usage data (four cores). Joining the signals in such a way is possible due to the fact that there is an equal number of readings from all the dimensions in every segmented time chunk and these data points are aligned in time.

We also extract frequency domain features from the accelerometer, the gyroscope, and the force sensors. This requires that we first apply a Fast Fourier Transform (FFT) over the different data segments and then calculate the corresponding features. We take each time segment chunk and use linear interpolation to ensure that we have data points spaced exactly 5 ms apart. Next, we perform the FFT with the Hann windowing function on the processed signal and calculate the magnitude at each frequency. As our sampling rate is precisely 200 Hz, the calculated magnitude range is between 0 Hz and 100 Hz. Finally, we calculate the mean of each 10 Hz frequency magnitude bin. Table 3 lists all of the extracted features.

We examine the initially extracted features and make the following observations that guide us towards defining additional features:

(1) The PC memory usage signal is mostly stationary with a few variations that are correlated to users' behaviour i.e. opening or closing a program. Thus, we introduce the notion of memory 'jumps' defined as peak values of the first derivative of the memory usage signal, and we extract four related features: (a) count (the number of jumps in a session), rate (the ratio between the count and the length of the signal), mean value (mean jump magnitude) and mean inter jump interval (mean time interval between neighbouring jumps).
(2) Network data is of cumulative nature. In fact, we recorded the number of sent and received network packages from the beginning of the session. The analysis shows that network data follows patterns similar to the memory signal. Therefore we reuse the notion of 'jump' and we also extract the cumulative bandwidth used by a task run.

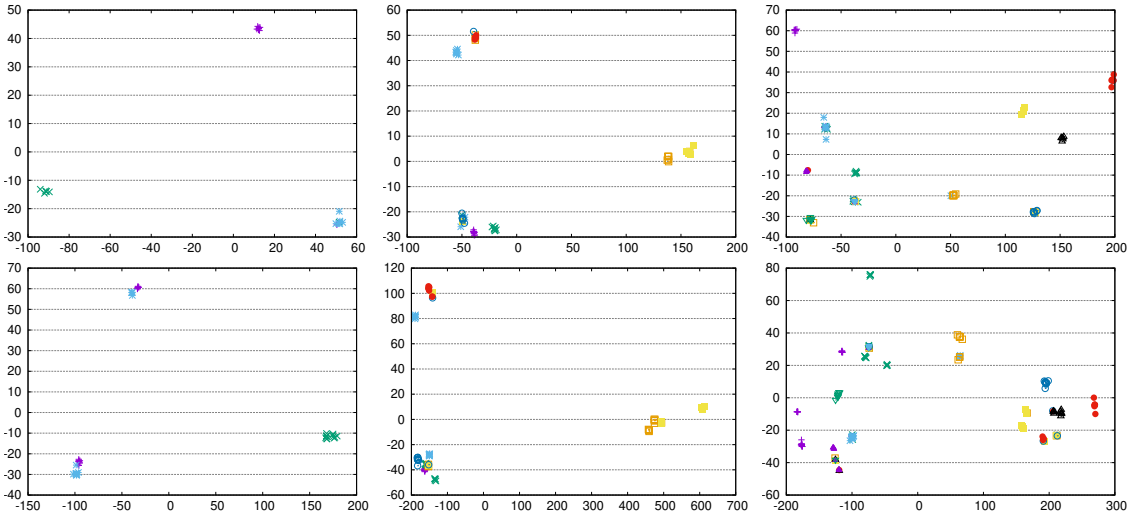

Fig. 2. Clustering in 2-dimensional LDA space for 3, 7 and 11 different users. Each colour represents data describing the behaviour of a single user during Task 3. The top row only takes into account data from the first task run, while the bottom row takes both. We observe a clear separation between users when only taking data from a single run, while inducing data of multiple runs per user introduces much higher rate of inter-class blending.

(3) Extracting high significance features from infrared sensors is a challenging task. This is because the sensitivity and the sensing range of the sensors we utilise (hc-sr501) are difficult to adjust to the given environment (we used peephole boxes to reduce the detection angle of the sensor). Instead, as soon as a person starts moving around the room, multiple sensors are triggered, often irrespective of the distance between the user and the sensor. Thus, we perform a joint analysis combining the data from different infrared sensors. We discover that a repeatable sensor activation pattern emerges and that it could be classified into one of three different user behaviour classes – *no movement*, *seating* and *walking* – and we use the discovered patterns as the values for a high-level feature we construct.
(4) We observe variation in the calculated features' values depending on the segmentation interval. In order to determine its optimal value, we calculate all the features at the following time-segment values: *s = [1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s]*.

5.4 Preliminary Analysis of Dataset Informativeness

Is the feature space we engineered in Section 5.3 informative enough to allow discrimination among our test users? To answer this question we apply the Linear Discriminant Analysis (LDA) algorithm [37] to different data subsets and extract the two most prominent components of the results. We experiment with a different number of random participants and also data segmented at different interval lengths, and examine the clustering of LDA-processed data points belonging to different users. The emergence of clear clusters in the LDA space among data taken from different users would indicate that our features are indeed informative enough and could serve as a basis for user identification.

We plot the first two components of the LDA (Figure 2) for different number of users at time segment length that yields the highest Calinski-Harabasz index [10] score for the selected user subset. We observe that with a growing number of participants, the highest score is achieved with longer segmentation intervals, as one might

expect. Furthermore with a growing number of users, the separability of individual clusters gradually decreases. Consequently, and in line with the conclusions of the meta-study of authentication mechanisms by Sugrim et al. [54], we expect the accuracy of our user authentication mechanism to drop with a rising number of users.

Figure 2 (top row) depicts data points taken from a single task run of a group of users. To assess the invariability of the sensed behaviour in the bottom row of the figure we plot the LDA points of both runs of the same groups as above. We observe that points of the same colour (corresponding to the same user) do not necessarily form a coherent cluster, indicating that the behaviour need not be stable over multiple repetitions of the same task. Moreover, LDA points calculated from data gathered during a subsequent run executed by the same user are often closer to another user's LDA points then to the LDA points of the initial task run of the original user. Projecting the data in higher dimensions and using advanced machine learning algorithms may help with carving a unique space for each user that would then ensure more robust identification. Nevertheless, we believe that significant volatility of user behaviour, evident from the above plots, precludes the "static" approach where a classifier is trained on data recorded in a single session. Instead, we advocate for an approach where the identification method is constantly updated with the most recent user behaviour data. The two identification approaches – "static" and "continuous" – are developed and evaluated in Sections 6 and 7, respectively.

6 IDENTIFICATION

In this section we examine identity verification that is based solely on data sensed by IoT sensors. Despite a wealth of approaches that rely on very diverse data, from those pertaining to driving behaviour to those related to writing styles, behaviour biometrics solutions are generally considered to be of insufficient explanatory power to provide a sound basis for user identification [60]. Thus, our goal in this work is not only to provide a standalone solution for user authentication, but also to evaluate the potential of non-sensitive data collected in IoT environments for modelling individual user behaviour and providing support to orthogonal user authentication mechanisms.

We base our analysis on the collected dataset described in Section 5. Our dataset consists of data sensed while each of the users performed each of the three tasks two times. To avoid overfitting, we take the collection of the first runs of each task type and use the aggregated data as a training set; similarly we use the aggregated second runs of each task type as a test set. Individual data instances consist of features calculated on data collected during a continuous time segment during a task run. On the average, the first runs of the tasks were slightly longer, thus we obtain a 57%-43% split of the number of instances between the training and test set per each task type. Each of the three task types is handled separately, so we generate three separate training and test sets.

We construct a machine learning (ML) pipeline where we train an ML model with the training data and evaluate its accuracy of user identification on the test data. As explained above, the training and the test data are never from the same task session. To further assess the informativeness of such an approach we examine the role of data segmentation, feature selection, and user count on the classification result:

- The feature space we use consists of 229 dimensions (engineered features) which we aim to reduce in order to discover the most informative features and sensors. Therefore, whenever we construct a classifier, we apply a univariate feature selection algorithm, utilising the mutual information [30] scoring function. We take the ten most informative features every time, as we found that this yields the best results in terms of the classification accuracy.
- After a thorough analysis of the impact of time segments on the classification accuracy (elaborated in Section 6.2) we set on using eight-second time segments throughout Section 6, unless stated otherwise. The reason for such choice is the ratio between accuracy and computational time that eight-second time segments provide. Furthermore, using longer time segments would introduce potentially intolerable delay in the identification process.

- The success of user identification approaches is often highly dependent on the number of users on which the approach is tested [54]. To assess the scalability of our solution, we examine its performance with a varying number of users. Each time we take a random sample of users from the complete dataset, repeat the process until we exhaust all user combinations, and state the average, the best, and the worst result.

In the rest of the section we first pinpoint the most promising classification method (Section 6.1), then examine how the ability to authenticate users with the collected IoT data varies with the number of users and the length of data segments (Section 6.2), continue with the identification of the most informative features and sensors (Section 6.3), and finally, inspect the classification confidence levels (Section 6.4).

6.1 Comparison of Machine Learning Algorithms

We face a classification problem, as our target classes are the user labels linked to our experiment participants. The highest possible identification accuracy is our primary goal, therefore we focus on ensemble algorithms such as Random Forest and AdaBoost that often outperform simpler alternatives when it comes to user identification. However, as we are examining sensor-based identification in smart environments we are also interested in lightweight algorithms, such as Linear Discriminant Analysis, k-Nearest Neighbours, and Naive Bayesian, that can run on resource-constrained platforms. To summarize, we experiment with the following algorithms and their settings:

- **k-Nearest Neighbours (kNN)** with the k value set to 5.
- **Linear Discriminant Analysis (LDA)** with the number of components set to 2.
- **Naive Bayesian** with Laplace smoothing.
- **Adaptive Boosting (AdaBoost)** with 1000 estimators and a decision tree set as a base estimator.
- **Random Forest** with 800 estimators and maximum number of features set to $\sqrt{10}$.

The above classifiers' hyper-parameter values were found through a random search in the hyper-parameter space of each classifier type. Performing a random search, we first define the relevant hyper-parameters of each classifier, keeping in mind the dataset we have at hand, and then determine the parameters' value ranges based on the premise that the computational time must remain within a one second threshold. The initial hyper-parameter space is large, so we decide to perform multiple iterations of this random search. Within each iteration the algorithm takes different random value sets of hyper-parameters and performs user identification, providing classification accuracy as the output and ranks the hyper-parameter sets based on the achieved classification accuracy. After each iteration we prune the parameter value ranges towards the best scoring ones in order to avoid searching in the unpromising parts of said value ranges. The initial idea was to finish it off with an additional grid search step, but it proved redundant as it did not provide any performance benefits, while being time and resource consuming.

For the scope of our evaluation, we separately take each task of the already-split data of all users (17, 20, and 18 users for each of the three task types, respectively). We then apply feature selection to the training dataset and take the ten best-performing features. Finally, we classify the test set using each of the classifiers mentioned above. Observing the results presented in Figure 3, Random Forest is on average the highest scoring classifier with 68% accuracy for Task 1, 50% for Task 2, and 63% for Task 3, followed closely by AdaBoost with 3% lower mean classification accuracy, trailing in the first task by 12.8%, but leading by 3.7% and 1.1% in the second and the third task respectively. Next is LDA which is on average performing 9.4% worse than AdaBoost. Naive Bayes and kNN algorithms also outperform the baseline of the majority classifier, but cannot compete with other algorithms we evaluated. To examine the robustness of the approach, we apply the same classification pipeline while randomising a task run included in the training and the testing set (initially, we trained on the first runs, tested on the second runs). The results remain stable and do not change in any significant manner.

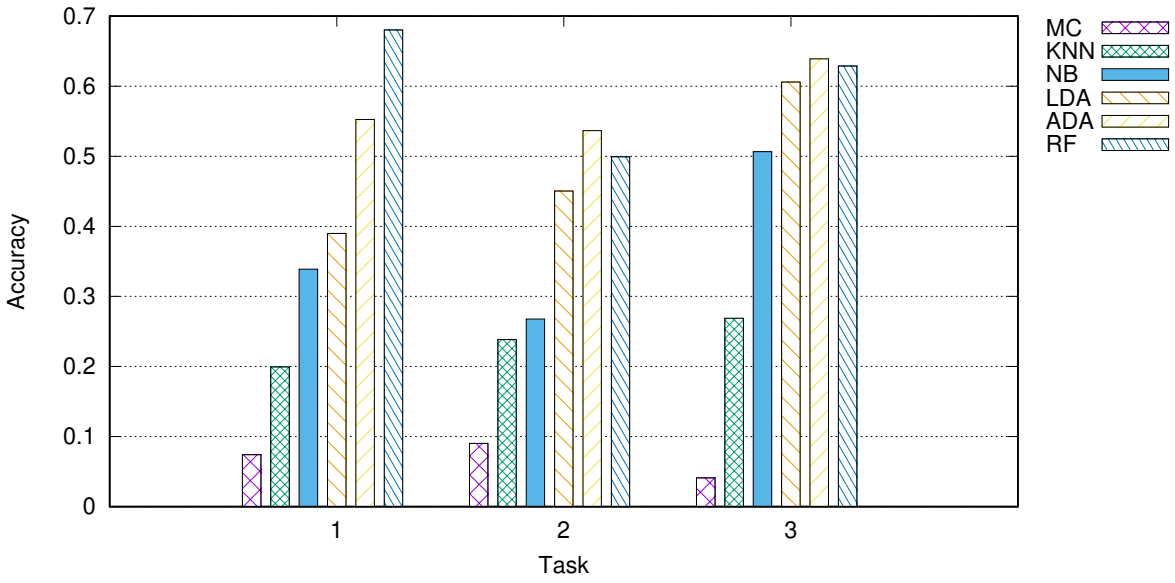

Fig. 3. Comparison of user identification accuracy between different machine learning algorithms and the baseline majority classifier (MC). The accuracy is calculated on per task basis.

Both ensemble learning methods, Random Forest and AdaBoost, perform better than other ML approaches we experimented with. The superior performance of these two compared to other algorithms we tested might stem from a large number of heterogeneous features we have at the input, since ensemble algorithms implicitly perform further feature selection during the model construction. Comparing the two, Random Forest clearly dominates in Task 1, whereas the approaches appear to be somewhat tied for the other two tasks. Thus, in the rest of the paper we focus on this method.

6.2 Number of Users & Time Segment Intervals

It should be mentioned that the choice of the ML classifier is not the only parameter we have to consider while devising a continuous authentication system. Two other important parameters we have to scrutinize are (a) the number of legitimate users that will use the system, and (b) the time segment interval used in data aggregation.

The number of legitimate users of the system is directly connected to the level of the performance we can expect from the proposed system when discerning between different users. By choosing to include more legitimate users in a system, we introduce more target classes for the classifier to choose from and also increase the probability of having two or more users with similar behaviour patterns that may further hinder the classification accuracy of our classifier.

Aggregating more data into a single data row of the data set may have a two-fold effect on the accuracy of the identification. By taking larger chunks of data, we potentially overgeneralise the data and lose information nuances that are only present in shorter time periods. In addition to this, in a real-life application it also takes a longer time for the system to be able to identify the user for the first time. On the other hand, taking time segment intervals that are too fine can have an opposite effect in the sense that we miss information about the bigger picture that can only be obtained by using longer time segment intervals.

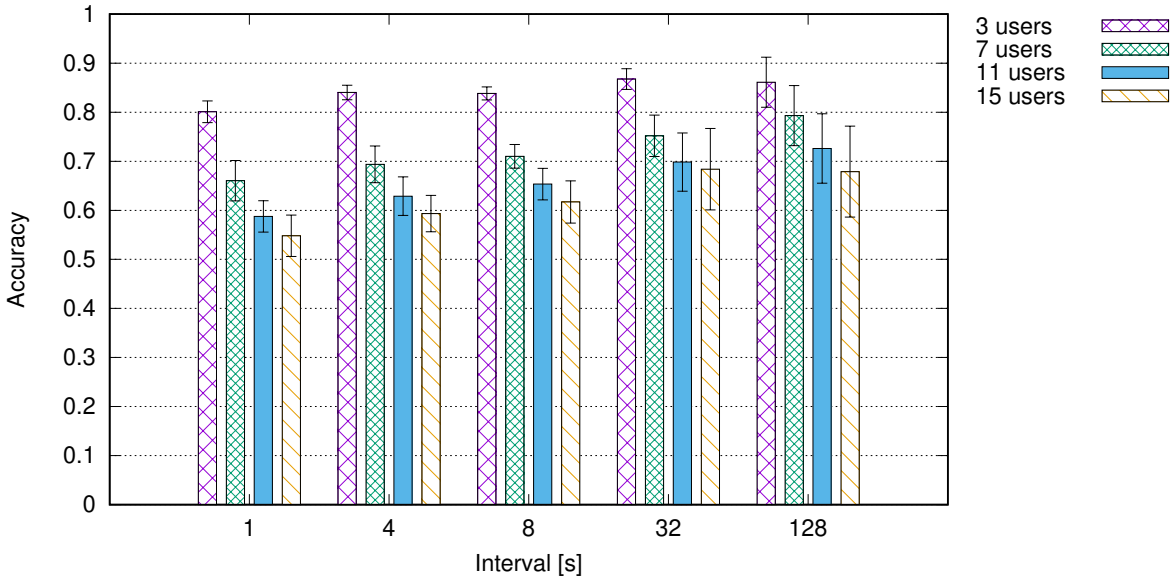

Fig. 4. Comparison of the classification accuracy for different number of users and different time segment intervals, using the Random Forest Classifier. The error bars display the standard deviation of accuracy, due to the fact that we performed 500 iterations with every parameter combination, taking random users in each iteration.

In order to test these two parameters, we define a set of different numbers of users $N = \{3, 7, 11, 15\}$ and a set of different time segment intervals $S = \{1s, 4s, 8s, 32s, 128s\}$. As taking only one user subset would likely lead to biased results, as the users differ among themselves with regards to identifiability, we take the Cartesian product of sets N and S. For each pair we perform 500 iterations for each of the three tasks, taking $n \in N$ random users at each iteration and utilising the ML pipeline, described in the introduction of this section, but limiting ourselves to the Random Forest classifier only.

Figure 4 depicts the average classification accuracy of the three tasks achieved by taking pairs of different numbers of users and time segment intervals, as described above. Observing the classifiers' performance with different number of users we see a clear trend confirming our assumption that introducing more users has a negative effect on the classification accuracy of the classifier. Furthermore we observe that the drop in accuracy lessens with the rising number of users, as well as with the rising time segment interval. This may indicate relative scalability of the approach, albeit this should be confirmed with data collected from a larger pool of users.

Differences in accuracy regarding time segment intervals are much less prominent. Still, we can recognise the 32 second segments as the best performing in terms of the classification accuracy. Even though, having in mind only minor accuracy differences, shorter time segments would be a more preferable choice – identifying the user after more than half a minute and then having to wait for another half a minute for each consecutive identification could be prohibitively long for certain real-life application. In light of this observation, we will use eight-second time segment intervals for the remainder of this section.

6.3 Sensor and Feature Informativeness

Understanding which engineered features and consequently which sensors are the most informative is essential for guiding future work on IoT-based authentication systems. Furthermore, it would also help us better understand

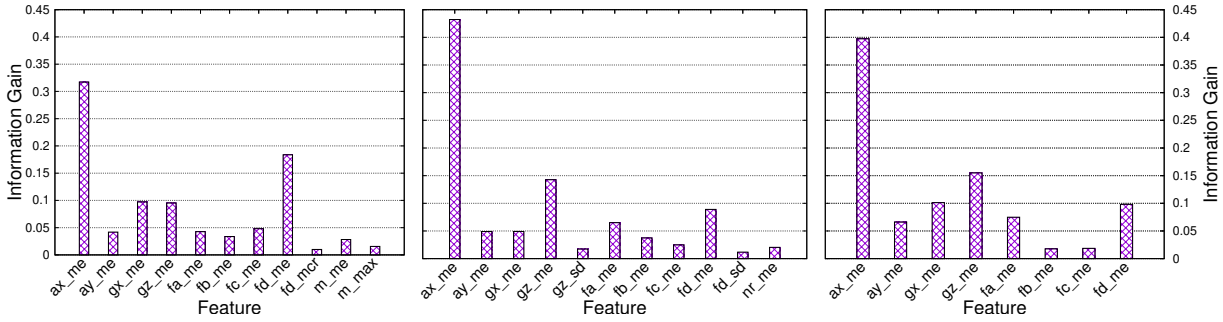

Fig. 5. Information gain of features with an amount higher than 1% of total entropy for each of the three tasks.

which user traits are the most distinguishable and could potentially also indicate how sensors should be placed in a smart environment in order to achieve the highest user authentication accuracy.

To assess the feature informativeness, we use the Information Gain (IG) as it is commonly used in feature selection and informativeness calculation [4, 11]. Different tasks require different routines and user behaviour, so we calculate IG on a per-task basis in order to detect the discrepancies in the amount of information each feature provides. In Figure 5 we plot the IG of features that surpassed the 1% mark of the total entropy.

The feature acronyms noted on the X-axis follow a simple key. The first character denotes the sensor, the second (optional) character denotes the axis (or no character which represent joined axes or none) and the two or three characters after the underscore denote the feature calculated from the denoted sensor/signal (*me* – mean, *max* – maximum, *sd* – standard deviation, *mcr* – mean crossing rate). We observe that the mean value is by far the most informative feature, as it takes twenty six out of thirty places on the three plotted figures and it is the exclusively plotted feature of the third task. The remaining three features which seem relevant are the maximum and the mean crossing rate in the first, and the standard deviation in the second task. Interestingly, none of the frequency domain features were selected in any of the task classifiers. The prevalence of the mean, especially with the accelerometer, gyroscope, and force sensor data, is interesting, as it implies that certain care should be taken when selecting the ranges in which these sensors operate. While we normalised the data beforehand, thus ensured that the findings are not over-fit to the given equipment, we note that sensors of drastically different sensing ranges might provide over- or under- saturated readings limiting the discernibility among the behaviours.

Sensor-wise the accelerometer, the gyroscope and the force sensors are prevailing. They account for the 90% of the amount of features with a higher than 1% total entropy. Memory and network usage are the only other sensors found among the most informative ones. The accelerometer values (*ax_me* and *ay_me*) correspond to 2D horizontal acceleration, while the vertical movement proves much less prominent. These values correspond to users' typing force. The gyroscopes' (*gx_me*, *gz_me* and *gz_sd*) reflect small movements of the keyboard, induced either by user typing or an actual readjustment of the keyboard by the user. Signals from the force sensors correspond to users' stance while working on the computer with the *fa_me*, *fb_me*, *fc_me*, *fd_me*, *fd_sd* and *fd_mcr* representing the top-left, bottom-left, bottom-right, and top-right corner of the force plate respectively. Meanwhile the maximum and mean value of memory usage and mean value of received network packets (*m_max*, *m_me* and *nr_sd*) represent the PC usage behaviour patterns of the user, such as the kind and the amount of the programs opened and used at any given time. There are no prominent PC monitor features present in the third task and, contrary to our expectations, PIR-related features are not present either.

6.4 Confidence Levels

Belonging to the ensemble family of the machine learning algorithms, Random Forest can also provide per class probabilities of every test case by calculating the mean predicted class probabilities, where the class probability of a single tree is fraction of samples of the class in a leaf. Using this calculation we formulate two metrics which we deem helpful while constructing the continuous authentication system – true and false confidence level. The true confidence level is the average probability of the predicted class (i.e. a user's identity) when the user identity was predicted correctly, while the false confidence level is the average probability of the predicted class when the predicted and actual user identity mismatch. These two metrics can be calculated on a user, as well as on a population level.

At this point we analyse the average true and false confidence level of the whole user population for the three tasks. We observe a 15.7% disparity between the true and false confidence level, consequently demonstrating that our algorithm is on average 15.7% more confident that it predicted the true user when indeed it predicted the true user, than in cases when it predicted a wrong one. With this knowledge we can devise a continuous authentication system based on the premise that the higher the confidence level of any new predicted data, the higher the chance that the algorithm correctly identified a person. Based on the consistency of the both levels throughout the tasks one may even suggest that we analytically define the threshold and, based on the confidence level, tell if the algorithm made a mistake predicting a certain user.

7 CONTINUOUS AUTHENTICATION

The context of the users' actions changes even over the course of a single session. For example, in our experiments every user starts by entering the room, walking to the workstation, at which point she may do some additional movement around the room, type on the keyboard or browse the Internet, and so on. As demonstrated in Section 6, such a constant change of context makes it difficult to identify and authenticate a user from the data aggregated over the whole session. Nevertheless, the collected IoT sensor data can be used for continuous authentication of an already authenticated user. In this section we propose a time-sequence based approach to continuous authentication where a user's behaviour is monitored as the task plays out in the IoT environment and assess the suitability of this approach for ensuring that the originally authenticated user is indeed the one performing the task. Unlike the classification method developed in Section 6, the method presented in this section does not require a classifier to be constructed on the previous task runs. Instead, the approach examines whether the currently sensed data is in line with a recently observed human behaviour pattern and as such, in the limit case, can even be task-oblivious.

In this section, we construct a one-vs-all classifier enabling continuous authentication of a legitimate user. Rather than taking the initial task runs for training and the remaining runs for testing, we simultaneously traverse, one step at a time, over task runs coming from multiple users. At point τ in time we take the past data of the sessions and use them as a training set. The previously identified best performing classifier, Random Forest (see Section 6), is trained upon these data. We then assume that one session was authenticated by a robust one-off authentication method (e.g. a password) in the past and we then use the above classifier to infer whether the currently sensed data belongs to the initially authenticated user. Our decision to train the classifier on data coming from "parallel" sessions is a convenience that helps us ensure a balanced dataset. The data need not be collected at the same time for all the users, the only requirement is that the data correspond to different users performing the same task. However, the fact that the data can be processed in real time and in parallel for multiple users opens up interesting application domains, such as providing a CA mechanisms for a larger factory hall where a number of operators perform the same tasks in the same shift.

In our evaluation we move through the data one second at the time (contrary to the eight second time segments used in Section 6) and calculate the per-user probability distribution on the last second of the sensing data, as

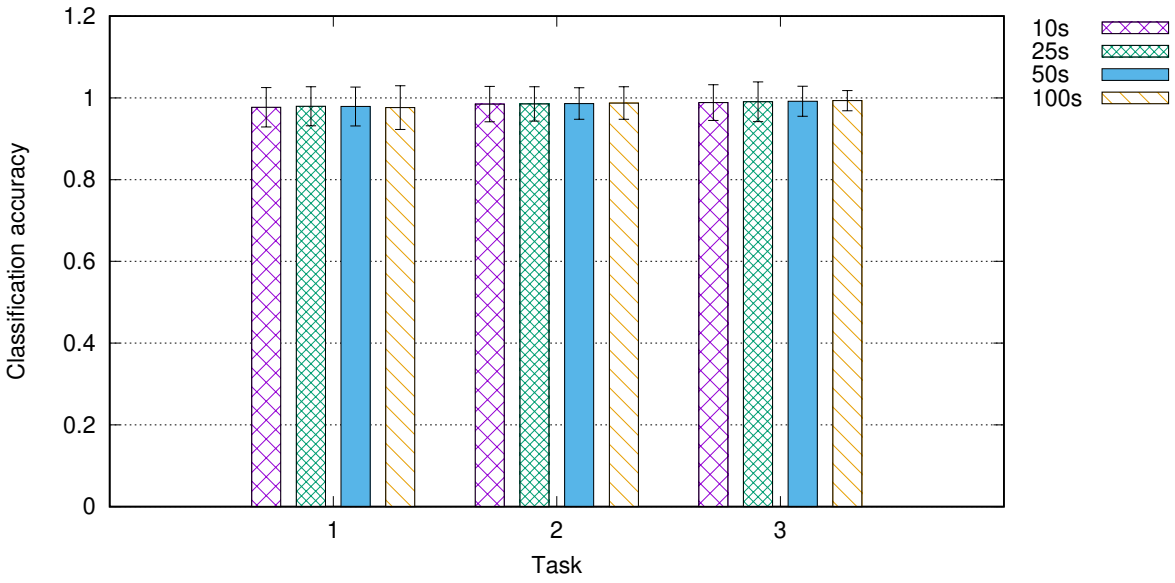

Fig. 6. Classification accuracy of user identification using parallel user data for different lookback windows (10s, 25s, 50s and 100s). The analysis is performed on per-task basis.

defined in Section 7.2. While iterating through the time steps, the training set grows and with the changing contexts we want to limit the amount of data taken into account in each iteration. Therefore we introduce a *n-lookback* time window, which only takes n last time steps of the data, instead of the whole available data from the beginning of each run, where $n = [10, 25, 50, 100]$.

7.1 Lookback Window Length

Without definitive knowledge on how often the context changes, we need to explore different lookback window sizes to capture the data pertaining to the currently relevant context and minimise the impact of previously active context on the identification accuracy. We take all users' runs (first or second) of a given task and perform user classification of every time step along the way, using previous n seconds of the data as the training set[8].

Results presented in Figure 6 indicate that there are no significant differences in classification accuracy when different lookback window sizes are used. Overall, the achieved accuracy is very high, ranging from 97.3% in Task 1 with the window size set to 100 seconds, up to 99.3% in Task 3 with the same window size. With marginal differences achieved by varying the lookback window size, we propose taking only the previous ten steps of the data, as taking a larger lookback window leads to a longer computational time without any clear classification accuracy benefits. Furthermore, our investigation shows that taking a ten-second lookback window remains sufficient for real-time retraining and re-executing the model within the one-second time step, hence enabling the final authenticator to authenticate the user once every second.

[8]It should be noted that we focus on a task-level classification, thus we do not attempt to re-align time sequences from different users, i.e. a larger variety of task execution speed will lead to a larger dissimilarity among users in the training set.

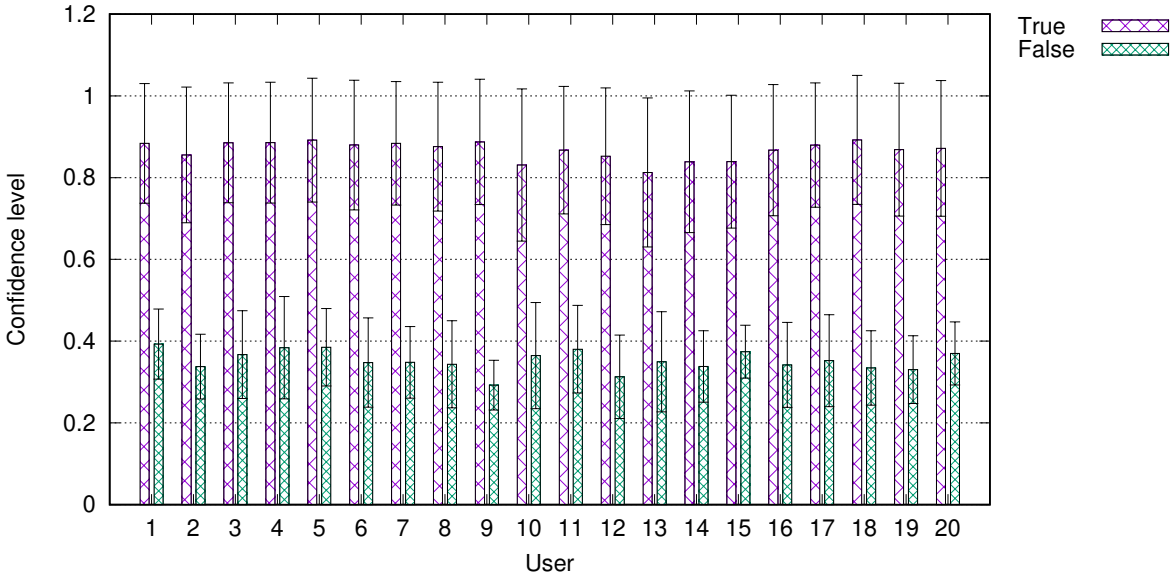

Fig. 7. Per user confidence levels for true (current user) and false (another user) class. Error bars represent the standard deviation of the confidence levels.

7.2 Continuous Authentication Confidence Levels

The above analysis confirms that an individual's within a task behaviour is consistent and that our IoT sensing-based method provides a reliable tool for continuous authentication of a previously identified user. But can we augment it with the ability to detect a session takeover by an illegitimate user? To provide a basis for such a detection and to further improve the performance of CA, similarly to the approach taken in Section 6.4 we harness the discrepancy between the classification confidence levels when either the true or the false class is predicted.

We calculate the confidence level as the mean predicted class probabilities, where the class probability of a single tree is fraction of samples of the class in a leaf, as we did in Section 6.4. This represents a simplified, but less time demanding version of the quantile regression forests [38], that can be calculated within our desirable time to authenticate. This class can either be *true*, i.e. correspond to the actual user, or *false*, i.e. correspond to a different user. In Figure 7 we plot the mean confidence levels of all tasks for each user when either the true or the false class was predicted, with the error bars representing standard deviation. Comparing to the confidence levels achieved during the standalone identification (Section 6.4), we observe even more striking differences between true and false confidence levels. With an average true confidence level at 86.9% and an average false confidence level at 35.2%, and with a fair amount of consistency between users, we hypothesise that taking the confidence level into account when deciding on user authentication we would be able to construct a robust session takeover detector.

7.3 Session Takeover Detection

Our ultimate goal is to devise a system that continuously confirms the identity of the intended user and reliably and rapidly detects a potential security breach. The breach model we are focused on is relevant for a number of shared spaces, such as those found in hospitals, factory control rooms, or reception desks. A breach may occur after a legitimate user authenticates herself using a standard means of authentication, but fails to properly

de-authenticate before leaving the monitored environment. This presents an opportunity for a malicious user to access the system impersonating a legitimate user. The details of the threat model are presented in Section 3.

Using the data of both runs of Task 1[9] in the collected dataset we emulate the session takeover attack and develop a detection mechanism that relies on our continuous authentication method and classifier confidence metric. Taking all different pairs of participants, we designate one as the victim and the other as the attacker and vice-versa. All remaining participants are treated as other legitimate users of the system. We then combine the first half of the task run data of the victim and the second half of the task run data of the attacker and merge them into a single "user", simulating a security breach roughly halfway into the task run. In summary the dataset that we use is a merger of the victim-attacker data and other participants' data. Of course, the original (non-modified) victims' and the attackers' task runs are not present in this dataset. The data from time segments that display a low enough confidence, based on the given threshold, is discarded and does not become the part of future random forest models. We include this mechanism to combat idle times and abrupt (but only momentary) behaviour discrepancies.

Our detection method is based on the approach explained in the introduction of this section and relies on the *n-lookback* window analysis from Section 7.1 and the confidence level analysis from Section 7.2. We expect that in the moment of the attack i.e. the first second of data originating from a malicious user, the confidence level will fall significantly and stay low as we already established that there is a significant discrepancy between the confidence level of the classifier when the true and the false class is predicted. From the preliminary analysis (Figure 7), it follows that the confidence level of the false class rarely goes above certain values. Therefore, we base our approach on appropriate thresholding of the classification confidence. To fine-tune the confidence level threshold, which serves as a segregation line between a legitimate and a malicious user, we use the first task runs of the users to determine the threshold value (in a sense "threshold training"). We experiment with values between 0.45 and 0.7 in 0.00625-wide steps. After setting the threshold, we use the second task runs to obtain the final results as per the previously described methodology.

We evaluate our approach along four relevant metrics, which have been used in the related work before [35]. The False Rejection Rate (FRR) [33] represents the number of false negatives (a legitimate user considered malicious), compared to the number of all legitimate user checks. The False Acceptance Rate (FAR) represents the number of false positives (a malicious user considered legitimate), compared to the number of all malicious user checks. As there is a trade-off between these two metrics, we also use the Equal Error Rate (EER) in order to provide a one-off metric that is comparable to related work. For numerous practical purposes, however, the most important issue is how reliably and how soon is a malicious user logged out of the system once an intrusion is detected? This can be gauged from the balance between the FRR and the fourth relevant metric we calculate – the time delay between the attack beginning and its detection.

We plot the FRR and FAR metrics for different lookback windows lengths and different confidence level thresholds in Figure 8. We observe an increase in FRR and a decline in FAR values with the increasing confidence level threshold. This is expected as more and more examples are classified as non-legitimate with a higher threshold values. The lowest achieved EER is with a lookback value of 20 and a threshold of 0.58 at 7.9%, followed closely by lookback values of 40 and 10 with and EER of 8.3% (threshold 0.6) and 8.5% (threshold 0.57). The smallest lookback value of 5 is trailing behind with an EER of 10.1% (threshold 0.56). Furthermore we plot the attack detection delay in Figure 9. Observing the different lookback values we are able to conclude that at a given confidence level threshold a higher lookback value exhibits a longer delay. This is as expected due to the greater momentum (number of past values of legitimate users we still take into account) of a higher lookback value. With an increasing threshold those differences diminish and the overall detection delay is shortened.

[9]First task runs as the threshold training and the second ones as the validation.

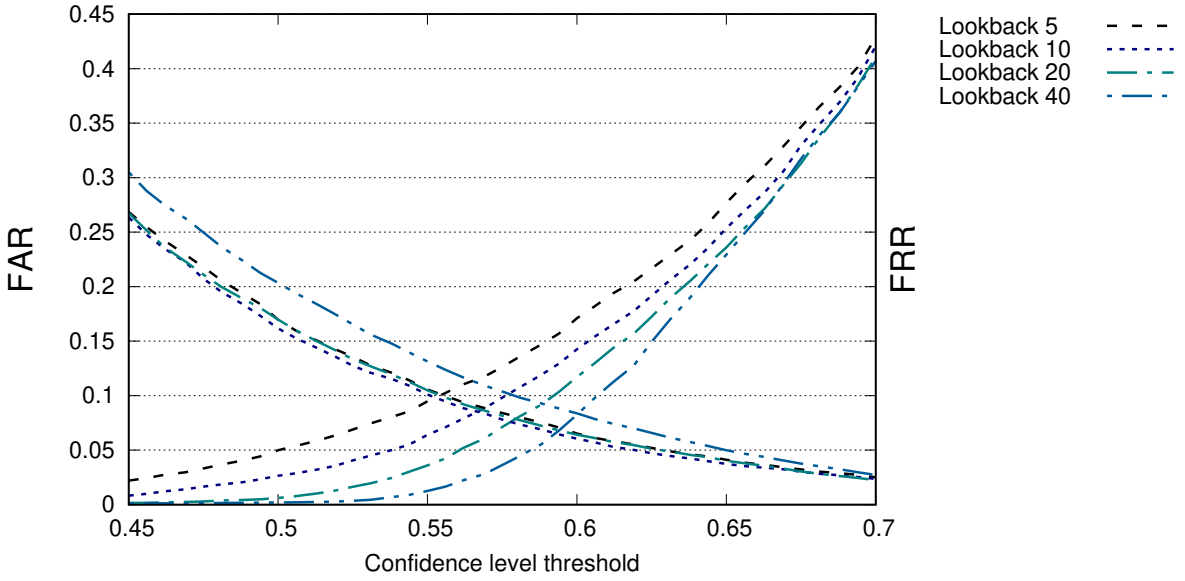

Fig. 8. False rejection and false acceptance rate for a given lookback and confidence level threshold values. Lines of the same colour (type) present the same lookback value with the ascending lines displaying the FRR and the descending the FAR values, respectively.

Concentrating on the end result, the general trend in the data points out that the increase in the lookback window length leads to a higher detection latency, but brings a lower FRR. The increased accuracy of the system with a longer lookback window might stem from the noisy nature of sensor measurements and occasional “glitches" in user behaviour; the impact of both of these effects is minimised if the data is averaged over a longer time window. From the practical point of view, the choice of the confidence level threshold should be dependent on the context of the system usage. If reliable operation is crucial, taking a longer lookback window allows setting a threshold that keeps the FRR virtually at zero. In such a case, our approach needs about fifty seconds to detect an intruder. On the contrary, in delicate environments, where security is of utmost importance, a 5-second lookback window and 0.35 confidence level threshold would successfully detect an intruder after only ten seconds, with the FRR remaining below 1%.

Finally, we apply both of the proposed parameter pairs (the lookback window of 40 and 5 seconds, with the confidence level threshold of 0.375 and 0.35, respectively) to the second task runs of Task 1. With the longer lookback window and the higher threshold we achieve a FRR of 0.2% with an intrusion detection delay of 44 seconds. The second parameter pair yields a FRR of 1% while detecting the intrusion after only 12 seconds. The similar results to those achieved during the analysis of the first run (above), indicates that the selected parameter values generalise over different task runs.

8 DISCUSSION

This paper provides an alternative to a common assumption that authentication has to rely on sensitive data, such as biometrics, or that it requires access to a user’s personal device, such as a smartphone [32]. Other solutions that harness the environment for authentication rely either on network traffic analysis [43], which limits their applicability to situations in which a user is actively using the network, or on wireless signal analysis [23], which

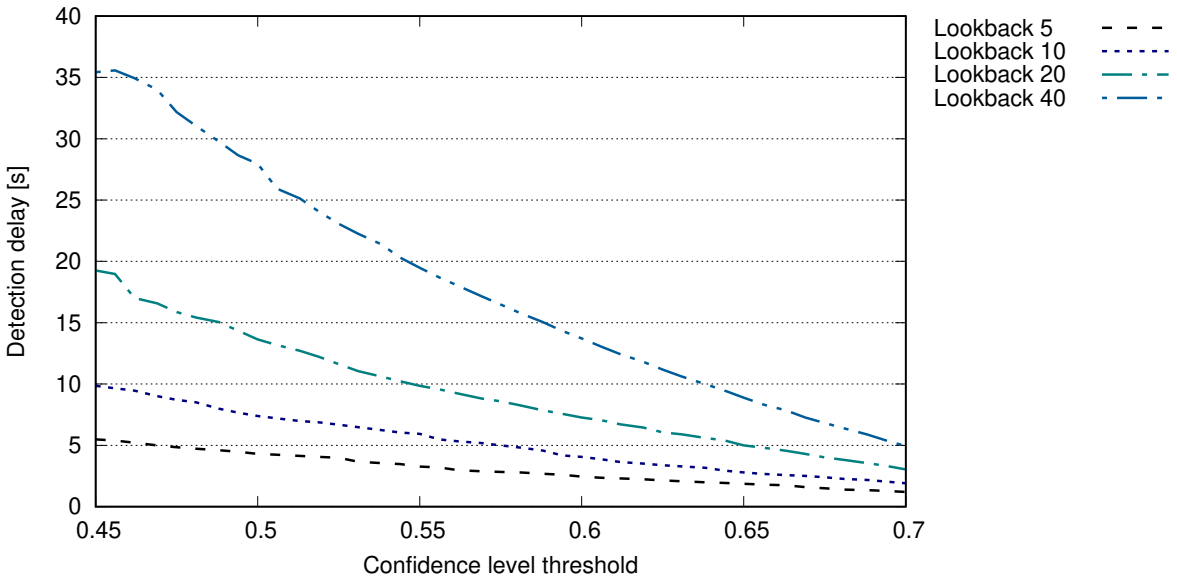

Fig. 9. Average time delay (in seconds) between the start of the attack and the attack detection for different lookback windows and confidence level thresholds.

is limited to detecting a user's physical activity. Our work, on the other hand, uses multimodal non-sensitive data sensing – we investigate the informativeness of accelerometer, gyroscope, force sensor, PC usage indicators, PIR, and hall sensor sampling. Our analysis reveals that the "closer" the sensor is to the object of interaction (e.g. a keyboard), the more discriminative the sampled data is for authentication. Thus, in our testbed, sensors placed on the desktop, such as the accelerometer, gyroscope, and force sensors alone are sufficient for models with above 60% identification accuracy. The key issue limiting the discriminating power of IoT-sensed data for the purpose of user identification is the volatility of human behaviour. We observe that the same user sometimes performs the same task in a drastically different manner, rendering identification approaches that rely on modelling upon the historically collected data unusable. Yet, a deeper analysis of the collected data reveals that intrinsic short-term behavioural patterns get captured by IoT-sensed data. Consequently, in our work we develop an online-updateable continuous authentication method that reliably confirms the identity of a previously authenticated user.

8.1 Performance Evaluation and Comparison to Other Approaches

A direct comparison between the method presented in this paper and the existing approaches for behaviour-based authentication is not straightforward due to the difference in the utilised datasets, including the number of users, the length of the experiments, and the modalities of the collected sensor data, but also due to the applications for which the proposed methods were envisioned. Unlike most of the related work, our authentication system is tailored to shared devices rather than personal devices – the continuous authentication mechanism we present in this paper does not require a specific device, nor a specific device type, to be carried by a user. Nevertheless, in order to identify the most promising avenues for future work, in the remainder of this subsection we compare the standard evaluation metrics, such as EER, FAR, and FRR of different approaches.

The approach based on keystroke and mouse dynamics presented in [41] achieves an identification accuracy rate of 62.2% in a multi-modal scenario, albeit only based on two data sources. In our analysis of standalone

identification based on IoT sensor data presented in Section 6 we obtain 68% accuracy. We believe that additional modalities, such as force sensors placed on the desk, lead to this improvement. Another approach based exclusively on keystroke dynamics [8] achieves better results, with a FRR of 4% and a FAR of less than 0.01%. Our best FAR rate reaches 1.8% with the FRR of 49%. Nevertheless, the results must be considered in the light of quite a substantial privacy trade-off, as the authentication system presented in [8] requires access to the typing data of the users. This is a privacy implication that is unlikely to be acceptable in shared environments considered in our work.

Feng et al. [19] add a new dimension to keystroke authentication by also considering touch pressure as well, achieving EER of less than 10%, 20% or 5% depending on the classifier used. Our best EER of less than 8% makes our system comparable to this approach. Methods based on pure touch dynamics, such as [20], provide the best overall performance, with an achieved EER of 4%. This is perhaps expected, as touch screen interactions can be very sophisticated. Intuitively, the closer an authentication layer is related to the physical movement of the user, the more distinctive the authentication seems to be. However, the study was run on smartphones rather than on shared devices. It is unclear whether users would interact differently with a device they are less used to (e.g. a shared public terminal). It is also possible that, while touch screen interactions offer high levels of distinctness in the way users interact with the screen, different screen sizes and different, perhaps shorter, interactions would reduce the amount of freedom allowed to the user, potentially making different users look more similar in the eyes of the classifier. Nevertheless, touchscreen interaction, and gesture analysis in general, present a promising avenue for future research in the area of IoT-based authentication.

8.2 Implications for IoT Environments

Smart environments are often denounced as insecure [28]. In this paper we demonstrate that IoT sensing can actually be used to *boost* the overall security. The same sensors that would in any case be installed for other purposes (e.g. PIR sensors for controlling lights, hall sensors for detecting inventory changes) can actually be used for security enhancement. Further, in the presented work we demonstrate that the quality of inference increases with additional sensors. This has also been discussed in a recent survey of continuous authentication approaches, which concludes with the authors' call for multimodal sensing solutions [60]. The ramification could be an increased diversity of sensors in future smart spaces. In case the sensors are not already present, the environment can be equipped with IoT devices whose exclusive purpose is to provide support for continuous authentication. Naturally, such a setup incurs an additional cost. Yet, the commodification of IoT devices makes this cost rather low – the full setup we used in Section 4 costs less than $200 USD. The setup costs can be further reduced, if the software that we aim to release, and that we used in our experiments, can be applied to the environment at hand.

The multimodality of our approach also allows our system to be more resilient against mimicry attacks. While in a mono-modal continuous authentication system (be it keystroke-, touch-, or more in general biometrics-based) an attacker only needs to learn patterns of behaviour related to a well-defined subset of attributes (which can be more or less easy to replicate), in a multimodal system, such as the one presented in this paper, an attacker would have to replicate different sets of attributes in order to successfully perform a mimicry-based attack. The usage of different categories of attributes at the same point in time adds complexity to any adversarial approach based on the mirroring of user behaviour, both from a computational perspective in the case of devices built to break authentication systems, and from a training perspective in the additional effort to train adversaries to learn multi-dimensional patterns of behaviour [27, 50].

The approach presented in this paper can also have a tangible impact on human-computer interaction (HCI). One of the key HCI issues with authentication is frequent authentication prompts [22]. In the future we envision a system that optimises the amount of prompting by tracking a user's behaviour in an IoT environment and re-authenticating a user only at times when the CA mechanism's confidence drops below a certain threshold.

The issue of when to actually prompt a user for re-authentication is a complex one and should take into account not only the reliability of different authentication mechanisms, but also the security needs, as well as a user's availability for impromptu interaction [5].

We believe that our work, together with the publicly-released dataset, represents a starting point for future efforts on heterogeneous authentication approaches that merge IoT-sensed data, data sensed via personal devices (e.g. a user's smartwatch), and dedicated authentication mechanisms (e.g. fingerprint readers, passwords). Furthermore, our approach can be expanded to include novel wireless sensors, such as Google Soli, that enable micro gesture recognition even without physical placement of a sensor on the object, and represent a promising platform for unobtrusive non-sensitive data sensing [15]. Together, these sources define an intriguing yet-to-be-charted research space. Finally, we believe that our work provides ground for other efforts relying on IoT data for different purposes, such as for service personalisation.

8.3 Generalisability of Our Approach

Guided by the exploration of the boundaries of sensor-based identification in this work we focused on experimentation with tasks that are rather well specified. Such tasks minimise the discrepancy among individuals that may arise due to one's "own way" of performing a task and ensure a higher generalisability of the tested methods. While a looser task specification would likely incur a higher identification accuracy, our results demonstrate that sensor-based authentication could also be beneficial in more rigid environments, such as factories, where tasks might indeed be rather well defined. The level of detail in the task definition also has consequences on the performance of the CA method developed in Section 7: training on data collected in well defined scenarios would ensure that the classifier is built on the intrinsic behavioural traits, not on task execution deviations, and would therefore make it more difficult for an adversary to pose as a legitimate user even if the users can be observed.

The generalisability of the identification method developed in Section 6 is indeed limited by the available task-specific data. The classifiers considered therein require sensor data collected when the participants are performing a certain predefined task. Should users be instructed to perform a different task at the testing time, the classification would fail. However, the continuous authentication approach developed in Section 7 retrains a classifier at each time step leading to two important generalisability consequences. First, the variations that, as shown in Section 6, occur between subsequent task runs conducted by the same user do not impact the classification performance, since the model adapts to the current task run. Second, in the limit case where multiple users enact the same task in parallel the approach can be task oblivious, adapting in real time to any task exciting the given sensors in an environment.

In our future work we plan to explore the limits of cross-task generalisability of IoT sensor-based identification. Our first step in this direction will be a construction of an adversarial deep learning pipeline that automates feature extraction so that the generated features provide a good basis for user identification, yet render the task identification impossible (similarly to [34]).

In the initial study we focused on a single smart office environment. Nevertheless, the concepts we developed apply to other environments equipped with similar sensors. Therefore, we envision our approach being used in smart homes, for example, where smart speakers, currently relying on a user's voice only, could harness multimodal IoT environment sensing for improved security. Shared public spaces, such as libraries, could also benefit from the CA approach developed in this paper, as terminals would recognise that a user has not logged out, thus prevent misuse. In Section 6 we achieve relatively modest user identification rates, yet, tuned to err on the safe side, the approach could be used for augmenting other surveillance mechanisms. For instance, a surveillance camera could be triggered only when a previously unknown movement/user is detected, therefore reducing the amount of irrelevant content recorded. Finally, while the initial analysis focuses on a lone user in the space, we believe that the continuous authentication approach can simply be extended to recognise, even track,

multiple persons in the same room. The LDA analysis from Figure 2 demonstrates that individual behaviour within a single task run remains consistent and easily separable even in a low-dimensional space.

8.4 Limitations

The findings of our study should be considered in the light of the limitations of the acquired dataset. Our experiments were conducted with a relatively modest number of users – we have experimented with data originating from between 17 and 20 users, depending on the task type. In the related work datasets involving twice as many and more users can be found. However, these approaches aim to authenticate users from brief interactions (e.g. touchscreen gestures, typing short text). Feature spaces emerging from such interactions are narrow. The tasks conducted by our subjects, despite being well defined, are relatively long and complex. The resulting multimodal feature space is broad allowing for differences among users to naturally emerge. Therefore, the performance degradation with an increasing number of users, although probable, is unlikely to be as drastic as it would be if we were to rely on short monomodal tasks. This is hinted in Figure 4, yet we have already prepared a new testbed in a research institution with more than 2,000 employees where we plan to thoroughly evaluate our approach with a larger number of users and with additional tasks of different duration. Finally, we should note that the relative homogeneity of our users is not a limitation, but an advantage of our dataset, as it makes the authentication process more difficult.

Relying on a novel authentication method may introduce overlooked security issues. As IoT-based authentication support becomes more prominent we might expect new attacks that would render it less useful. Such attacks targeting different sensing modalities might include viruses that would generate CPU and network activity that mimic usage patterns generated by a legitimate user; similarly, an adversary with a physical access to the environment could place motors producing vibrations similar to the ones produced by a legitimate user typing. Preventing such attacks requires further consideration of an IoT system cyber- and physical- security. Having in mind the results of user identification in Section 6 we argue against using our proposed IoT-based sensing mechanism as a standalone solution for authentication. Instead, we propose using it to continuously assess the identity of a user that has previously been authenticated through a conventional means of authentication. This ensures that, in the worst case, even if all the sensors are compromised, the barrier for a successful attack remains at the same level as with a single traditional authentication mechanism, whereas in the case the system functions as expected it makes the attack drastically more challenging to realise.

9 CONCLUSIONS

In this work we conducted a non-sensitive data collection study in an IoT environment with up to twenty users. We assessed the potential of the collected data for user authentication. Our findings highlight the informativeness of data sources placed “close" to the points of user interaction. The random forest model we construct from the above sensor data is able to detect the right user with 68% accuracy in Task 1 of the test dataset consisting of previously unseen interaction sessions. Nevertheless, the IoT sensor data-based approach exhibits its true potential when used for continuous authentication. Our solution achieves 99.3% accuracy in confirming a pre-authenticated user’s identity, whereas a deception attack remains detectable with more than 99% accuracy. Finally, to facilitate further investigation of this topic, we publicly release all the collected data as well as our analysis and machine learning programming code which can be accessed at https://gitlab.fri.uni-lj.si/lrk/ca-iot.

ACKNOWLEDGMENTS

The authors would like to thank study participants for their time. The authors would also thank Prof. Petko Bogdanov for valuable advice about multidimensional data separability and Dr. Igor Nai-Fovino for his help with

problem formulation. This work was partly supported by the Slovenian Research Agency (ARRS) under grants P2-0098 and N2-0136.

REFERENCES

[1] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Kemal Akkaya. 2018. WACA: Wearable-assisted continuous authentication. *Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018*, 264–269. https://doi.org/10.1109/SPW.2018.00042 arXiv:1802.10417

[2] Gunes Acar, Marc Juarez, Nick Nikiforakis, Claudia Diaz, Seda Gürses, Frank Piessens, and Bart Preneel. 2013. FPDetective: Dusting the Web for Fingerprinters. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer Communications Security (CCS '13)*. Association for Computing Machinery, New York, NY, USA, 1129–1140. https://doi.org/10.1145/2508859.2516674

[3] Furkan Alaca and P. C. van Oorschot. 2016. Device Fingerprinting for Augmenting Web Authentication: Classification and Analysis of Methods. In *Proceedings of the 32nd Annual Conference on Computer Security Applications (ACSAC '16)*. Association for Computing Machinery, New York, NY, USA, 289–301. https://doi.org/10.1145/2991079.2991091

[4] Taqwa Ahmed Alhaj, Maheyzah Md Siraj, Anazida Zainal, Huwaida Tagelsir Elshoush, and Fatin Elhaj. 2016. Feature Selection Using Information Gain for Improved Structural-Based Alert Correlation. *PLOS ONE* 11, 11 (11 2016), 1–18. https://doi.org/10.1371/journal.pone.0166017

[5] Christoph Anderson, Isabel Hübener, Ann-Kathrin Seipp, Sandra Ohly, Klaus David, and Veljko Pejovic. 2018. A survey of attention management systems in ubiquitous computing environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (2018), 1–27.

[6] Adam J. Aviv, Benjamin Sapp, Matt Blaze, and Jonathan M. Smith. 2012. Practicality of Accelerometer Side Channels on Smartphones. In *Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC '12)*. Association for Computing Machinery, New York, NY, USA, 41–50. https://doi.org/10.1145/2420950.2420957

[7] Peter Baumann, Stefan Katzenbeisser, Martin Stopczynski, and Erik Tews. 2016. Disguised Chromium Browser: Robust Browser, Flash and Canvas Fingerprinting Protection. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society (WPES '16)*. Association for Computing Machinery, New York, NY, USA, 37–46. https://doi.org/10.1145/2994620.2994621

[8] Francesco Bergadano, Daniele Gunetti, and Claudia Picardi. 2002. User Authentication through Keystroke Dynamics. *ACM Trans. Inf. Syst. Secur.* 5, 4 (Nov. 2002), 367–397. https://doi.org/10.1145/581271.581272

[9] Daniel Buschek, Alexander De Luca, and Florian Alt. 2015. Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 1393–1402. https://doi.org/10.1145/2702123.2702252

[10] Tadeusz Caliński and Harabasz JA. 1974. A Dendrite Method for Cluster Analysis. *Communications in Statistics - Theory and Methods* 3 (01 1974), 1–27. https://doi.org/10.1080/03610927408827101

[11] Rasim Cekik and Alper Kursat Uysal. 2020. A Novel Filter Feature Selection Method Using Rough Set for Short Text Data. *Expert Systems with Applications* (2020), 113691. https://doi.org/10.1016/j.eswa.2020.113691

[12] Mario Parreño Centeno, Yu Guan, and Aad van Moorsel. 2018. Mobile Based Continuous Authentication Using Deep Features. In *Proceedings of the 2nd International Workshop on Embedded and Mobile Deep Learning (EMDL'18)*. Association for Computing Machinery, New York, NY, USA, 19–24. https://doi.org/10.1145/3212725.3212732

[13] Centrify. 2019. Privileged Access Management in the Modern Threatscape. https://www.centrify.com/resources/centrify-privileged-access-management-in-the-modern-threatscape-2019/

[14] Nathan Clarke, Fudong Li, and Steven Furnell. 2017. A novel privacy preserving user identification approach for network traffic. *Computers Security* 70 (2017), 335 – 350. https://doi.org/10.1016/j.cose.2017.06.012

[15] Klen Čopič Pucihar, Christian Sandor, Matjaž Kljun, Wolfgang Huerst, Alexander Plopski, Takafumi Taketomi, Hirokazu Kato, and Luis A Leiva. 2019. The missing interface: micro-gestures on augmented objects. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.

[16] Heather Crawford, Karen Renaud, and Tim Storer. 2013. A framework for continuous, transparent mobile device authentication. *Computers Security* 39 (2013), 127 – 136. https://doi.org/10.1016/j.cose.2013.05.005

[17] Ingo Deutschmann and Johan Lindholm. 2013. Behavioral biometrics for DARPA's Active Authentication program. *IEEE International Conference of the Biometrics Special Interest Group (BIOSIG'13)*, 1–8.

[18] Serge Egelman, Sakshi Jain, Rebecca S. Portnoff, Kerwell Liao, Sunny Consolvo, and David Wagner. 2014. Are You Ready to Lock?. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*. Association for Computing Machinery, New York, NY, USA, 750–761. https://doi.org/10.1145/2660267.2660273

[19] T. Feng, X. Zhao, B. Carbunar, and W. Shi. 2013. Continuous Mobile Authentication Using Virtual Key Typing Biometrics. In *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*. 1547–1552.

[20] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. 2013. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security* 8, 1 (2013), 136–148.

https://doi.org/10.1109/TIFS.2012.2225048 arXiv:1207.6231

[21] Yu Guan, Xingjie Wei, Chang Tsun Li, and Yosi Keller. 2014. People identification and tracking through fusion of facial and gait features. In *Biometric Authentication (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics))*, Vol. 8897. Springer, United States, 209–221. https://doi.org/10.1007/978-3-319-13386-7_17

[22] Marian Harbach, Emanuel Von Zezschwitz, Andreas Fichtner, Alexander De Luca, and Matthew Smith. 2014. It's a Hard Lock Life: A Field Study of Smartphone (Un)Locking Behavior and Risk Perception. In *Proceedings of the Tenth USENIX Conference on Usable Privacy and Security (SOUPS '14)*. USENIX Association, USA, 213–230.

[23] Anna Huang, Dong Wang, Run Zhao, and Qian Zhang. 2019. Au-Id: Automatic User Identification and Authentication through the Motions Captured from Sequential Human Activities Using RFID. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1–26. https://doi.org/10.1145/3328919

[24] Jiaju Huang, Daqing Hou, Stephanie Schuckers, Timothy Law, and Adam Sherwin. 2017. Benchmarking keystroke authentication algorithms. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)*. IEEE, 1–6.

[25] Felix Juefei-Xu, Chandrasekhar Bhagavatula, Aaron Jaech, Unni Prasad, and Marios Savvides. 2012. Gait-id on the move: Pace independent human identification using cell phone accelerometer dynamics. In *2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, 8–15.

[26] Christina Katsini, Marios Belk, Christos Fidas, Nikolaos Avouris, and George Samaras. 2016. Security and Usability in Knowledge-Based User Authentication: A Review. In *Proceedings of the 20th Pan-Hellenic Conference on Informatics (PCI '16)*. Association for Computing Machinery, New York, NY, USA, Article 63, 6 pages. https://doi.org/10.1145/3003733.3003764

[27] Hassan Khan, Aaron Atwater, and Urs Hengartner. 2014. A Comparative Evaluation of Implicit Authentication Schemes. In *Research in Attacks, Intrusions and Defenses*, Angelos Stavrou, Herbert Bos, and Georgios Portokalidis (Eds.). Springer International Publishing, Cham, 255–275.

[28] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. 2017. DDoS in the IoT: Mirai and other botnets. *Computer* 50, 7 (2017), 80–84.

[29] Ross Koppel, Sean W Smith, Jim Blythe, and Vijay H Kothari. 2015. Workarounds to computer access in healthcare organizations: you want my password or a dead patient? *ITCH* 15, 4 (2015), 215–220.

[30] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Phys. Rev. E* 69 (Jun 2004), 066138. Issue 6. https://doi.org/10.1103/PhysRevE.69.066138

[31] David M. Kristol. 2001. HTTP Cookies: Standards, Privacy, and Politics. *ACM Trans. Internet Technol.* 1, 2 (Nov. 2001), 151–198. https://doi.org/10.1145/502152.502153

[32] Fudong Li, Nathan Clarke, Maria Papadaki, and Paul Haskell-Dowland. 2013. Active authentication for mobile devices utilising behaviour profiling. *International Journal of Information Security* 13 (06 2013), 229–244. https://doi.org/10.1007/s10207-013-0209-6

[33] Yantao Li, Hailong Hu, Gang Zhou, and Shaojiang Deng. 2018. Sensor-based continuous authentication using cost-effective kernel ridge regression. *IEEE Access* 6 (2018), 32554–32565.

[34] Sicong Liu, Junzhao Du, Anshumali Shrivastava, and Lin Zhong. 2019. Privacy Adversarial Network: Representation Learning for Mobile Data Privacy. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 4 (2019), 1–18.

[35] U. Mahbub, J. Komulainen, D. Ferreira, and R. Chellappa. 2019. Continuous Authentication of Smartphones Based on Application Usage. *IEEE Transactions on Biometrics, Behavior, and Identity Science* 1, 3 (July 2019), 165–180. https://doi.org/10.1109/TBIOM.2019.2918307

[36] Apostolos Malatras, Dimitris Geneiatakis, and Ioannis Vakalis. 2017. On the efficiency of user identification: a system-based approach. *International Journal of Information Security* 16, 6 (Nov. 2017), 653–671. https://doi.org/10.1007/s10207-016-0340-2

[37] A. M. Martinez and A. C. Kak. 2001. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 2 (Feb 2001), 228–233. https://doi.org/10.1109/34.908974

[38] Nicolai Meinshausen. 2006. Quantile Regression Forests. *Journal of Machine Learning Research* (2006), 983–999. http://www.jmlr.org/papers/volume7/meinshausen06a/meinshausen06a.pdf

[39] Emiliano Miluzzo, Nicholas D Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B Eisenman, Xiao Zheng, and Andrew T Campbell. 2008. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*. 337–350.

[40] Emiliano Miluzzo, Alexander Varshavsky, Suhrid Balakrishnan, and Romit Roy Choudhury. 2012. Tapprints: Your Finger Taps Have Fingerprints. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*. Association for Computing Machinery, New York, NY, USA, 323–336. https://doi.org/10.1145/2307636.2307666

[41] Soumik Mondal and Patrick Bours. 2016. Combining keystroke and mouse dynamics for continuous user authentication and identification. In *2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*. IEEE, 1–8.

[42] Pardis Emami Naeini, Sruti Bhagavatula, Hana Habib, Martin Degeling, Lujo Bauer, Lorrie Faith Cranor, and Norman Sadeh. 2017. Privacy expectations and preferences in an IoT world. In *Thirteenth Symposium on Usable Privacy and Security ({SOUPS} 2017)*. 399–412.

[43] Talha Ongun, Oliver Spohngellert, Alina Oprea, Cristina Nita-Rotaru, Mihai Christodorescu, and Negin Salajegheh. 2019. The House That Knows You: User Authentication Based on IoT Data. *CoRR* abs/1908.00592 (2019). arXiv:1908.00592 http://arxiv.org/abs/1908.00592

[44] Vishal M. Patel, Rama Chellappa, Deepak Chandra, and Brandon Barbello. 2016. Continuous User Authentication on Mobile Devices: Recent progress and remaining challenges. *IEEE Signal Processing Magazine* (2016), 49–61. https://doi.org/10.1109/MSP.2016.2555335
[45] Soumen Roy, Utpal Roy, and DD Sinha. 2015. Distance Based Models of Keystroke Dynamics User Authentication. *International Journal of Advanced Engineering Research and Science (IJAERS)* (2015), 89–94.
[46] Soumen Roy, Devadatta Sinha, and Utpal Roy. 2017. User authentication: keystroke dynamics with soft biometric features. *Internet of Things (IoT): Technologies, Applications, Challenges and Solutions* (2017), 99.
[47] Jukka Ruohonen and Ville Leppänen. 2018. Invisible Pixels Are Dead, Long Live Invisible Pixels!. In *Proceedings of the 2018 Workshop on Privacy in the Electronic Society (WPES'18)*. Association for Computing Machinery, New York, NY, USA, 28–32. https://doi.org/10.1145/3267323.3268950
[48] Chao Shen, Shichao Pei, Zhenyu Yang, and Xiaohong Guan. 2015. Input extraction via motion-sensor behavior analysis on smartphones. *Computers Security* 53 (2015), 143 – 155. https://doi.org/10.1016/j.cose.2015.06.013
[49] Elaine Shi, Yuan Niu, Markus Jakobsson, and Richard Chow. 2010. Implicit authentication through learning user behavior. In *International Conference on Information Security*. Springer, 99–113.
[50] Babins Shrestha, Nitesh Saxena, Hien Thi Thu Truong, and N. Asokan. 2014. Drone to the Rescue: Relay-Resilient Authentication using Ambient Multi-sensing. In *Financial Cryptography and Data Security*, Nicolas Christin and Reihaneh Safavi-Naini (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 349–364.
[51] Zdenka Sitova, Jaroslav Sedenka, Qing Yang, Ge Peng, Gang Zhou, Paolo Gasti, and Kiran S. Balagani. 2015. HMOG: A New Biometric Modality for Continuous Authentication of Smartphone Users. *CoRR* abs/1501.01199 (2015). arXiv:1501.01199 http://arxiv.org/abs/1501.01199
[52] Yingbo Song, Malek Ben Salem, Shlomo Hershkop, and Salvatore J. Stolfo. 2013. System Level User Behavior Biometrics using Fisher Features and Gaussian Mixture Models. In *2013 IEEE Security and Privacy Workshops*. 52–59. https://doi.org/10.1109/SPW.2013.33
[53] Statista. 2019. Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025. https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/
[54] Shridatt Sugrim, Can Liu, and Janne Lindqvist. 2019. Recruit Until It Fails: Exploring Performance Limits for Identification Systems. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–26.
[55] Hoang Minh Thang, Vo Quang Viet, Nguyen Dinh Thuc, and Deokjai Choi. 2012. Gait identification using accelerometer on mobile phone. In *2012 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE, 344–348.
[56] Zhen Tu, Runtong Li, Yong Li, Gang Wang, Di Wu, Pan Hui, Li Su, and Depeng Jin. 2018. Your Apps Give You Away. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1–23. https://doi.org/10.1145/3264948
[57] Dwi Ana Ratna Wati and Dika Abadianto. 2017. Design of face detection and recognition system for smart home security application. In *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. 342–347. https://doi.org/10.1109/ICITISEE.2017.8285524
[58] Flynn Wolf, Ravi Kuber, and Adam J. Aviv. 2019. "Pretty Close to a Must-Have": Balancing Usability Desire and Security Concern in Biometric Adoption. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper 151. https://doi.org/10.1145/3290605.3300381
[59] Zhi Xu, Kun Bai, and Sencun Zhu. 2012. TapLogger: Inferring User Inputs on Smartphone Touchscreens Using on-Board Motion Sensors. In *Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks (WISEC '12)*. Association for Computing Machinery, New York, NY, USA, 113–124. https://doi.org/10.1145/2185448.2185465
[60] Roman V Yampolskiy and Venu Govindaraju. 2008. Behavioural biometrics: a survey and classification. *International Journal of Biometrics* 1, 1 (2008), 81–113.
[61] Yafang Yang, Bin Guo, Zhu Wang, Mingyang Li, Zhiwen Yu, and Xingshe Zhou. 2019. BehaveSense: Continuous authentication for security-sensitive mobile apps using behavioral biometrics. *Ad Hoc Networks* 84 (2019), 9 – 18. https://doi.org/10.1016/j.adhoc.2018.09.015
[62] Yinghui (Catherine) Yang. 2010. Web user behavioral profiling for user identification. *Decision Support Systems* 49 (2010), 261 – 271. https://doi.org/10.1016/j.dss.2010.03.001
[63] Xiang Zhang, Lina Yao, Kaixuan Chen, Xianzhi Wang, Quan Z. Sheng, and Tao Gu. 2017. DeepKey: An EEG and Gait Based Dual-Authentication System. *CoRR* abs/1706.01606 (2017). arXiv:1706.01606 http://arxiv.org/abs/1706.01606

A TEXT CONTENT USED IN TASK ONE

(1) **First run:** John von Neumann was a Hungarian-American mathematician, physicist, computer scientist and polymath. Von Neumann was generally regarded as the foremost mathematician of his time and said to be the last representative of the great mathematicians. A genius who was comfortable integrating both pure and applied sciences.

He was a pioneer of the application of operator theory to quantum mechanics in the development of functional analysis and a key figure in the development of game theory and the concepts of cellular automata, the universal constructor and the digital computer.

(2) **Second run:** Alan Mathison Turing was an English mathematician, computer scientist, logician, cryptanalyst, philosopher and theoretical biologist. Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a general-purpose computer.
Turing is widely considered to be the father of theoretical computer science and artificial intelligence. Despite these accomplishments, he was not fully recognised in his home country during his lifetime, due to his homosexuality, and because much of his work was covered by the Official Secrets Act.

B INSTRUCTIONS IN BOXES IN TASK THREE

B.1 Box X

- Turn on the lights,
- find the box, marked with an 'Y',
- follow the instructions in the box.

B.2 Box Y

- In the room, there are 4 boxes marked with a 'Z',
- one of those boxes contains further instructions,
- find the instructions and follow them.

B.3 Box Z

- Take the card with the data,
- go to the PC and turn it on,
- ignore (do not close) the black window that appears on the screen,
- create a graph with the data presented in the data card,
- send an email with a graph to the ——@——.—,
- return the data and instruction cards to any of the boxes, marked with a 'Z',
- turn off the computer,
- turn the lights off,
- leave the room.