UNIVERSITY OF CRETE
DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCES AND ENGINEERING

Mitigation of Cyber Attacks in Wearable Devices

by

Andrei Kazlouski

PhD Dissertation

Presented

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

Heraklion, May 2023

UNIVERSITY OF CRETE

DEPARTMENT OF COMPUTER SCIENCE

Mitigation of Cyber Attacks in Wearable Devices

PhD Dissertation Presented

by **Andrei Kazlouski**

in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy

APPROVED BY:

Author: Andrei Kazlouski

Supervisor: Prof. Evangelos P. Markatos, University of Crete, Greece

---

Committee Member: Prof. Elena Ferrari, University of Insubria, Italy

---

Committee Member: Prof. Barbara Carminati, University of Insubria, Italy

---

Committee Member: Prof. Yannis Tzitzikas, University of Crete, Greece

---

Committee Member: Prof. Paraskevi Fragopoulou, Hellenic Mediterranean University, Greece

---

Committee Member: Prof. Kostas Magoutis, University of Crete, Greece

---

Committee Member: Prof. Panagiota Fatourou, University of Crete, Greece

---

Department Chairman: Antonis Argyros, Professor, University of Crete

Heraklion, May 2023

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Evangelos Markatos, for his guidance, support, and encouragement throughout my thesis work. His insights and expertise have been invaluable in shaping my research and improving my writing. I would also like to thank the members of my thesis supervisory committee, Elena Ferrari and Barbara Carminati, for their insightful comments and helpful suggestions on my thesis.

I am particularly grateful to my co-author Thomas Marchioro, for assisting me in every possible way since the very first day. The experience of working with you and improving our skills together is invaluable, and I am deeply thankful for that.

I would like to express my gratitude to all my colleagues in the Distributed Computing Systems Lab and in particular to Harry Manifavas, Meltini Christodoulaki, and Christos Papachristos for their friendship, support, and collaboration during my thesis work.

Finally, I would like to express my gratitude to my family and in particular, my parents, who have always been my role models and a source of strength throughout my life. I am deeply grateful for their love, support, and sacrifices that have made this achievement possible.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813162. The content of this work reflects the views only of their author (s).

Abstract

The consumer wearable market has been growing relentlessly since 2013, reaching unprecedented sales numbers during the first two years of the COVID-19 pandemic. At present, hundreds of millions (potentially *billions*) of users worldwide use such devices to monitor their private lives twenty-four hours a day and seven days a week. However, the lightning spread of wearable device and commercial fitness trackers, in particular, has not been complemented by the adequate security and privacy protection of their ever-growing userbase.

In this dissertation, we investigate if average consumers of commercial wearable devices are at risk. More specifically, whether conventional usage of fitness trackers by regular people may lead to a significant loss of privacy. In particular, we explore 2 aspects of consumer wearables: the devices with the associated ecosystem, and the data they produce.

We demonstrate that private information on users of prominent consumer fitness trackers may be inferred, when the devices transmit the collected data to the permanent storage of their manufacturer. An adversary may obtain insights on how often users exercise and measure their heart rate, whether they have trouble sleeping, or if they are overweight. We proceed to study the third-party companies that are contacted by wearable devices as part of their functioning. We show that significant and sometimes deeply personal data may be transferred to these "unwanted" third parties without explicit consent from users.

We further establish that sharing data generated by wearable "as is" may lead to significant privacy exposure and even full re-identification of users. By possessing very limited amounts of wearable data, a competent adversary may learn insights on person's gender, weight, height, and even reconstruct a "wearable fingerprint" – a unique pattern of daily routine.

To combat the above threats, we suggest a methodology for blocking unwanted connections of wearables, severely limiting the possibilities for privacy leaks. We further present comprehensive guidelines for privacy-preserving release of wearable data by both regular users and data controllers, who aggregate such information into datasets.

We emphasize that all proposed defense mechanisms can be easily employed by regular users with limited technical expertise and do not require any additional equipment.

Supervisor: Professor Evangelos P. Markatos

Περίληψη

Η αγορά των έξυπνων ρολογιών, και των έξυπνων συσκευών γενικότερα, αυξάνεται αδιάκοπα, έχοντας φτάσει σε ανεπανάληπτους αριθμούς πωλήσεων κατά τα δύο πρώτα χρόνια της πανδημίας του COVID-19. Αυτή τη στιγμή, εκατομμύρια ή ακόμη και δισεκατομμύρια χρήστες παγκοσμίως χρησιμοποιούν τέτοιες συσκευές για να παρακολουθούν τις προσωπικές τους δραστηριότητες (π.χ. άθληση, ύπνο, περπάτημα, κλπ.) είκοσι τέσσερις ώρες το εικοσιτετράωρο και επτά μέρες την εβδομάδα. Ωστόσο, η γρήγορη διάδοση αυτών των συσκευών δεν συνοδεύεται απαραίτητα από επαρκή προστασία ασφαλείας και προστασία της ιδιωτικότητας της ραγδαία αυξανόμενης βάσης χρηστών τους.

Σε αυτή τη διατριβή, ερευνούμε εάν η χρήση αυτών των συσκευών μπορεί να οδηγήσει σε σημαντική απώλεια της ιδιωτικότητας. Ειδικότερα, εξετάζουμε δύο πτυχές αυτών των συσκευών: (1) τις συσκευές μέσα στο σχετικό οικοσύστημά τους σε συνδυασμό με το λογισμικό που χρησιμοποιούν, και (2) τα δεδομένα που παράγουν.

Δείχνουμε ότι ιδιωτικές πληροφορίες των χρηστών μπορούν να εξαχθούν, όταν οι συσκευές μεταδίδουν τα συλλεγόμενα δεδομένα στη μόνιμη αποθήκευση του κατασκευαστή τους. Τέτοιες πληροφορίες περιλαμβάνουν (1) πόσο συχνά ασκούνται οι χρήστες (2) ποιος είναι ο παλμός της καρδιάς τους, (3) εάν αντιμετωπίζουν προβλήματα ύπνου, (4) αν είναι υπέρβαροι, κλπ. Παράλληλα δείχνουμε ότι αυτές οι συσκευές επικοινωνούν με ιστοσελίδες που ανήκουν σε τρίτες οντότητες (όχι στον κατασκευαστή της συσκευής) και στέλνουν δεδομένα σε αυτές τις ιστοσελίδες.

Επιπλέον, δείχνουμε ότι η κοινοποίηση δεδομένων που δημιουργούνται από αυτές τις συσκευές, μπορεί να οδηγήσει σε σημαντική αποκάλυψη προσωπικών δεδομένων και ακόμη και πλήρη επαναταυτοποίηση των χρηστών. Κατέχοντας πολύ περιορισμένες ποσότητες δεδομένων από αυτές τις συσκευές, μια επιδέξια εχθρική πλευρά μπορεί να αποκτήσει πληροφορίες σχετικά με το φύλο, το βάρος και το ύψος του χρήστη και ακόμη και να ανακατασκευάσει ένα 'αποτύπωμα της συσκευής' – ένα μοναδικό μοτίβο καθημερινής ρουτίνας.

Για να αντιμετωπιστούν οι απειλές που αναφέρθηκαν, προτείνουμε μια μεθοδολογία για τον αποκλεισμό ανεπιθύμητων συνδέσεων αυτών των συσκευών, περιορίζοντας σοβαρά τις δυνατότητες διαρροής προσωπικών πληροφοριών. Παρουσιάζουμε επίσης οδηγίες για την διατήρηση της ιδιωτικότητας κατά τη δημοσίευση δεδομένων αυτών των συσκευών, τόσο από τους κανονικούς χρήστες όσο και από τους ελεγκτές δεδομένων, οι οποίοι συγκεντρώνουν αυτές τις πληροφορίες σε σύνολα δεδομένων. Επισημαίνουμε ότι όλοι οι προτεινόμενοι μηχανισμοί άμυνας μπορούν εύκολα να χρησιμοποιηθούν από κανονικούς χρήστες με περιορισμένες τεχνικές γνώσεις και δεν απαιτούν επιπλέον εξοπλισμό.

Επόπτης: Καθηγητής Ευάγγελος Π. Μαρκάτος

Contents

List of Publications

In this dissertation I refer to the following published works by their Roman numerals.

I Kazlouski, A., Marchioro, T., Manifavas, H. and Markatos, E.P., 2021, February. "I still See You! Inferring Fitness Data from Encrypted Traffic of Wearables." In *HEALTHINF* (pp. 369-376).

II Kazlouski, A., Marchioro, T., Manifavas, H. and Markatos, E., 2021, March. "Do partner apps offer the same level of privacy protection? The case of wearable applications." In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)* (pp. 648-653). IEEE.

III Kazlouski, A., Marchioro, T. and Markatos, E.P., 2022, November. "I just wanted to track my steps! Blocking unwanted traffic of Fitbit devices." In *Proceedings of 12th International Conference on the Internet of Things* (pp. 96-103).

IV Kazlouski, A., Marchioro, T. and Markatos, E.P., 2022. "What your Fitbit says about you: De-anonymizing users in lifelogging datasets." In *SECRYPT* (pp. 341-348).

V Marchioro, T., Kazlouski, A. and Markatos, E., 2022. "How to Publish Wearables' Data: Practical Guidelines to Protect User Privacy." *Studies in Health Technology and Informatics*, 294, (pp.949-950).

VI Yfantidou, S., Karagianni, C., Efstathiou, S., Vakali, A., Palotti, J., Panteleimon Giakatos, D., Marchioro, T., Kazlouski, A., Ferrari, E. and Girdzijauskas, S., 2022. "LifeSnaps, a 4-month multi-modal dataset capturing unobtrusive snapshots of our lives in the wild." *Scientific data*, 9(1):663, Oct 2022.

This work also discusses the following related publications that are not included in the main body of the thesis:

a Kazlouski, A., Marchioro, T., Manifavas, H. and Markatos, E., 2020. Do you know who is talking to your wearable smartband?. *Integrated Citizen Centered Digital Health and Social Care*, p.142.

b Marchioro, T., Kazlouski, A. and Markatos, E., 2021. "User Identification from Time Series of Fitness Data." In *SECRYPT* (pp. 806-811).

c Marchioro, T., Kazlouski, A. and Markatos, E., 2023. “Practical Crowdsourcing of Wearable IoT Data with Local Differential Privacy.” In *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation* (pp. 275-287).

Author's Contributions

Publication I: "I still See You! Inferring Fitness Data from Encrypted Traffic of Wearables."

All authors conceived and designed the study. I collected the data. I led the implementation of the experiments, assisted by Thomas Marchioro. Thomas Marchioro and I wrote the paper with the help of other co-authors. All authors reviewed and commented on the manuscript.

Publication II: "Do partner apps offer the same level of privacy protection? The case of wearable applications."

The methodology of the paper was designed by all authors. Thomas Marchioro and I performed the experiments. Thomas Marchioro and I led the writing of the article with the assistance and feedback of other co-authors.

Publication III: "I just wanted to track my steps! Blocking unwanted traffic of Fitbit devices."

Thomas Marchioro and I conceived the study and wrote the manuscript. I performed the data collection. I led the implementation of the experiments, assisted by Thomas Marchioro. All authors read and approved the final manuscript.

Publication IV: "What your Fitbit says about you: De-anonymizing users in lifelogging datasets."

All authors conceived the study. Both presented de-anonymization attacks were designed and analyzed by Thomas Marchioro and I. Thomas Marchioro and I implemented the experiments. Thomas Marchioro and I wrote the article with the help of our co-author. All authors reviewed the paper.

Publication V: "How to Publish Wearables' Data: Practical Guidelines to Protect User Privacy."

Evangelos Markatos conceived and supervised the study. Thomas Marchioro and I designed the study and wrote the manuscript. All authors commented on and approved the submitted manuscript.

Publication VI: “LifeSnaps, a 4-month multi-modal dataset capturing unobtrusive snapshots of our lives in the wild.”

Sofia Yfantidou, Stefanos Efstathiou, Athena Vakali and Šarūnas Girdzijauskas conceived the study. Sofia Yfantidou and Stefanos Efstathiou designed the study, and Sofia Yfantidou conducted the study. Sofia Yfantidou, Christina Karagianni, Stefanos Efstathiou and Joao Palotti analyzed the results. Dimitrios Panteleimon Giakatos handled the database management. Thomas Marchioro, Sofia Yfantidou, Christina Karagianni, and I anonymized the data. All authors reviewed the paper.

List of Figures

List of Tables

Acronyms

BPM Beats Per Minute

BMI Body Mass Index

DNN Deep Neural Network

DP Differential Privacy

FTSN Fitness Tracking Social Networks

IoT Internet of Things

ISP Internet Service Provider

KDE Kernel Density Estimation

KNN K-Nearest Neighbors

LDP Local Differential Privacy

LSTM Long Short-term Memory networks

MAE Mean Absolute Error

ML Machine Learning

MITM Man in the Middle

NRMSE Normalized Root Mean Square Error

RF Random Forest

RMSE Root Mean Square Error

SVM Support Vector Machines

Chapter 1
Introduction

Technological advances have made wearable devices more sophisticated and accessible for the average consumers in recent years. Therefore, wearables market have been showing an unprecedented growth over the past decade, reaching a stunning total of 500 million units in 2021 [64, 135]. The most recent available data suggest that almost 150 million wearables have been sold in Q3 2022 [136]. In fact, the wearable market has never been in decline since 2013 [64], making it one of the most demanded electronic products readily available. The present surge of remote working, deteriorating lifestyles, and self-tracking are likely to maintain the interest in wearables in the foreseeable future. Pew Research Center has estimated that at least one in five Americans utilized a fitness tracker as of 2020 [19]. The up-to-date number is likely significantly higher due to the sharp increase for shipment of wearables during the recent pandemic. COVID-19 has become a driving force in sales of wearable devices, especially during the period of lockdowns and gym closures. Since the possibility to exercise had been severely restricted, users have been seeking alternative ways to exercise and keep active. Therefore, the highest number of wearables *ever sold* has been recorded in Q4 2020 – at the height of the pandemic – with more than 153.5 million devices shipped [136]. Furthermore, major providers of exercising apps have recorded unprecedented number of new users during COVID-19 [67, 139, 149, 159]. According to MoEngage, the number of downloads for health and fitness apps grew by 46% between Q1 and Q2 2020 [159]. In particular, one of the most popular fitness tracking application Strava has reported a 33% downloads increase during 2020, gaining 2 million new users each month. Given that the pandemic restrictions are still enforced in some parts of the world (as of Q1 2023), it is highly likely that the current trend will continue. Unfortunately, the proportion of privacy-aware individuals among the userbase of wearables remains extremely low [7]. The issue of protecting wearable devices is of paramount importance in light of the aforementioned insights.

Categories of wearables considered in this Thesis. Technically, wearable devices comprise a wide range of on-body devices, including smartbands and smart watches, earbuds

and headphones, smart jewelry, smart clothes, implantable devices, etc. In this dissertation, however, we mostly focus on wrist-worn fitness trackers. We center our research on such devices for a number of reasons:

- Consumer fitness trackers constitute a significant amount of the total shipment for wearables [136].
- Consumer-level fitness trackers collect much more data than any other category of wearables. Moreover, such information are often aggregated into datasets, which may become publicly available.
- Consumer fitness trackers can be paired with an ever-growing number of various fitness applications that communicate sensitive data over the Internet.
- Consumer-level activity trackers are vastly employed not only for activity/recreational purposes but also for medical and health studies.
- Most importantly, consumer fitness trackers are easily accessible and can be operated by regular users with limited technical/medical expertise.

Henceforth, we interchangeably employ various wearable-related terms, including wearable devices, wearables, consumer wearables, smartbands, wristbands, smart watches, fitness trackers, activity trackers or just trackers to specifically refer to the consumer-level wearable fitness trackers.

The wearables we consider in this thesis collect a vast amount of fitness information and other activity-related data. Such devices are able to track a wealth of various diverse fitness parameters, including steps, distance, calories, workouts, weight, heartbeat, sleep, etc. According to a recent consumer study [24], most users purchase wearables to monitor the above parameters. Note that such data are not measured directly by the trackers but instead calculated from the low-level sensor information. At present, commercial wearable trackers may be equipped with modern sensors, including accelerometer, gyroscope, GPS receiver, altimeter, as well as heartbeat, blood pressure, and skin temperature sensors. Occasionally, fitness trackers may contain ambient light and multi-purpose electrical conductance sensors as well. All the above sensors collect raw data that are being converted to the higher granularity metrics. Finally, the latest generations of wearables are able to monitor even more sophisticated attributes, such as stress, mood, anxiety, and emotional levels.

Naturally, given the penetration of consumer fitness trackers throughout the world, a number of privacy-related concerns have been raised in recent years [25, 61]. Despite the recent consumer surveys indicating that users of wearables are somewhat aware of the privacy risks [39, 118, 152], the vast majority of consumers tend to view them as mostly

hypothetical and rarely exert caution when using their devices. Indeed, why should regular individuals care if "they" (i.e., vendors or any other entities) learn some information on steps and calories of users? Unfortunately (for regular users), the possible information exposure goes way deeper. Indeed, consumer wearable devices not only monitor fitness-related parameters but also regularly synchronize with the mobile companion applications that, in turn, constantly communicate with the Internet (the communication pipeline for consumer wearables is depicted in Figure 2.1 and is discussed in more detail in the next sections). This, in turn, raises significant concerns in regards to *ubiquitous data collection* by wearables. This term has been widely accepted in association with mobile phones [127], which gather enormous volumes of high-granularity information. Indeed, modern cellphones are endowed with a wide-variety of sensors and broadcasting modules; they transmit Wi-Fi and Bluetooth signals, communicate with GPS satellites, connect to the cell towers, and gather wealth of accelerometer, gyroscope, and other sensor data. More importantly, however, mobile phones are constantly connected with the Internet and are practically spying on users with their implicit consent. It is evident that ubiquitous data collection may lead to mass surveillance and profiling of both individual users, as well as particular groups based on the collected information.

Given that consumer wearables are literally worn 24/7 and constantly collecting or monitoring sensitive information, which may be communicated over the Internet, such devices reiterate the problem of ubiquitous data collection. Moreover, since fitness trackers do not only collect arguably "more sensitive" data compared to mobile phones but also are constantly on-body, even when sleeping, they may be considered more privacy-unfriendly. Overall, we believe that consumer wearable devices do represent the second coming of ubiquitous data collection, and the above concerns are extremely relevant and should not be neglected.

Another aspect of consumer wearables that has been flagged by privacy activists is related to a so-called *quantified self* – the concept of self tracking using digital technology [87]. In principle, any device that enables users to track some elements of their lifestyles may be characterized as providing quantified self. At present, wearables are one the most wide-spread tools for self tracking, since they are able to collect and aggregate a wide range of various health metrics, as well as analyze other physiological and psychological metrics. Since these data may potentially be accessed by external entities, quantified self, in the context of wearables, have been considered as a valid privacy concerns. More specifically, wearable data of users are available not only to the manufacturers of the devices; some of the aggregated statistics or even unmodified samples may be shared with various third parties. Furthermore, users themselves tend to share their fitness snippets with the general public via Fitness Tracking Social Networks (FTSN) such as Strava and Fitbit. Regular users of wearables may be also recruited for various fitness studies, involving usage of wearables, where their activity data may be disclosed to the research community.

Nevertheless, relinquishing self-tracked data may not appear overly threatening to a regular user of wearables. For example, fair questions to ask would be:

> Why is it so bad if my daily step count and calories consumption are being shared? What can they realistically infer from my data? Surely, it is not possible to deduce any sensitive information from just my steps, calories, and daily distance, isn't it?

However, the recent research indicates that wearable data may contain a wealth of insights that can be extremely incriminating. Naturally, simply analyzing the daily activity patterns of users may indicate their routine, lifestyle, and fitness proficiency. Conversely, it may enable identification of routine irregularities, such as whether a user went out at night or did not go to work, etc. Moreover, previous articles have suggested that even more nuanced insights can be derived from wearable data, such as detecting unhealthy habits, diseases, pregnancy, and even the precise location of secret military bases [34, 73, 111, 147, 169]. In fact, in 2017, a provider of FTSN services, Strava, released a global heatmap of all GPS activity ever uploaded to their servers. The world map depicted the location-specific fitness trends using more than 3 trillion fitness routes. Military analysts were able to identify undisclosed facilities of the US Army in Afghanistan, Djibouti, and Syria by studying the territories where consistent usage of wearables seemed unlikely. Moreover, consumer wearable trackers, and Fitbit devices in particular, have been of utmost assistance to law enforcement in solving a number of homicides. By analyzing the activity data of victims and/or perpetrators, authorities were able to solve these crimes [106, 146, 158]. Finally, data produced by wearable devices have been widely used recently to infer COVID-related attributes of regular consumers. Such insights include predicting the pandemic trends [172] and detecting positive cases [6, 103, 116].

Nevertheless, it appears that the vast majority of wearable owners tend to critically underestimate the importance of the collected data. For example, Fitbit has its own internal FTSN, where thousands of individuals around the globe share their daily fitness snippets, disclosing their activity trends, workouts, and even quality of sleep data to the world. Such information may be utilized not only for profiling a single person (e.g., by stalkers and doxers), but also by various data harvesters, who may infer insights about specific demographic groups and minority individuals.

Furthermore, once the wearable data have been collected, users tend to eventually lose control over them [28]. In particular, they may be unable to execute several fundamental rights of the data subject granted by GDPR [27], e.g., the right to *be forgotten* (Art. 17-19) and *data portability* (Art. 20). In other words, it may be challenging for regular users of wearable devices to erase all traces of their data or obtain information that has been generated by their device and shared with the manufacturer, including low-level data collected directly by sensors.

Overall, at present millions of wearable consumers use their devices in the *out-of-the-box* mode without being aware for the extent of privacy risks. In this settings, regular users purchase their trackers, set them up according to the instructions, download the corresponding companion app (Figure 2.1), and never change the default settings. Henceforth, we may refer to such settings as the *off-the-shelf* mode for convenience. There is, thus, a need to raise awareness of potential privacy risks and suggest *simple* but effective ways to combat possible leaks, in order for regular consumers with limited technical expertise to utilize them.

1.1 Research Questions

This dissertation discusses several aspects of consumer wearables that motivate the included works. In particular, the research areas concerning the privacy of wearables are covered by the following primary research questions (RQs):

- *Attacks on consumer wearables* – RQ1: Are there any *practical* attacks that may compromise the security and privacy of average wearable users?
- *Privacy of consumer wearables* – RQ2: What are the privacy risks for regular consumers of wearable devices? How can they be mitigated?
- *Attacks on wearable data* – RQ3: Is it safe for regular users to share data generated by wearable devices with the research community or post it online?
- *Privacy-preserving release of wearable data* – RQ4: What can be done to mitigate privacy risks when sharing wearable data?

	Research area	Concern w.r.t. to wearables	
		Ubiquitous collection	Quantified self
Avg. User	Attacks on consumer wearables	Publication I	–
	Privacy of consumer wearables	Publications II, III	–
	Attacks on wearable data	–	Publication IV
	Privacy-preserving release of such data	–	Publications V, VI

Table 1.1: Summary of the problems studied in this thesis and their relation to the included publications. We emphasize that we focus on the perspective of an *average* user for all research questions and areas. The articles that have not been included in the main body of the thesis are also related to the studied research question (in particular, Publications a and b).

We depict the distribution of the published articles across the studied research areas in Table 1.1.

1.2 Contributions

Thesis Statement: In this dissertation, we empirically show that *out-of-the-box usage of consumer wearables is associated with significant and realistic privacy leaks*. We demonstrate that both *wearable devices* and *the data* they produce may be *successfully* attacked by a competent adversary. We propose several *defense strategies*, the effectiveness of which we prove empirically.

The central contributions of this thesis are as follows:

- **Attacks on consumer wearables**: The thesis demonstrates that it is possible to infer personal fitness data of users by analyzing encrypted communication traffic for companion applications of prominent vendors (Publication I).
- **Privacy of consumer wearables**: The thesis shows that popular companion applications and their partner apps share sensitive insights with various "unwanted" third-party entities. It also proposes and empirically attests an effective mechanism to mitigate these leaks (Publication II and III, respectively).
- **Attacks on wearable data**: The thesis identifies several novel attacks against wearable data that may lead to user de-anonymization and inference of sensitive undisclosed attributes (Publication IV).
- **Privacy-preserving release of wearable data**: The thesis summarizes the most effective ways to protect wearable data against sensitive inference (Publication V). We publicly release an anonymized wearable data collection that outperforms previous datasets in terms of privacy provided to the participants (Publication VI).

1.3 Outline

The rest of the dissertation is structured as follows. In Chapters 2-5, we address the research areas depicted in Table 1.1. Of these, Chapters 2-3 discuss the insufficient security and privacy of the prominent wearable devices and the associated companion applications. More specifically, Chapter 2 focuses on practical attack against wearable devices. Chapter 3 examines the exposure for sensitive information of regular users, with Section 3.1 investigating data sharing to unwanted third parties, and Section 3.2 introducing the mechanisms for preventing such leaks. In contrast, Chapters 4-5 address the possibilities for inferring undisclosed insights from data collected by consumer fitness trackers. Building on the aforementioned threats, Chapter 5 presents guidelines for protecting wearable data and discusses the fitness dataset we have publicly released to the research community. Finally, Chapter 6 summarizes our research, outlines the main contributions of this thesis, and details relevant directions for future work.

Chapter 2
Security of Wearables

In this chapter, we discuss how consumer-level wearable trackers operate (Section 2.1.1), and the ways to learn what data they share (Section 2.1.2), even when such data are encrypted. Finally, we present a novel practical attack on fitness trackers in Section 2.2. We mostly discuss Publication I, but Publications II, III are also mentioned.

2.1 Background and Tools

This section outlines the setup that was utilized to study the security and privacy of consumer wearable devices. As it is not feasible to directly observe the data that trackers send to the manufacturer's cloud, it is necessary to somehow intercept and analyze the traffic. The setups employed in Publications I-III are described in this section. Additionally, previous attack vectors on commercial wearables are also discussed.

2.1.1 Workflow of Wearable Devices

As this thesis focuses on the privacy and security threats associated with consumer-level wearables, it is important to first understand how these devices operate. *Regular users* of fitness trackers typically (i) purchase the device, (ii) download the companion application from an app store, (iii) create an account, (iv) pair the wearable with the app, and, finally, use the device. However, in practice, the data collected by the wearable do not solely belong to the user. The manufacturer of the device ultimately obtains all the collected information when the fitness tracker is synchronized. In this section, we describe the typical operation of consumer-level wearables and how they collect and store data. Understanding this process is crucial in identifying potential security and privacy risks that may arise. The full synchronization pipeline is as follows:

1. The wearable collects the data with the built-in sensors. Some of the data are aggregated into high-level metrics and activities, such as steps, heart rate, and distance, which can be instantly visible on the device screen. Other data are transferred further as is.

2. Both low- and high-level data are sent via Bluetooth to the companion application.

3. The data arrive to the companion application, which is a mobile app that enables most of the functionality for the device. It aggregates the low-level data into sophisticated metrics, such as sleep, workout, stress, etc., and provides a user interface (UI).

4. The companion application sends the data over the open Public Internet to the final destination, which is the permanent storage of the vendor. At this stage data are in *transit* and may traverse countries and even continents.

5. The data reach the vendor's servers, where they are stored and can be accessed by the user and the manufacturer of the tracker at any time. Furthermore, additional data preprocessing and insight inference may be done in the cloud.

Figure 2.1 depicts how commercial wearable trackers operate and handle the collected data. Note that there are ways to utilize some models of the trackers without sending any data to the cloud whatsoever. In particular, several models of the fitness trackers are compatible with non-official custom applications, such as GadgetBridge [42, 45]. These application process all the data locally and do not connect to the Internet. We will mention these "jailbreak" applications in the next chapters.

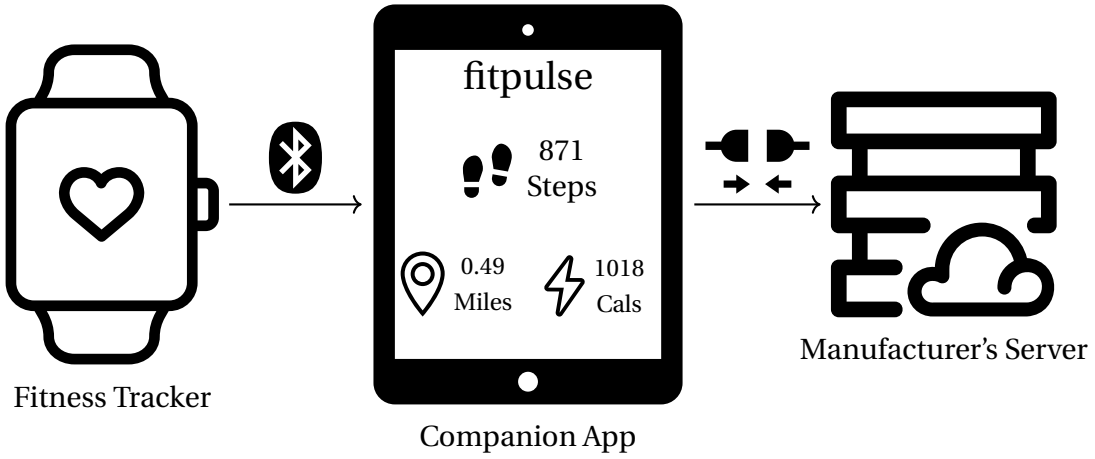

Figure 2.1: Workflow of wearable data in modern consumer-level fitness trackers. Data are being collected on the device which is connected to a companion mobile application via Bluetooth. The app processes and aggregates the data on the smartphone and dispatch them to the permanent storage of the manufacturer via the open Public Internet.

2.1.2 Analysis of Encrypted Traffic. Man in the Middle

According to Google Transparency Report [51], approximately 94% of the total Internet traffic is encrypted, adopting the HTTPS protocol. Therefore, it is reasonable to expect the communication between the smartphone and the cloud to be encrypted. In that case, although we can capture the IP packets, the payload of these packets will be encrypted. Now, unless we are able to decrypt the data, we will never be sure about the information being exchanged. One might naively think that capturing all IP packets will also result in intercepting the IP packets that contain the "keys" for the decryption of the encrypted traffic. However, this is not possible. The "keys" are themselves encoded and they can not be found. This is a fundamental property of the Public Key Infrastructure on the Internet and can not be broken with existent algorithms [101]. Such naive approaches to decrypting encrypted information just do not work. We need to find another way.

To decrypt the data being sent from the smartphone to the cloud, it is feasible to interpose a specific proxy known as Man in the Middle (MITM). Such a proxy (i) decrypts the traffic, (ii) examines the packet contents, (iii) re-encrypts the traffic, and (iv) finally sends the traffic to its destination. Step (ii) above enables inspection of packets to determine what exactly is being communicated between the smartphone and the cloud. Steps (iii) and (iv) are necessary to ensure that communication between the smartphone and the cloud happens uninterrupted.

Adding a MITM proxy for the above steps (i)-(iv) is easier said than done. Indeed, since it is not possible to find the decryption keys and decrypt the communication between the smartphone and the cloud, the only way to intercept the communication (and implement the above step (ii)) is for the MITM proxy to *convince* the smartphone, that it (i.e., the proxy) *is* the manufacturer's server. Indeed, when the smartphone wants to communicate with the cloud, the MITM proxy responds, "I am the cloud server – connect to me." When the smartphone receives this response, it will demand proof that the contacting entity (i.e., the MITM proxy) is indeed the cloud server – evidence in the form of a certificate signed by a trusted third party [112]. It is apparent that if no such proof is demanded, any computer on the Internet can impersonate any server on the Internet, which would lead to chaos. Obviously, a MITM proxy does not have such a certificate because no respectable trusted third party would issue a certificate accrediting that the proxy is not a proxy but a cloud server instead. Fortunately, if one has physical access to the smartphone, it is relatively trivial to add a special MITM certificate – known as certificate authority (CA) certificate – as a trusted certificate. Once this is done, the companion application will be convinced that the MITM proxy *is* a trusted third party that has issued a valid certificate, essentially making the proxy appear as the manufacturer's server! However, even if the companion application is convinced to treat the proxy as a cloud server, there still might be an obstacle to overcome: the wristband may employ a technique used to avoid MITM proxies (and

associated attacks) which is called SSL pinning [109]. SSL pinning essentially remembers (pins) the previous certificates (or encryption keys), and when it sees a new one, it wonders, "Why did the certificate of the cloud server change? Could this possibly be an attack?" Several tools can be used to disable validation of manually added certificates. In this thesis, we utilized a reverse engineering toolkit Frida [43], and the EdXposed framework[1]. Both solutions disable code responsible for certificate validation. Therefore, it is feasible to see the plaintext data that the companion app sends to the cloud. A possible MITM setup is illustrated in Figure 2.2. In our works, for MITM we utilize the Burp suite scanner [113]. To summarize, in order to analyze encrypted traffic of consumer-level wearables, the following two steps need to be undertaken: (i) installing a MITM proxy between the smartphone and the cloud server, and (ii) tricking the smartphone to believe that the MITM proxy is the cloud server.

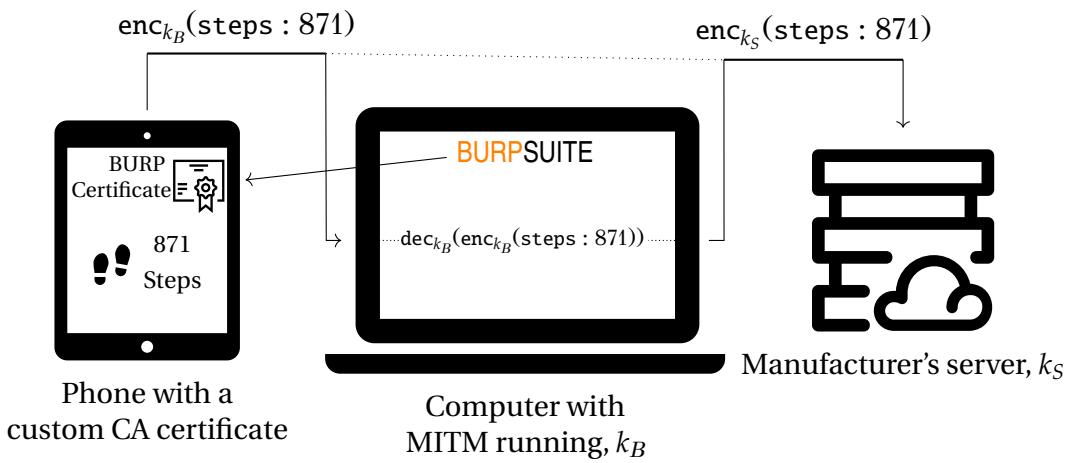

Figure 2.2: Man in the Middle (MITM) between the companion application and the vendor's servers. This setup enables analysis of the encrypted traffic, decoding all the fitness activities shared with the manufacture of the devices. If the companion app supports SSL certificate pinning, it needs to be disabled to validate the custom certificate.

2.1.3 Previous Attacks on Wearables

Since the penetration of consumer-level wearables has been steadily increasing, a considerable amount of research has been conducted on their security and privacy. In particular, a significant number of works have investigated various attack models against wearables, including:

[1] https://www.xda-developers.com/edxposed/

- Firmware modifications attacks [26, 88, 120, 128].
- Attacks on the Bluetooth communication of the devices [29, 52, 170].
- Utilizing wearables as a Side-Channel [85, 86, 90, 91, 125, 153, 154].
- Other specific attacks [14, 77].

While most of the above works have demonstrated that several prominent commercial devices may be vulnerable to the mentioned attacks, we argue that they might not be of great concern to regular users of fitness trackers. Firstly, the vast majority of the previous threat models require an adversary to be in close proximity to the device/user, thus, making the attack rather impractical. Furthermore, physical access to the tracker is often required, making the *malicious user* the only possible adversary for the attacks. Moreover, the previously described attacks are active, meaning that they will inevitably leave a trace. Therefore, most of the described threats tend to be short-lived as the vulnerabilities can be discovered and "patched" quite conveniently. Finally, such attacks may work only against specific manufacturers and models of wearable devices and are unable to target the vast majority of regular wearable users.

2.2 Attacks on Wearable Devices

Background. It has long been established that encrypted HTTPS traffic may leak information under several conditions [17, 21, 22, 30, 62, 82, 141]. Therefore, several previous works have studied the possibilities for the privacy leaks of the encrypted Internet of Things (IoT) communications [2, 5, 8, 12, 13, 63, 99, 102, 121, 122, 126, 130–132]. In these works, the adversary utilizes the size, frequency, order, and destination of the packets generated by IoT devices (other than consumer wearable trackers) to identify the device and its activities. We briefly provide several notable examples of the inferred IoT activities. In [12], the adversary was able to learn insights on users from the encrypted traffic of four IoT devices, including a camera, sleep monitor, and smart speaker. In particular, the authors were able to identify whether a user is in bed, and the security camera is recording footage. Furthermore, by monitoring the IP addresses of the destinations domains, they were able to distinguish the interactions with the smart speaker system. Acar et al. [2] studied the encrypted communications for a wide range of the IoT devices and showed that adversary may identify the devices and some of their actions with more than 90% probability. Some of the identified activities include live view of the camera, opening the smart door, turning lights off/on, and measuring weight. Alshehri et al. [8] also investigated the possibility of identifying the IoT devices based on the encrypted packets and were able to achieve 83% accuracy for a dataset of 14 units. They utilized the similar features, including packet size

and order. To our knowledge, however, we are the first to apply traffic analysis mechanisms to consumer-level fitness trackers.

As mentioned in the introduction, wearable devices collect personal and sometimes deeply confidential information. This includes a user's heart rate, sleep patterns, stress, and oxygen saturation levels, all of which contain important medical information that should remain confidential. Moreover, this information can be used to draw critical conclusions about the user's physical and mental health. For instance, knowing that a user has experienced severe sleep deprivation for several nights is alarming and sensitive information that should not be leaked to unauthorized third parties. Similarly, if a user measures their heart rate every five minutes, it is a sign of significant concern for both the user and their health.

Although such information leaks seem to be a problem that concerns only a small number of privacy-conscious individual users, it is not. It is a problem that concerns the entire society. Indeed, the problem it is not about whether a single person slept well at night or not. It is about monitoring millions of users without their knowledge, their clear understanding, or even their consent. As a result, it is important to understand what kind of information can these devices leak, who would be able to access this information, and how easy such leaks can be.

Although there are several places where this information can be leaked, including (i) the wristband used by the end user, (ii) the smartphone running the companion app, or even (iii) the cloud where the data are stored (see Figure 2.1), we believe the communication link between the smartphone and the cloud is the most vulnerable. Our choice is based on the following reasons:

- The devices themselves (i.e., the wearable, the smart phone, and the cloud storage of data) can be "hardened" by their manufacturer, making them more resilient to leaks and attacks [41, 138]. They can also be protected with traditional security defenses, such as antivirus systems, firewalls, and similar services that detect and mitigate cyberattacks [55, 140]. As a result, attacking the devices themselves is getting increasingly difficult, limiting the applicability of such a threat model.

- Data "at rest" in the wristband, in the smartphone, and even in the cloud storage, can be protected via strong encryption. Even if they are leaked or stolen, it will be very difficult to reverse the encryption and decrypt the original data [76, 80, 156]. Given the relevant recommendations of the European General Data Protection Regulation (GDPR), lots of providers move towards encrypting data "at rest" [27].

- Data "in transit" between the smartphone and the cloud are the most vulnerable: they traverse the open Internet, they may cross several ISPs, country boundaries, legal jurisdictions, and even continents. Any one of these entities (i.e., ISPs, countries, etc.) may have the motives and the technical capabilities to extract information from

these data. To make matters worse, end users of the wearable device have little, if any, choice over the decisions taken during these data transfers. Indeed, regular consumers generally have no say in which ISPs will carry their data (except possibly for the first one) and they also have no choice in which countries their data will cross over. To make matters worse, even if the user carefully monitors all ISP-related information, the dynamic nature of the Internet Protocol implies that data transfers may be carried over new and unexpected paths without asking any permission from the user.

As a result, data "in transit" are the most vulnerable as they cross potentially hostile territories with different or unfriendly legal jurisdictions. Specifically we focus on attacking regular end users of wearables who utilize their devices in the out-of-the-box mode.

2.2.1 Threat Model

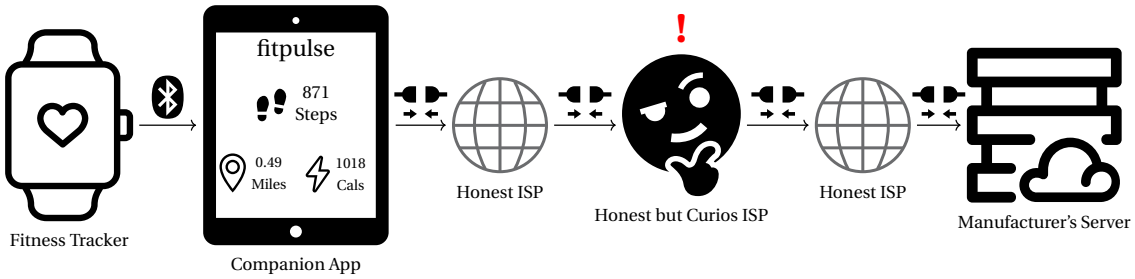

Figure 2.3: Threat model of a novel attack against consumer-level wearables. We assume that the device is paired to the companion application, which uploads activity data to the manufacturer's servers over the Public Internet, with the data traversing one or more ISPs along the way. We assume that at least one of the ISPs is *honest but curious*: it accepts and delivers IP packets (and the data they contain) to the manufacturer's cloud. At the same time, however, the ISP tries to infer as much insights as possible from the encrypted data. The ISP does never try to actively attack the user: it does not modify or delay the incoming IP packets.

We consider the following threat model for attacking consumer-level wearable devices: The device is connected to a mobile phone via Bluetooth and the smartphone is connected to the manufacturer's servers (where the data collected by wearables are permanently stored) via the open Public Internet. We assume that one of the ISPs who connects the user's mobile phone to the server aims to find information on the user. Such insights may include whether the user owns a smartband, how often the user exercises, and what the duration of the workouts is. This assumed ISP may be the first one that links the user to

the Internet, or even another ISP in the connection path between the user's mobile phone and the cloud. We assume this ISP to be "honest but curious." Note that such ISP is *not* trying to actively attack the user by manipulating the Internet traffic generated by the individual. Indeed, the attacker will not install a MITM proxy (Figure 2.2) or try to exploit any vulnerabilities in the mobile phone or the companion application. The ISP will *honestly* do its job: deliver the user's IP packets to their destination. At the same time, however, the ISP is expected to be "curious" and may try to find as much information as possible from the IP packets it was given to deliver to their destination.

Previous works [37, 38] in this area assumed that the ISP is malicious and may try to attack the user by installing a MITM proxy. This MITM proxy will actively try to break encryption and will attack the user in order to find all the information contained in the IP packets sent by this user to the cloud.

Although it is a valid threat model, we chose not to focus on it for two main reasons:

- This threat model is *short-lived*. Indeed, installing a MITM proxy to decrypt users' IP packets is illegal, similar to opening a user's mail. Therefore, actors who engage in this type of illegal attack can only do it for a short time, risking exposure, capture by Law Enforcement Agents, and shutdown of their activities.
- This threat model is very *difficult* to deploy on a large scale. In fact, most of the current MITM attack deployments require physical access to the user's smartphone for installing a forged certificate, which is usually difficult to achieve. Thus, while it can be used to monitor a small number of individuals, it cannot be used for mass surveillance of regular users.

On the contrary, the threat model we propose (i.e., an "honest but curious" ISP in the path from the user's smartphone to the cloud)

- is much easier to deploy (any ISP can do it), and
- is very difficult to discover, as ISPs do not actively engage in any hostile activities: they just passively collect information from the IP packets they are given to deliver to their eventual destination. They do not break any encryption, they do not engage with either the users, or with the cloud servers.

In Figure 2.3 we summarize the threat model we consider. We assume that the user has a wearable device which collects information. The device is connected to a smartphone app via Bluetooth. The smartphone is connected via Wi-Fi to a router and from there to an ISP (or directly via cellular Internet). As said, the ISP is assumed to be "honest but curious." That is, since it is honest, it will receive the IP packets, and it will transmit them promptly to their destination. However, since it is also "curious," the ISP will try to extract as much information about the user as possible from those IP packets, even if the data within the packets are encrypted.

2.2.2 Attack Description

Following the above threat model (Figure 2.3), the adversary aims to passively profile users of wearables based on the encrypted traffic it receives. To execute the proposed attack, the ISP follows the following 3-step pipeline:

1. Learning the *ground truth*.
2. Discovering leaks in the encrypted traffic.
3. Mass profiling of *regular* users for consumer wearables.

The first two phases are performed on a *local* isolated setup to prepare for the attack. The ISP then launches the third step *globally* by analyzing all the IP traffic it receives and navigates.

Obtaining a wearable. A malicious ISP begins by purchasing a wearable tracker (preferably one that is widely used) in order to determine whether it is vulnerable to the attack. Unlike attacks in other domains, without access to the device itself, it is not possible to simulate a regular user and send health data to the manufacturer's servers. The attacker's goal is to produce and send all possible activity values to the cloud and learn what traffic is generated during synchronization. The attacker then collects and analyzes the resulting dataset, which consists of various activity names and their corresponding encrypted traffic.

Discovering ground truth. The adversary begins by studying the data that are sent from the mobile phone before encryption. By synchronizing a single activity at a time, the adversary can isolate the encrypted traffic that corresponds to a specific activity. This traffic partitioning enables the ISP to identify patterns in encrypted traffic that indicate which activity data are being communicated. To analyze the traffic, the adversary installs a MITM proxy between the smartphone running the companion app and the manufacturer's server. Such proxy allows the adversary to observe all the data in plain text. To clarify, the adversary utilizes MITM only on a preliminary *local* setup; it does not attempt to intercept the actual traffic produced by regular users, unlike in [37, 38].

Identifying data leaks. Once the adversary learns what data are communicated to the server after synchronization, it can correlate them with the encrypted traffic they produce. More specifically, the attacker relies on the particularities of the HTTPS/TLS encryption: unlike hashing, the size of the output is not constrained. In other words encryption algorithms approximately retain the size of the input. To learn the ground truth for all possible fitness activities, the ISP synchronizes a single activity at a time, constructing all possible variation. The attacker tries to retrieve information about this activity based on the MITM data and the size of the corresponding encrypted packets. However, if the activity produces more than one encrypted packet, in addition to packet size, the attacker needs to consider packet order. For example, an adversary may record workouts of various duration and observe the difference between the sent packets.

Establishing mass profiling. The attack can be extended to *all* the traffic that the ISP operates once it identifies activities that are represented by specific sequences of encrypted packets. However, as mentioned previously, the attacker cannot utilize MITM on its *global* setup; it can only operate with the encrypted IP packets that need to be delivered to their final destinations. Say, the ISP has established that the *running* activity produces three packets of 2000, 3000, and 4000 bytes. Since major ISPs forward billions of encrypted packets from potentially millions of different IP addresses, just the sizes of packets may not be enough to accurately pinpoint the exact packets of interest. Therefore, the attacker needs to utilize additional features to increase the recall of the packets that do contain data generated by wearable trackers. In practice, the unique IPs assigned to the manufacturer's servers are one of the most "telling" features in wearable traffic analysis. By constantly synchronizing the companion application, the adversary eventually learns all the unique static IP addresses of the cloud storage. While TLS enables limited encryption of the URL addresses, there is no way to conceal IP addresses and prevent an honest but curious ISP from learning them during the first 2 stages of the attack. Indeed, the ISP needs to somehow send the packets towards the destination. Consistent with previous works on traffic analysis, in our threat model, the attacker utilizes the size, frequency, and the order of the packets, as well as the corresponding IP addresses. Overall, to launch a mass profiling of regular wearable users, the ISP needs to:

- Filter the incoming traffic by the list of IP addresses known to belong to the manufacturer's servers.
- Apply the metadata patterns (features) established in previous steps, including types of activities, size, order, and frequency of the packets.

2.2.3 Settings

Studied fitness trackers. We studied two of the most popular wearables readily available (as of 2020): *Xiaomi MiBand* 4 and *Samsung GearFit* 2. Both vendors are in the top 5 of wearable market share as of Q3 2022 [65, 136]. Henceforth, for convenience, we refer to the smartbands as MiBand and GearFit respectively. In the published version of Publication I, we do not use the real company and product names, and adopt pseudonyms instead to adhere with the responsible disclosure policies. Typically, responsible disclosure introduces a 90- to 120-day interval that allows the affected party to "patch" the vulnerability. However, since it has been more than 2 years since the publication, in this dissertation, we fully report our results. When submitting Publication I, in a responsible disclosure, we notified the affected companies about the vulnerabilities. We also had shared with them the corresponding sections of our manuscript draft version. Both companies responded that they were considering our submission.

Discovering ground truth. For MITM, we employ the application security testing software Burp Suite [113]. In Publication I, we utilize an open-source Frida toolkit [43] to disable certificate pinning. To analyze the encrypted traffic, we employ the network protocol analyzer Wireshark [161]. We evaluate the latest Android companion applications (as of June 2020) on a Google Nexus 6 phone.

2.3 Results

We begin by presenting the activity data that can be learned for both fitness trackers studied. The high-level insights that can be inferred by an honest but curious ISP are presented in Table 2.1. We proceed to explain our findings and data leaks in more detail in Section 2.3.1 for *MiBand* 4 and Section 2.3.2 for *GearFit* 2.

Activity	*MiBand* 4	*GearFit* 2
Measuring heart rate	Record	Record & extreme values
Measuring weight	Record	Record & extreme values
Workout	Record & duration	Record
Steps	-	Record
Sleep	Record	Record

Table 2.1: Insights on the activities of 2 popular wearable trackers that can be inferred by an honest but curious ISP from encrypted traffic only. The *Record* field indicates that the attacker can identify whether an activity has been performed since previous synchronization.

2.3.1 *Xiaomi* Wearable

The official companion mobile application for *Xiaomi MiBand* 4 is called *MiFit*. Being downloaded 50 million times (as of July 2020), it is one of the most used wearable app. *MiFit* utilizes a number of state-of-the-art security techniques, such as encryption of the outgoing wearable traffic, certificate pinning, and source code obfuscation. These mechanisms enhance the overall security and reduce the effectiveness of tampering with the devices/application.

Detected Activities

The *Xiaomi* smartband provides users with various functionality, such as measuring their heartbeat, counting their steps, tracking their sleep, recording their workouts, and receiving weather updates. To obtain sensitive information about a user, an ISP must identify activities from the corresponding encrypted traffic We define an *activity* as an action that a

Activity	Type	File(s)	Size of File(s)
Heart rate	Band	$H1$	$981 + 142 * K \pm 1$
		$H2$	16450 ± 50
Workout	Band	W1	≥ 1293
		W2	$S = 1725 \pm 25$
Sleep	Band	S1	8140 ± 40
Weight	App	We1	1182 ± 3

Table 2.2: Sizes of encrypted packets corresponding to activities of *MiFit*. Some activities result in multiple plaintext JSON files that are sent to the manufacturer's cloud. Size is measured in bytes. For heart rate detection, K represents the number of measurements done before synchronization.

person can perform using their wearable alone or in combination with a mobile application. We differentiate between *phone activities* initiated by the user from the companion app and *band activities* that are measurements collected by the wearable and then synchronized with the phone. It came to our attention that different activities correspond to different traffic patterns in the encrypted communication of the wearable (as expected). As a result, we set out to explore whether it is possible to "fingerprint" different services (and the associated user actions) by studying these patterns. To explore this possibility, we performed various actions related to the above services and collected the corresponding encrypted traffic to analyze the traffic patterns associated with different activities. We also observed that for most of the supported activities, *MiFit* encodes data in URL format before sending them. Each activity is represented by a specific sequence of packets that are almost the same length. The slight differences in the packet sizes for the same activities may be due to metadata that are sent with the wearable information. For example, the URL representation for the activity timestamp may differ by 1 character due to the 24-hour format (e.g., $6:00$ and $21:30$). This disparity is also reflected in the corresponding encrypted traffic, which would differ by 1 byte. For such activities, we introduce the range of potential payload sizes $\pm X$ bytes that accounts for such possibilities. We describe individual traffic sequences in greater detail in the following sections. The summary of the encrypted packets, corresponding to activities that can be recorded by *MiBand* 4, is depicted in Table 2.2.

In general, by analyzing the encrypted traffic, the attacker can determine the frequency and duration of a user's exercise and heart rate measurements, as well as whether they have recorded weight changes and slept since the last synchronization.

Listing 1 An example of H2(/v1/data/heartrate.json) for heart rate = 80 BPM. The 142-byte value that records a heart measurement is enclosed in square brackets. userid, time, device_id, and timezone are anonymized.

```
1 userid=<userid>&appid=428135909242707968&callid=1593161115220&channel=
2 play&country=US&cv=50309_4.1.1&device=android_23&device_type=android_
3 phone&heart_rate=[{"time":1***161075,"rate":"UA==","type":2,"device_id":
4 "<device_id>","source":25}]&lang=en_US&timezone=<timezone>
```

Heart Rate Detection

Measuring heart rate is deemed by many users as an essential reason for purchasing a wristwatch [24]. Monitoring one's heartbeat, and reacting to anomaly heart rates is an important step towards a healthy lifestyle. Although the precision of wearable devices in general may be somewhat lower than that of specialized medical blood pressure monitors [47], still these are highly sensitive data and should not be relinquished.

For *MiBand* 4, heart rate can be measured by pressing an associated button on the device or recorded automatically by the band, although this requires changing the default settings. The recordings of heart rate are stored in two JSON files: (i) *H1* and (ii) *H2*, which are sent to the cloud upon synchronization. Both files contain data in the URL encoding format. The *H1* file includes metadata such as the user ID, time since the last synchronization, and application ID, as well as the user's profile settings including their daily step goal, number of steps taken, and current number of calories burned. However, only *H2* contains the "numbers" of heart rate measured by the device. It contains all the heart rate measurements (could be more than one measurement) that were taken by a user since the preceding synchronization with the manufacturer's cloud. We noticed that a single heart rate measurement always results in *H2* of 1123 bytes. We have observed that if a user measures their heart rate twice, the application will send a $1123 + 142 = 1265$ bytes file. Similarly, if a user takes three heartbeat measurements before synchronization, the app will send a file that is $1123 + 2 \times 142 =$ bytes. In fact, each extra heart rate record increases the length of *H2* by 142 bytes.

For example, a heart rate measurement of 69 Beats Per Minute (BPM) would result in the following string been added to the JSON file: `',{"time":1591966729,"rate":"RQ==","type":2,"device_id":"16-digit-ID","source":25}'`, as depicted in Listing 1. Each value of heartbeat is represented by *four* characters (e.g., `RQ==` or `UA==`) regardless of the BPM value, which makes it BPM agnostic. We verified that the URL encoding of the produced sequence corresponds to a 142-byte string. A similar behavior is observed if the user takes four, five, etc., heart rate measurements. It appears that for each extra measurement, the

appended value is always 142 bytes. For example, if 3 measurements are taken before synchronizing, the packet would be appended by two 142 byte-long strings, etc. We took as many as 50 measurements before synchronization, and verified that the following formula always holds: $S = 981 + 142 \times K \pm 1$, where S is the size of the *H2* file and K is the number of heart rate records.

Listing 2 An example of W2(/v1/sport/summary.json) for a workout of 10 minutes. userid, deviceid, and timezone are anonymized.

```
max_stroke_speed=-1&trackid=1593163064&swolf=-1&type=16&left_landing_
time=-1&userid=<userid>&right_landing_time=-1&avg_stride_length=0&
avg_frequency=0&forefoot_ratio=-1&dis=0&avg_heart_rate=75&total_trips=-1
&distance_ascend=-1&total_step=0&landing_time=-1&right_flight_ratio=-1&
total_strokes=-1&climb_dis_descend=-1&run_time=600&avg_stroke_speed=-1&
avg_distance_per_stroke=-1&climb_dis_ascend_time=-1&climb_dis_descend_
time=-1&end_time=1593163665&altitude_ascend=0&version=12&swim_style=
-1&altitude_descend=0&swim_pool_length=-1&flight_ratio=-1&min_heart_rate=
-1&calorie=13&max_heart_rate=0&left_flight_ratio=-1&add_info=&appid=
428135909242707968&avg_pace=0.0&bind_device=0:MILI_CINCO_L:25:V0.25.17.5&
callid=1593164715329&channel=play&city=&country=US&cv=50309_4.1.1&device=
android_23&device_type=android_phone&deviceid=<deviceid>&lang=
en_US&location=&max_pace=0.0&min_pace=1.8000001&sn=dba7aab636e2&source=
run.25.huami.com&timezone=,<timezone>
```

Workout Duration Detection

MiBand 4 can record a number of various workouts, including walking, running, cycling, swimming, etc. Pressing the associated button on the device will invoke one of the above exercises. All types of workouts produce similar activity packets, making it impossible do distinguish them from each other. We established that a workout record is sent to the cloud via 2 JSON files: (i) *W1* and (ii) *W2*. *W2* contains statistical information on the workout, including the maximum reached speed, average stride length, minimum and maximum BPM during the exercise, and the number of burned calories. The structure of *W2* appears to be independent of the workout type, duration, and intensity. That is, the size of *W2* is always $S = 1725 \pm 25$ bytes. Listing 2 shows the summary file of a 10-minute workout recorded by *MiBand* 4.

W1, instead, describes the intensity of the workout at every given moment. More specifically, it encodes the "trace" of the user's heart rate during the workout, detailing every change in heartbeat. In Listing 3 an example of a heart rate trace generated during a 1-minute workout is depicted.

Listing 3 An example of W1(/v1/sport/run/detail.json) for a 1-minute workout. The heart rate trace grows linearly over time (starts with &heart_rate=). UserID and timezone are anonymized.

```
trackid=1591346638&userid=<userid>&version=12&accuracy=
&air_pressure_altitude=&altitude=&appid=428135909242707968
&callid=1591348077861&channel=play&country=US&cv=50300_4.1.0
&device=android_23&device_type=android_phone&distance=
&flag=&gait=&heart_rate=23,79;2,-1;,-1;,-1;2,-1;2,-1;,-1;6,-2;,
-1;3,-1;5,-1;4,-1;5,-1;5,0;3,2;,1;,-1;2,-1;5,0;2,1;3,1;,-1;
2,-1;2,1;3,1;5,0;3,-1;,-1;,-1;,-1;2,-1;5,0;4,1;4,1;,1;,-1;
3,-1;4,-1;,-1;3,-2;4,3;,1;5,2;,-1;2,-2;,-1;5,0;5,-2;6,0;,1;,
1;3,-2;5,0;5,0;5,0;6,0;5,2;5,0;5,0;5,0;4,-1;2,-1;2,2;4,1;3,1;
2,2;2,3;,1;,1;4,-1;5,0;4,-1;2,-1;2,1;,2;6,0;,1;,2;5,1;2,1;,-2;
2,-2;5,0;5,-2;3,1;5,0;2,-1;3,-1;5,1;2,-3;,-1;5,0&kilo_pace=
&lang=en_US&longitude_latitude=&mile_pace=&pace=&pause=
&provider=gaode&source=run.25.huami.com&speed=&stroke_speed=
&time=&timezone=<timezone>&v=2.0
```

In the above example, the trace record begins with 79; 2, −1, indicating that the starting heart rate is 79 BPM, followed by an increase of 2 ($79 + 2 = 81$), and then a drop to 80 ($81 + -1 = 80$), and so on. Since *W1* accounts for every change of user's heart rate, we assumed that the length of *W1* should be nearly linearly correlated with the *duration* of the workout. To verify this hypothesis, we manually performed a total of 21 workouts of various lengths, plotted the resulting sizes of payload, and attempted to fit the linear curve to the data points. According to Figure 2.4, the empirical data can be approximated by the equation: payload $= 1145 + 2.9 \times$ length, where payload is in bytes and length is measured in seconds. By analyzing the plaintext packets, we established that the actual size of a 0-second workout (without any hear beat change) is 1194 ± 1 bytes. Hence, the obtained empirical results appear to align with the ground truth for workout detection.

Unlike heart rate records, workout data are not transferred as a single JSON file. Instead, they are sent as corresponding pairs *W1* and *W2*. To detect when regular users synchronize workouts, the ISP needs to filter adjacent IP packets that fall within the size ranges for *W1* and *W2*. Similarly to heart rate, the workout activity is represented by multiple consecutive packets, which significantly increase the probability of successful detection. Therefore, an honest but curious ISP may be able to estimate the number and duration of workouts performed by the user since the previous synchronization. This information could reveal whether users are engaging in irregular workouts or following a specific training program. The above insights could be a basis for profiling users based on their activity routines.

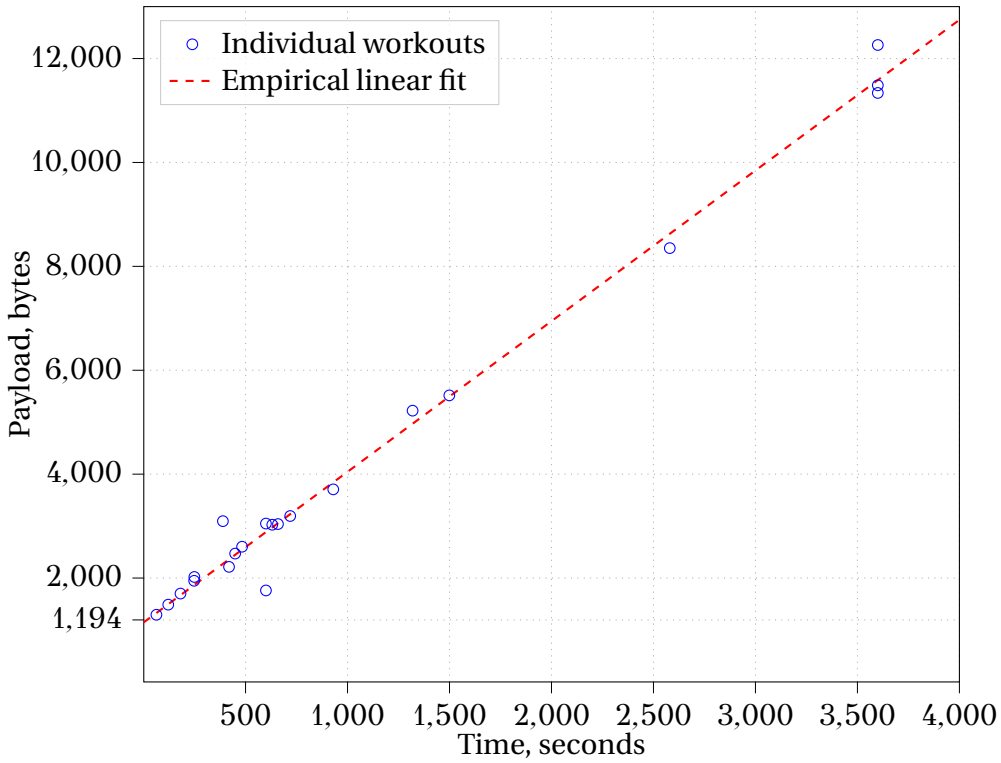

Figure 2.4: Size of the encrypted payload depending on the duration of the workout recorded by *MiBand* 4. Note that once the workout exceeds ≈ 100 minutes, the payload is divided into multiple packets, making it significantly harder to detect the workout activity. Such split occurs because the maximal payload size of a TLS packets is 16 KB, and, generally, after 100 minutes the payload exceeds this amount.

Sleep Tracking

Tracking sleep is another vital feature supported by *MiBand* 4, which appeals to many users [24]. The fitness tracker utilizes the built-in heart rate sensor to automatically detect sleep. For the wearable to start recording sleep, two conditions need to be satisfied: the user's heart rate must remain temporarily unchanged, and this must occur during night time (midday naps would not be recorded). Unlike other activities, sleep synchronization does not correspond to a file/files that are exclusive to sleep. Instead, *MiBand* 4 communicates the sleeping activity via the *S1* file, which also contains other device information such as firmware version, hardware version, battery level, etc. We noticed that the corresponding file appears to be transferred only if a user has slept since the previous synchronization. Our experiments suggest that the encrypted *S1* file is 8180 ± 2 bytes. Since the activity is represented a single file, the attacker needs to specifically find lone packets that are

around 8180 bytes. Naturally, there are other activities that can produce similarly sized packets (as indicated in the previous paragraphs). Therefore, the probability of correctly detecting sleep is expected to be lower than that of the previous activities. However, it works the other way around: if an ISP *does not see* any packets of the representative size since the preceding synchronization, it can safely assume that the user has not slept since the previous session with a very high probability. This information allows the attacker to learn if a victim has sleep troubles or was out of bed. Moreover, if an individual synchronizes their band frequently, the adversary might be able to recover a complete sleeping schedule of the user.

Step Count Tracking

MiBand 4 allows users to track their daily steps, which would be valuable information for our curious ISP. However, it appears that step data are sent in the *H2* file, which is sent practically every time the band is synchronized, even when the device has been idle. This makes it challenging for the attacker to pinpoint the exact *H2* file that carries the actual data, rendering it practically impossible for the ISP to obtain any insights regarding step count. The sleep detection approach cannot be employed here since the *H2* file is always present and sent during every synchronization, leaving no opportunity for absence-based detection.

Weight Tracking

Listing 4 An example of huami.health.scale.save.json (*We1*) for a 113 kg (250.0 lbs), 170 cm (5'6") person. UserID and timezone are anonymized.

```
1 userid=<userid>&devicetype=1&appid=428135909242707968&
2 callid=1593166280695&channel=play&country=US&cv=50309_4.1.1&device=
3 android_23&device_type=android_phone&jsondata=[{"fuid":-1,"wt":113.3975,
4 "ts":1593166246,"uid":"3061134679","dt":1,"ht":170,"wdt":2,"bmi":39.2,
5 "src":-1}]&lang=en_US&timezone=<timezone>
```

Users of *MiFit* can input/adjust their weight in the application. Since this activity can only be initiated from the application and not from the device itself, it is considered an "app activity." Our observations indicate that the corresponding *We1* packet is 1182 ± 2 bytes. Similarly to detection of sleep, recording weight produces only a single packet per activity, which increases the probability of false positives. Users who measure their weight frequently may be viewed as health-conscious individuals who prioritize tracking their physical fitness. Alternatively, they may be individuals with obsessive tendencies or body image concerns.

Such insight can be used for profiling by an adversarial ISP. An example of *We1* is illustrated in Listing 4.

2.3.2 *Samsung* Wearable

This section describes the results obtained for *Samsung GearFit* 2. The official mobile application for *GearFit* 2 is called *SamsungHealth*, which is a popular application for wearables available on Play Market. The app encrypts data using TLS, but unlike *MiFit*, it does not obfuscate the source code or use certificate pinning (as of July 2020). *GearFit* 2 has the same sensors as *MiBand* 4 (heart rate monitor, etc.), but allows users to input more of their fitness and health data. Moreover, unlike *MiBand* 4, the *GearFit* 2 uses regular Bluetooth instead of Bluetooth Low Energy (BLE). Similarly to the *Xiaomi* wearable, we focus on detecting heartbeat, sleep, number of steps, workouts. and weight changes.

Synchronization of Packets

In contrast to *MiFit*, FitB sends all the data in one go during synchronization, which can be initiated manually by the user or set to occur automatically. During synchronization, a total of *113* distinct files that represent activities in the JSON format are transmitted to the server. However, some of these files can be mapped to a single bigger activity. For example, 9 distinct files are related to food (e.g., a burger) that a user inputs into the app, including *total_bilirubin, total_cholesterol, total_protein, food_info, food_intake, food_favorite, food_frequent, food_goal, food_image*. By contrast, only a single *heart_rate* file is generated for the heart rate activity. Even if the user has not recorded any new data, all 113 files with the */changes* suffix (e.g., *heart_rate/changes*) will be sent to the server during synchronization. We discovered that for each of the files the app checks whether the value has changed (user did a corresponding action); if it is the case, it will send two files: (i) */set*, and (ii) */changes*. If the value has not changed, the app will only send a (ii) */changes* file. Hence, it is the */set* file that contains the actual data, which the adversary would like to obtain. If a user performs every possible activity before synchronizing, the app would send $2 * 113 = 226$ files at ones. If a user does nothing before synchronization, the app would send 113 files, all with the */changes* affix. Therefore, if an ISP observes a 119-file synchronization, it can conclude that user's activity led to the change of 6 files (113 */changes* and 6 */set*). Each file is carried by a single TLS data packet. All 113 */changes* packets can be easily detected, as their POST requests contain the default fields (URL, User-Agent, Content-Type, etc.) without any content. This translates into their encrypted payload being between 580 and 620 bytes long (depending on the URL length). Hence, packets that exceed this size are */set* packets that actually carry the personal data of users. Naturally, the ISP aims to find patterns in encrypted */set* packets to detect various activities.

Order of Synchronized Files

By studying the source code and observing MITM data, we were able to split 113 */changes* files in two "buckets." The first bucket consists of 10 files that are always sent before others: device_profile, step_count, step_daily_trend, activity.goal, tracker.pedometer_day_summary, goal_history, user_profile, exercise, food_info and tracker.pedometer_event. The remaining 103 JSON files represent the second bucket; they are sorted alphabetically before being sent. Files in the first bucket, however, are not sorted. Unfortunately for the attacker, the app does not always retain the same order when sending the files. Occasionally, it slightly (by $1-5$ positions) changes the arrangement of records. Since the attacker does not have access to the plaintext data, they cannot deterministically infer that the file #42 represents heartbeat. Instead the adversary has to consider the interval between #41-#46; any file in this range could potentially correspond to heart rate. In the previous paragraph we mention that if an activity was performed, the app sends two files: */changes* and */set*. It turns out that */set* files always precede the corresponding */changes* files by $1-5$ positions. These findings suggest that every encrypted */set* file can correspond to up to 10 different activities.

Detected Activities

Summarizing previous paragraphs, an adversary can intercept encrypted synchronization session containing between 113 and 226 files. Each JSON is represented by a single TLS data packet. All 113 */changes* packets can be easily detected because they are always empty. Their POST requests contain default fields (URL, User-Agent, Content-Type, Accept, etc.) without any content. This translates into their encrypted payload being between 580 and 620 bytes long (depends on the URL length). Hence, packets that exceed this interval are */set* packets that actually carry the personal data of users. Therefore, even without any patterns the ISP can guess with at least 10% probability which activity an encrypted */set* packet represents due to alphabetical sorting of the files before synchronization. For the attacker to successfully detect an activity two conditions need to be satisfied:

- The packet's TLS payload needs to be of a length corresponding to a particular activity (e.g., for heartbeat it is 1077 bytes).

- The packet needs to be within a sending order interval for this activity (e.g., for heart rate it is between #41 and #46 out of 113 JSON files).

Table 2.3 depicts studied activities and corresponding encrypted patterns. To summarize, the attacker is able to detect heart rate measuring frequency, and the instances that are above 100 BPM, workout frequency, whether the user slept, took steps, or recorded weight.

Activity	Files	Interval	Size of File(s)
Heart rate	H	42-47	$751 + (326 + \frac{1}{2}) \cdot K \pm \frac{K}{2}$
Workout	W	1-10	≥ 1720
	C	72-73	> 700
Sleep	S	111-113	998 or 1002
Steps	St	1-10	≥ 1216
	St2	1-10	> 700
	T	1-10	> 700
	C	72-73	> 700
	A	61-65	> 700
Weight	We	60-61	901 or 902

Table 2.3: Size of encrypted activity packets for *GearFit* 2. The column *Files* depicts what and how many plaintext JSON files are submitted to the cloud. *Interval* describes possible order of the packets during synchronization. *Size* is measured in bytes. For heart rate detection, K represents the number of measurements done before synchronization.

Heart Rate Detection

To measure heart rate using the wearable, users can press a representative button on the screen to send the heartbeat data to the server via the *heart_rate/set* (denoted as H) file. The H file is alphabetically located between positions $42 - 47$ and has a size of $1077 - 1078$ bytes. The 1 byte difference occurs because in real-world settings person's heart rate can be either a 2- or a 3-digit number. Multiple measurements of heartbeat are sent in a single file, and the size of the H file increases by $326 - 327$ bytes for every additional measurement before synchronizing the app. We verified that for an arbitrary K heartbeat measurements, the size of the H file lies within the $I = [751 + 326 \times K, \ldots, 751 + 326 \times K + K]$ bytes interval. Note that the size interval accounts for up to K measurements possibly being 3-digit heart rates. Hence, the adversarial ISP is able to detect both the number of measurements and the exact number of those over 100 BPM.

Workout Detection

Users of *GearFit* 2 have the option to select from 17 different workout types such as running, walking, cycling, and hiking, etc. The workout data are sent to the server in the form of a JSON file *exercise/set* called W, which is part of the first non-alphabetic bucket and can occupy positions $1 - 10$. This file includes information about the workout such as the description of the exercise and the trace of the user's heartbeat during the session. However, the detection technique from Section 2.3.1 cannot be applied to this file because multiple workouts are sent together in W, making it difficult to distinguish between a long workout

or multiple exercises. Despite this, the attacker can still estimate the minimal size of the file, with the shortest value obtained being 1720 bytes for a single workout that lasted only one second. The size of *W* has no upper limit, except for the 16 KB TLS limit. Therefore, detecting this activity is significantly more difficult for the adversary, and they can only infer whether a user did exercise since the last synchronization. The calories burned JSON file *calories_burned*, denoted as *C*, is also affected by the workout and occupies positions $72 - 73$.

Sleep Detection

GearFit 2 has an automatic sleep recording feature. However, users also have the option to manually edit or input their sleep data and provide a quality score. The sleep data is transmitted via the *sleep/set* (denoted as *S*) file which takes positions $111 - 113$. The size of the sleep file can either be 998 or 1002 bytes, depending on the user's rating of their sleep on a scale of 1 to 5. Detecting sleep activity appears to be relatively easy for the attacker since the encrypted payload can only be of two possible sizes. As a result, an adversary may be able to determine whether a user has slept since the last synchronization.

Retrieving Number of Steps

The *GearFit* 2 wearable automatically tracks the number of steps taken by users, and this information is recorded in the *step_count/set* (denoted as *St*) file that occupies positions $1 - 10$. When users take steps, it triggers the transmission of 4 other files to the server: *step_daily_trend* (#$1 - 10$), *tracker.pedometer_day_summary* (#1-10), *calories_burned* (#$72 - 73$), *activity.day_summary* (#$61 - 65$). The *St* file records all the steps taken since the last synchronization and displays them as intervals based on the user's speed. For instance, if a user takes 1000 steps at a normal pace and then runs another 1000 steps, the data will be recorded as two separate intervals. Therefore, similar to Section 2.3.2, there is no upper limit on the size of the *St* file (except 16 KB). To determine the minimum size of the file, we performed an experiment where we took a single step before synchronizing. In this case, the *St* file was 1216 bytes long. Given that taking steps triggers changes in four different files, the attacker can detect this activity with a high probability compared to other activities.

Weight Detection

The *SamsungHealth* application enables users to enter their weight, which is then transmitted to the cloud via the *weight/set* file (denoted as *We*) located in positions $60 - 61$. The size of the file can be either 901 or 902 bytes, depending on whether the weight entered is a 2- or 3-digit number in kilograms. This information can be exploited by an adversary to

identify users who weigh over 100 kg and may be classified as overweight.

2.4 Automatic Activity Detection

In this section, we propose an approach for automatic activity detection. Obviously, an "honest but curious" ISP would prefer to profile many users automatically rather than manually analyze each packet. The ISP handles vast amounts of IP traffic daily, including wearable data as well as millions of other packets from other users. It is impractical to store all the packets, and it is not crucial to detect activities "on-the-fly," since identifying health patterns of users requires multiple data synchronization sessions. The proposed pipeline for automatic activity detection is as follows:

- **Gathering all relevant IPs.** The attacker collects a list of IP addresses the wearables talks to. This is achieved by continuously initiating a connection with the smartband and recording corresponding IP addresses.
- **Traffic filtering.** The adversary filters traffic by the collected IP list, using an intrusion detection system, e.g., Snort [133].
- **Applying metadata rules.** The ISP applies the previously learned rules for detecting activities. For instance, TShark[2], a Python implementation of the network protocol analyzer Wireshark, can be used as it provides additional functionality for TLS processing.

Note that this approach can be transformed from a purely rule-based to a data driven method by changing the last step of the pipeline. In particular instead of relying purely on hand-crafted heuristics, the adversary may attempt to collect enough activity the data for a particular wearable (e.g., via parallelization), and train Machine Learning (ML) inference models. Given that a number of recent works on inferring insights from IoT traffic [2, 8] successfully utilized ML in their research, we believe a similar approach may be appropriate for consumer-level wearable trackers. We leave development of such an experiment for future work.

2.5 Applicability of the Attack and Countermeasures

TLS and wearables. There have been instances of side-channel leaks in the TLS encryption of various web-based systems. Although most of these attacks have only been successful in decreasing the entropy of the encrypted data, in our specific scenario, the attacker can derive significant insights on the activities of the users. In contrast to TLS leaks in

[2] https://www.wireshark.org/docs/man-pages/tshark.html

web services, where multiple requests can be sent in a single packet, our setup allows the adversary to transmit one activity at a time, which enables accurate identification of encrypted packets related to a specific activity. This capability allows the ISP to discover data leaks and gain sensitive insights on users

Possible countermeasures.

- *Modifying plain → cipher text size ratio.* It is possible to change the size of the encrypted payload by altering the plaintext JSON files. Since pruning those files may result in data corruption, the natural solution would be to selectively increase some of the transmitted files. This can be achieved by padding these files with "dummy" text until they reach a desired size. It should be noted that this approach may result in internet bandwidth loss for end users. However, if the amount of collectable data is not vast, the overhead should be negligible.

- *Concealing frequency of packets transmission.* To make it more challenging for an attacker to match pairs of plaintext with encrypted traffic, the app may introduce delays in sending user data to the server. Additionally, transmitting occasional meaningless or "fake" packets can further confuse the attacker. Similarly to the above approach, this security mechanism would "cost" additional bandwidth for the owners of the wearables.

- *Introducing randomness for order of packets.* Randomizing the order in which activities are sent to the server can make it more difficult for an adversary to map each activity to its encrypted counterpart. This adds an extra layer of complexity to the process and makes it harder for the attacker to identify patterns in the encrypted traffic.

Our suggestions are consistent with previous works in the area. We leave development of such a secure wearable traffic transmission system for future research.

2.6 What Can Regular Users Do?

As mentioned earlier, privacy activists have been developing so-called "jailbreak" apps that do not connect to the Internet whatsoever [42, 45]. Naturally, if no data are transmitted, there are no insights to be inferred. However, such apps may not support the full functionality of the original companion apps, potentially worsening the expected user experience. Furthermore, users lose the possibility of storing their data in the manufacturer's cloud; if they change/break their phone, all the valuable information is lost forever. Nevertheless, overly cautious users might still consider using a custom application over the official companion app if they own a supported device.

Another trick that does not require any additional installations or software manipulations involves synchronizing large amounts of wearable data at once. That is, if users do not synchronize their data every day, it may be significantly harder for an honest but curious ISP to infer specific activities among many encrypted packets. Even if the adversary, manages to recover all the fitness activities, the daily routine of users will likely be less reconstructable. We leave the identification of the optimal "synchronization delay" period for future work.

Chapter 3
Privacy of Wearables

This chapter discusses privacy leaks in consumer-level wearable devices. We identify connections of wearable apps that leak sensitive information in Section 3.1. Section 3.2 demonstrates the ways to reduce or completely prevent such leaks. While the discussion of this section is centered around Publications II and III, we also refer to another work of ours on the topic [68] (Publication a).

3.1 Unwanted Connections of Wearables

In our research, we mainly investigate the privacy leaks that occur when users utilize their consumer-level wearable fitness trackers in the out-of-the-box mode (as shown in Figure 2.1). We focus specifically on the *third-party connections* of wearable applications and the data that are shared with them. By third parties we imply any services that are provided by external entities (rather than the manufacturer of the device). Naturally, third-party connections of smart devices may be essential for their functioning. However, it is also entirely possible that some of the third parties are not required or even could be considered malicious. Previous works have analyzed third parties of various IoT devices [104, 119, 151], such as smart TVs, smart speakers, smart appliances, cameras, etc. In our research, we apply a similar approach to study the third-party connections of wearable devices. In particular, we focus on *unwanted*, *undesired*, *untrusted*, or *unnecessary* third parties, which we define as connections that are not essential for the functioning of the devices/applications.

The sharing of any personal data of users, including data derived from the network connections of a wearable, is regulated by the privacy policies of the vendor. Wearable suppliers must explicitly state which personal information they may disclose to third parties. It is evident that disclosing personal data to the entities not covered by privacy policies may result in heavy fines. However, allowing users to accept a policy without reading it significantly reduces the number of individuals who review privacy agreements. Meinert et al. found that fewer than 50% of users had ever read a privacy agreement [100]. Moreover, when users can skip through a privacy policy without reading it, they are less motivated to

consider the privacy risks [3, 137]. Besides, companies often deliberately draft terms and conditions in a specific way: to not clearly define how and to whom personal data may be shared [16]. Hence, vague policies authorize companies to uncontrollably gather and sell (or share) private data of users. In practice, end users of wearable trackers have very little control over which of their personal data are shared, and are largely unaware what third parties receive them. Clients are mandated to accept user agreements in order to use partner applications. Once that is done, users lose control over their own data. Therefore, *unwanted* connections of wearable applications may receive, in some cases, extremely sensitive information on regular users. In our research, we study such connections for both companion applications of wearable devices and their *partner* applications.

Partner applications. At present, major wearable vendors offer end users of wearables the ability to synchronize some of their health and activity data with partner-compatible applications of their choice. Such partners include various health services, major retailers, service applications, and even voice assistants. Regular users are able to authorize these apps to access various specific categories of their personal data, in order to improve the quality of service or enjoy a richer user experience.

In our research, we study the various aspects of privacy leakage in wearable devices, by answering the following questions:

> Who is communicating with consumer-level smartbands as part of their operation, and vice versa? What entities are being contacted by prominent wearable companion applications and their partner apps? Are they connecting only to the vendor servers in order to store wearable data? Are they only connecting with the vendor servers to store wearable data, or are they also communicating with other third-party entities? In the latter case, who are these third parties? Do regular users anticipate communicating with such entities? What data are being shared with them? Can unwanted third parties glean any sensitive insights?

3.1.1 Identification Pipeline

In this section, we describe the procedure to establish the contacted third parties and analyze data shared with them. To detect what third parties are contacted by partner apps we employ the following three-step pipeline.

Detecting contacted domains and IP addresses. Since wearable applications encrypt the communicated data, we intercept the traffic between the app and the cloud, using MITM (Figure 2.2). We obtain both the full URLs and IP addresses of third parties from such setup. We employ the Burp Suite [113] implementation of MITM. Since some of the studied companion/partner apps employ certificate pinning to prevent MITM, we utilize the EdXposed framework to disable it.

Identifying the data shared with third parties. Since MITM allows to view the IP packets,

we are able to check the contents of traffic that is sent from the partner apps to third parties. Once we have access to the plaintext data that are shared with third parties, we search for any private information inside. Contrary to popular belief, personal information can be sent via *both* POST and GET requests. We distinguish between fitness data that are collected by wearables, and other private information, including IP, location, phone characteristics, etc.

Learning about third parties. Finally, we try to establish what is the nature of the contacted third parties, i.e., what service they provide. This step is much more challenging than it appears at first glance. Indeed, some domain names for detected third parties may not instantly reveal who owns them or what they focus on. For example, a discovered third-party domain `d34yn14tavczy0.cloudfront.net` is one of the millions Content Delivery Network (CDN) hosted by private entities at Amazon CloudFront[1]. To learn the physical location of third parties we utilize Geoip [46]. To investigate the nature of third-party services, we employ Whois [160], Similarewb [129], and web search in general.

Note that since we conduct our experiments from Europe, the obtained results may not be reproducible in other parts of the world (e.g., America) due to different location of company servers, etc.

To summarize, for every wearable partner application we identify:

- The contacted third parties.
- Whether and what type of sensitive data are being shared.
- Origins and physical location of their servers.

3.1.2 Analysis of Popular Wearable Models

In our preliminary research [68], we analyzed unwanted third parties of 7 popular consumer-level wearable devices and their companion applications. In this dissertation, we briefly summarize our findings and describe the studied smartbands.

We start by describing the wearables and their associated companion applications:

1. Wearable: Xiaomi *MiBand* 4, Companion app: *MiFit*. The MiFit app is a popular health app developed by Xiaomi, which is one of the largest wearables producer [64]. We employed this device for our traffic analysis attack in Chapter 2.

2. Wearable: Samsung *Gear Fit* 2 *Pro*, Companion app: *Samsung Health*. Samsung is also present amongst the most popular wearable providers [64]. Likewise, we utilized this device for our traffic analysis attack in Chapter 2.

[1] https://aws.amazon.com/cloudfront/

3. Wearable: Huawei *Band 3 Pro*, Companion app: *Huawei Health*. Produced by Huawei, which is also a major manufacturer of wearables [64].

4. Wearable: the *Arbily* smartwatch, Companion app: *VeryFitPro*. VeryFitPro is a well-downloaded fitness application, but not as popular as the first three.

5. Wearable: the *RoHs* device, Companion app: *Wearfit*. A cheap device connecting to an outsourced companion app.

6. Wearable: the *M4* device, Companion app: *Tband*. Also a cheap device connecting to an outsourced companion app.

7. Wearable: the *Naxius* device, Companion app: *Yoho Sports*. Similar to the previous two, a cheap device connecting to an outsourced companion app.

App	Domain name	IP address	ISP	Origin	Site	Role
VeryFitPro	*IdoBleLogs*	47.244.67.196	Alibaba	China	Hong Kong	Logs
	abroad.apilocate.amap.com	205.204.101.28	Alibaba	USA	USA	Location
	cgicol.amap.com	198.11.136.99	Alibaba	China	USA	
	control.aps.amap.com	140.205.230.4	Alibaba	China	China	
	restapi.amap.com	47.246.74.109	Alibaba	China	USA	
MiFit	api.weibo.com	114.134.80.166	HGC	Hong Kong	Hong Kong	Social Network
	cgi.connect.qq.com	203.205.254.62	Tencent	China	Hong Kong	
	graph.facebook.com	31.13.84.8	Facebook	USA	Austria	
	logs.amap.com	203.119.211.252	Alibaba	China	China	Location
	abroad.apilocate.amap.com	47.88.68.79	Alibaba	China	USA	
	apilocate.amap.com	205.204.101.31	Alibaba	China	USA	
	restapi.amap.com	47.246.74.104	Alibaba	China	USA	
	login.sina.com.cn	58.63.236.212	ChinaNet	China	China	Ads
	xtrapath2.izatcloud.net	52.85.156.111	Amazon	USA	Greece	
Samsung Health	app-measurement.com	172.217.21.78	Google	USA	Germany	Analytics
Huawei Health	api.geetest.com	54.77.192.2	Amazon	USA	Ireland	API
TBand	iwhop.com	47.56.106.31	Alibaba	China	China	Weather
Wearfit	hmma.baidu.com	111.202.114.42	China Unicom	China	China	Ads
	openrcv.baidu.com	39.156.66.235	China Mobile	China	China	
	dxp.baidu.com	39.156.66.180	China Mobile	China	China	
	plbslog.umeng.com	203.119.214.123	Alibaba	China	China	
	iwhop.com	47.56.106.31	Alibaba	China	China	Weather
Yoho Sports	plbslog.umeng.com	203.119.214.124	Alibaba	China	China	Ads
	ulogs.umeng.com	203.119.214.124	Alibaba	China	China	
	log.umsns.com	203.119.215.106	Alibaba	China	China	

Table 3.1: Third parties that are contacted by various consumer-level wearable devices. *Origin* indicates the country of origin for ISPs. The *Site* column shows the *physical* location of the server. *Role* describes *why* the domain is contacted. The domain `ido-ble-lib.cn-hongkong.log.aliyuncs.com` is referred to as *IdoBleLogs*. Results for the Fitibt wearable are reported in the next sections along with its partner apps.

In Table 3.1 we summarize the third parties that are contacted by each smartband/app pair. Overall, companion apps contain a significant number of third-party services that

may not be anticipated by regular users. Notably, two applications (MiFit and VeryFitPro) contact a mapping service provided by Alibaba Group called Amap, sending location data to China even if a user registers from Europe. Moreover, MiFit establishes connections with three popular social networks: Tencent QQ (China), Weibo (China), and Facebook(US). Data are being exchanged with these networks even if the user is not registered there. Since users are never asked to consent to this contact and data sharing, they are likely largely unaware that their personal information is being disclosed. For example, Tencent QQ is contacted with a plain text GET request that contains the phone name and the OS version. Although this may seem not significant, it still enables the social network to gather data on individuals beyond its userbase.

For further details on the companion applications and results please refer to our published manuscript [68].

3.1.3 Analysis of Fitbit Partner Apps

We continue the work on identifying unwanted third-party connections of *partner* applications in Publication II. In our research, we focus on partner apps of Fitbit – one of the most popular wearable companies with very advanced API. While major vendors, including Apple, Samsung, Xiaomi, Fitbit, Huawei, etc., are challenged to protect privacy of their users, the partner apps often do not receive the same attention from privacy activists. Andrade et al. established that users are more likely to grant access to their personal data to companies with a credible reputation [9]. Given Fitbit's universal reputation as a company that values user privacy, its credibility may extend to its affiliated apps. In our research, we set out to investigate whether partner apps provide the same high standards of privacy protection and transparency as Fitbit.

Fitbit is currently partnering with over 40 apps[2]. In our research, we focused on 10 Android apps from partners that offer health services and have at least 50, 000 downloads on Google Play. That is, we did not study official retail partners such as Walgreens or Dick's Sporting Goods, which offer discounts based on how active a person has been. We also did not investigate applications like Amazon Alexa that are not directly fitness-related. Furthermore, we did not rigorously analyze apps that require special equipment. Many of the studied apps support other wearable trackers. We present partner apps sorted by the number of downloads (as of October 2020) in descending order:

1. *MyFitnessPal.* Downloads: 50 million. A popular health app that allows users to track various aspects of their health. It collects a number of burned calories from Fitbit to modify daily calories goal. The app is one of the most popular health apps available.

[2]The detailed description of the apps and the supported interactions with Fitbit can be found here: `https://staticcs.fitbit.com/partnership`

2. *Strava.* Downloads: 10+ million. A well-known fitness tracking app. GPS workouts recorded by Fitbit can be synchronized with the Strava application.

3. *MapMyRun, RunKeeper and Endomondo.* Downloads: 10+ million each. These apps are tracking running activities. Synchronization with Fitbit allows them to access workouts recorded by the smartband.

4. *MINDBODY.* Downloads: 1+ million. A training app that allows users to sign up for the classes in their local area. Mindbody requests the training data collected by a Fitbit device.

5. *Weightloss Running.* Downloads: 1+ million. An app that offers personal training plans for its users. The application pulls the training data from Fitbit.

6. *Hidrate Spark.* Downloads: 100 thousand. A health app that tracks water intake. It receives the steps information from Fitbit and adjusts the daily water consumption goal.

7. *Wokamon.* Downloads: 100 thousand. A mobile game that encourages adopting a healthy lifestyle. It accounts step data from a Fitbit tracker for in-game rewards.

8. *Nudge Health Tracking.* Downloads: 50 thousand. A health app, enabling users to connect with real-life coaches. The Nudge app synchronizes various health snapshots from the Fitbit account.

Contacted third parties. We depict a detailed description of the third parties and ISPs that host their servers in Table 3.2. We provide a comprehensive overview of the most "interesting" findings about the studied partners and report the most contacted third-party services.

Strictly speaking, Fitbit can be considered a third party, since it is not owned by any of the studied partner applications. However, since contacting Fitbit is expected for every app, we do not mention it in this section.

Facebook. We found that 9/10 (90%) of the analyzed apps share sensitive data with the Facebook social media. Since these apps allow users to register or sign in with their Facebook profile, it is natural to assume that the social network will be contacted. However, we established that Facebook is contacted, and the data are shared even when a user does not have a Facebook profile. This means that the social network can gather information about people beyond its own userbase. Most of the partner apps interact with Facebook through the `graph.facebook.com` domain, which facilitates easy interaction with the social graph. However, this process inevitably leads to the sharing of sensitive user data.

Data	Apps
Phone manufacturer, model, etc.	MyFitnessPal, Strava, Runkeeper, Weightloss Running, Wokamon, Nudge
AAID	MyFitnessPal, Strava, Runkeeper, Weightloss Running, Wokamon
Email	MyFitnessPal, Strava, Runkeeper
Connection details	MyFitnessPal, Runkeeper
Location	MyFitnessPal, Strava, Weightloss Running
Demographics	MyFitnessPal

Table 3.3: Sensitive information shared by studied apps with third parties (as of July 2022). For each app the data are shared with at least one of the unwanted third parties.

In particular, Facebook records every session of each partner app, collecting information on the phone manufacturer, localization, timezone, location (country), Sim carrier, and in some cases, even the gyroscope parameters. Facebook also obtains Android Advertising ID (AAID) – a unique cross-app identifier that allows for profiling users across different apps. For instance, if a user registers for Facebook, Instagram, and WhatsApp using three different email addresses, Meta (formerly Facebook) can still associate all three accounts with a single user. The ability to link multiple identities to a single user characterizes a so-called "permanent record."

Crashlytics/Google. Firebase, a platform for developing Android apps owned by Google, is used by half of the apps in the study (5/10). In particular, 40% of the partner apps make use of Crashlytics, a crash report service that is a subsidiary of Firebase. While Crashlytics is a helpful tool for identifying and fixing app crashes, it also records a significant amount of app-related information. In fact, it records every action taken by the user within the app and the state of the phone parameters during that step, including the battery level and velocity, proximity, screen orientation, and the amount of RAM and disk space being used. This can result in an unprecedented amount of data being collected about app usage and user behavior.

Branch. Half of the apps examined (5/10) use a deep linking service called Branch, which improves navigation within the app. However, in addition to providing its linking service, the Branch API also sends a large amount of private data to its servers. These data include details about the user's phone model and manufacturer, screen resolution and DPI, OS version and architecture, and the times of app installation and updates.

Leaked sensitive insights. In Table 3.4 we detail the sensitive data third parties receive from Fitbit partner apps. The obtained results suggest that most of the studied partner applications contact "unexpected" third parties. These third parties include social networks,

analytics providers, advertisement services, and weather APIs.

Since the original data presented in Table 3.4 were collected in October 2020, we repeated the procedure for discovering leaks in Publication III (July 2022). In Table 3.3 we report the "newer" high-level insights that are leaked. The most interesting findings include approximate and exact location, connection specifics (Wi-Fi or cellular), email, and even demographics.

For example, most of the apps share a unique cross-app AAID with `graph.facebook.com`. Additionally, `events.mapbox.com` receives precise latitude and longitude information every second (as seen in Listing 6), while `api.segment.io` receives massive amounts of private data, including email, gender, age, and lifestyle (as seen in Listing 5). Naturally, these types of data are extremely sensitive and can be used for mass profiling. Thus, unwanted third party may be able to identify individuals who lead sedentary lifestyles and rarely exercise, which in turn may be used for extensive profiling.

Despite most of the popular partner applications offering their service free of charge, the real price that the users pay is their sensitive data. That is, partner apps utilize the data of their users to indirectly "pay" for the convenience and service. Hence, it is an individual user who contributes data to fund a better application experience.

Listing 5 Fragments of a file shared with `api.segment.io` by MyFitnessPal. Email, gender, userId, and age are anonymized. The leaked data also include weight goal, and lifestyle.

```
"traits":{"device_theme_state":"light","gender":gender,"profile_country":
"US","email_encrypted":"07cc4d9626e09476ef577e464b03157ebd9e9ea1",
"facebook_connected":false,"email":email,"day_of_week":
"Monday","primary_step_source_set":"fitbit","anonymousId":
"85504873-d4d9-4210-b957-3b5ef0a45f28","weight_goal":"lose","email_
verified":true,"weekly_weight_goal":"lose_1.5_pound_per_week","userId":
"userId","age":age}
...
"lifestyle":"Sedentary"
```

Listing 6 Fragment of a file sent by Strava to `events.mapbox.com`. Precise coordinates (which are anonymized) are disclosed. Date and timezone are anonymized.

```
{"horizontalAccuracy":4.0,"altitude":99.0,"applicationState":"Background",
"created":"2022-**-**T**:**:**.000+**00","event":"location","lat":**,
"lng":**,"operatingSystem":"Android - 9","permissionStatus":"AllowAlways",
"sessionId":"74221bb1-f94b-4c40-a4de-50bb32139e00","source":"mapbox"}
```

App	Domain name	IP address	ISP	Origin	Site	Role
MyFitnessPal	z.moatads.com	104.107.144.129	Akamai tech	USA	Greece	Ads
	ads.mopub.com	192.48.236.12	MoPub	USA	USA	
	cdn.branch.io	52.85.158.100	Amazon	USA	Greece	
	api2.branch.io	52.85.158.120	Amazon	USA	Greece	
	aax-eu.amazon-adsystem.com	52.95.123.41	Amazon	USA	Ireland	
	s3.amazonaws.com	52.216.132.85	Amazon	USA	USA	
	api.amplitude.com	35.160.169.182	Amazon	USA	USA	Analytics
	crashlyticsreports-pa.googleapis.com	216.58.212.163	Google	USA	USA	
	api.ua.com	52.85.158.128	Amazon	USA	Greece	API
	config.88-f.net	104.19.161.19	Cloudflare	USA	Canada	
	d34yn14tavczy0.cloudfront.net	52.85.158.22	Amazon	USA	Greece	Photo
	graph.facebook.com	69.171.250.15	Facebook	USA	USA	Social
Strava	app.adjust.com	185.151.204.13	Adjust GmbH	Germany	Netherlands	Analytics
	sessions.bugsnag.com	35.190.88.7	Google	USA	USA	
	firebaseinstallations.googleapis.com	216.58.206.74	Google	USA	USA	API
	api2.branch.io	52.85.158.114	Amazon	USA	Greece	
	api.iterable.com	52.205.72.116	Amazon	USA	USA	
	dgalywyr863hv.cloudfront.net	52.85.155.138	Amazon	USA	Greece	Photo
	graph.facebook.com	31.13.84.8	Facebook	USA	Austria	Social
MapMyRun	ads.mopub.com	192.48.236.12	MoPub	USA	USA	Ads
	hub.samsungapps.com	34.254.23.31	Amazon	USA	Ireland	
	pubads.g.doubleclick.net	216.58.206.34	Google	USA	USA	
	pagead2.googleadservices.com	216.58.209.34	Google	USA	USA	
	cdn.branch.io	52.85.158.100	Amazon	USA	Greece	
	api.amplitude.com	54.203.10.108	Amazon	USA	USA	Analytics
	api2.branch.io	52.85.158.120	Amazon	USA	Greece	API
	graph.facebook.com	31.13.84.8	Facebook	USA	Austria	Social
RunKeeper	id-prod-age.prod.asics.digital	34.197.96.234	Amazon	USA	USA	Ads
	launches.appsflyer.com	18.203.26.15	Amazon	USA	Ireland	Analytics
	api.amplitude.com	52.40.97.110	Amazon	USA	USA	
	crashlyticsreports-pa.googleapis.com	216.58.206.67	Google	USA	USA	
	api.iterable.com	52.55.152.71	Amazon	USA	USA	API
	graph.facebook.com	31.13.84.8	Facebook	USA	Austria	Social
Endomondo	api.amplitude.com	52.40.97.110	Amazon	USA	USA	Analytics
	graph.facebook.com	31.13.84.8	Facebook	USA	Austria	Social
MINDBODY	cdn.branch.io	52.85.158.72	Amazon	USA	Greece	Ads
	mobile-collector.newrelic.com	151.101.2.110	Fastly	USA	USA	Analytics
	identity.mparticle.com	151.101.242.133	Fastly	USA	Italy	API
	sdk.iad-03.braze.com	151.101.17.208	Fastly	USA	USA	
	api2.branch.io	52.85.158.37	Amazon	USA	Greece	
	logx.optimizely.com	52.4.25.221	Amazon	USA	USA	
Weightloss	t.appsflyer.com	79.125.107.112	Amazon	USA	Ireland	Ads
	ads.mopub.com	192.48.236.9	MoPub	USA	USA	
	ads.verv.com	5.9.122.176	Hetzner Online	USA	Germany	
	cb.mopub.com	192.48.236.12	MoPub	USA	USA	
	firebase-settings.crashlytics.com	172.217.16.163	Google	USA	Germany	Analytics
	api.rockmyrun.com	52.89.196.53	Amazon	USA	USA	API
	graph.facebook.com	69.171.250.15	Facebook	USA	USA	Social
	www.facebook.com	69.171.250.35	Facebook	USA	USA	
	api.darksky.net	52.21.90.77	Amazon	USA	USA	Weather
HidrateSpark	reports.crashlytics.com	54.243.164.158	Google	USA	USA	Analytics
	graph.facebook.com	69.171.250.15	Facebook	USA	USA	Social
Wokamon	a.appbaqend.com	104.17.72.8	CloudFlare	USA	Canada	Ads
	outcome-ssp.supersonicads.com	52.85.158.20	Amazon	USA	Greece	
	gum.criteo.com	178.250.0.157	Criteo SA	France	France	
	devs.data.mob.com	116.211.155.227	ChinaNET	China	China	Analytics
	api.share.mob.com	118.212.233.191	China Unicom	China	China	API
	graph.facebook.com	69.171.250.15	Facebook	USA	USA	Social
Nudge	cdn.branch.io	52.85.158.64	Amazon	USA	Greece	Ads
	stats.pusher.com	52.90.41.11	Amazon	USA	USA	Analytics
	exp.host	104.197.216.164	Google	USA	USA	API
	d1wp6m56sqw74a.cloudfront.net	52.85.155.179	Amazon	USA	Greece	
	api2.branch.io	52.85.158.37	Amazon	USA	Greece	
	graph.facebook.com	69.171.250.15	Facebook	USA	USA	Social

Table 3.2: Third parties that are contacted by the partner apps (as of October 2020). *Origin* represents the headquarters location of ISPs. The *Site* column refers to the *physical* location of the contacted servers. *Role* describes services that third parties provide.

App	Shared Data	Third Party
MyFitnessPal	Phone model	Facebook
	Location	Facebook
	Phone Data	Branch
	Connection Data	Amplitude
	Phone Data	Amazon
	Phone Details	Google
Strava	Connection Data	Branch
	Phone Data	Branch
	Phone Details	Bugsnag
MapMyRun	Phone Data	Branch
	Phone Data	Amplitude
RunKeeper	Phone Data	Facebook
	Location	Facebook
	Email	Iterable
	Phone Data	Iterable
	Phone Data	Amplitude
	Location	Amplitude
	Phone Details	Google
Endomondo	Phone Data	Facebook
	Location	Facebook
	Location	Amplitude
	Phone Data	Amplitude
MINDBODY	Email	Mparticle
	Connection Data	Branch
	Phone Data	Branch
	Connection Data	Newrelic
Weightloss	Phone Model	Facebook
	Location	Facebook
	Sim Carrier	Facebook
	App Data	Appsflyer
	Phone Details	Google
	Phone Data	Facebook
HidrateSpark	Phone Data	Facebook
	Location	Facebook
	App Details	Facebook
	Phone Details	Google
Wokamon	Phone Data	Supersonicads
	Connection Data	Supersonicads
	Sensor Data	Facebook
	Phone Data	Facebook
Nudge	Phone Data	Facebook
	Location	Facebook
	Sim Carrier	Facebook
	Connection data	Branch
	Phone Data	Branch

Table 3.4: Data that are shared with the third parties during runtime of the Fitbit partner apps (as of October 2020). *Phone data* accounts for the manufacturer, model, OS, and screen resolution. Location is approximate, not precise coordinates.

3.2 Preventing Unwanted Connections of Wearables

Given the severe sensitivity of private data that are sent to unwanted third parties, as described in previous sections, the question arises: is it feasible to mitigate such leaks? In Publication III, we propose a methodology for blocking undesired traffic, and evaluate it empirically.

The methods currently available for using wearable devices while avoiding unwanted third-party contact include installing custom mobile applications [42, 45]. These apps prevent the device from connecting to the Internet, but may not support all the features of the original applications. They are also only compatible with a limited number of commercial wearable devices. Naturally, the above details renders such an approach not applicable for most of the regular users for wearables. Another solution designed to block unnecessary third-party communications in IoT devices [94,95] uses a dynamic approach to identify and block undesired traffic. However, the proposed methodology involves different blocking strategies for various device groups, making it difficult for regular end users to set up. Additionally, since it does not rely on existing maintained blocking lists, there could be false positives that lead to improper functioning of the devices or applications. Finally, since the previously proposed approaches require a specifically configured Internet access point, they cannot be used when the device is outside of designated networks.

Therefore, we set out to investigate if there is a simpler solution available for regular end users of wearables. Specifically, we aimed to determine if adblockers, which are browser content filtering extensions, could be a feasible solution. In 2019, an estimated 763.5 million people used adblockers [134] because they are easy to install, user-friendly, and highly effective at blocking advertisements and trackers without disrupting the user's browsing experience. Additionally, adblockers often have regularly updated blocklists that are tested to minimize the risk of false positive entries. The focus of our study is to analyze the third-party entities contacted by apps associated with Fitbit. We examine two popular blocklist collections: uBlock Origin [150] – one of the most popular browser content filtering extensions – and Firebog [40] – another well-known collection of maintained domain lists. We study whether blocking such unwanted destinations would cause any functional disruption to the official Fitbit apps and its partners.

The research questions that we investigate for this problem are as follows:

(Q1) Which third parties are being contacted by the applications associated with Fitbit?

(Q2) Does blocking domains from highly ranked blocklists impact the core functionality of the devices?

(Q3) What are the "most unwanted" third-party entities, and which blocklists are most effective in detecting them?

To the best of our knowledge, we are the first to investigate blocking unwanted traffic of wearable applications. Unlike previous studies on disabling unnecessary IoT communications, our approach is not affected by changes in network traffic of the analyzed applications because our blocking rules are based entirely on existing blocklist collections. Moreover, our method can be readily utilized by regular users of the devices, such as via mobile filtering applications (i.e., adblockers), and does not necessitate specialized network equipment.

3.2.1 Setup

Similar to Publication II, in our experiments, we utilize two Fitbit Versa 2 fitness trackers and two Xiaomi Redmi 7 phones that run the official Fitbit companion application and studied partner apps. The mobile phones are connected to the Internet through a Wi-Fi hotspot hosted by a laptop computer.

Discovering Third Parties. We employ the MITM approach (Figure 2.2) to identify all entities that are being contacted by the studied applications. We leverage the EdExposed framework to bypass certificate pinning.

Blocking Domains. To prevent unnecessary third parties from accessing the applications, we modify the *hosts* file for each app being studied. This file is generated by the operating system and maps domain names to their respective IP addresses. Since the hosts file is examined before the Domain Name System (DNS), unwanted domains can be resolved as a localhost (127.0.0.1), preventing packets from traversing the global web. Since the phone's Internet connection is established through a Wi-Fi hotspot, all traffic essentially goes through the laptop. Therefore, disabling domains on the computer also restricts the phone from connecting to them. We maintain a separate hosts file (a list of blocked domain names) for each application.

Employed Blocklist Collections. Our original plan was to test the possibility of blocking certain domains that adblockers typically filter, without breaking the wearable applications. To make this happen, we chose to use uBlock Origin (henceforth Ublock), a well-known and widely used content blocker with over 10 million Chrome[3] and nearly 6 million Firefox[4] users. Currently, Ublock has over 50 domain-based filtering lists that are regularly updated by developers and researchers. These lists cover a range of categories, including default, anti-advertisement, anti-tracking, anti-malware, and "annoyancess," as well as regional-specific. Overall, Ublock supports up to 600K blocking rules. It is worth noting that despite the high number of filters, the blocklists are regularly maintained, and there are very few instances of false positives, resulting in website breakage. Similarly to the previous works on content blocking [94, 151], in our research, we consider another blocklist collection called Firebog [40]. It aggregates various categories of rules, including malicious, advertising,

[3] https://chrome.google.com/webstore/detail/ublock-origin/cjpalhdlnbpafiamejdnhcphjbkeiagm
[4] https://addons.mozilla.org/en-US/firefox/addon/ublock-origin/

suspicious, and tracking & telemetry lists. Firebog contains a total of 60 blocklists, with over 5.3 million domains (excluding non-recommended sources).

It is worth noting that several blocklists are present in both collections, such as actively supported *EasyList* and *EasyPrivacy*.

3.2.2 Experiment

In this section, we detail the experiments that were carried out to block unwanted third parties. Again, the high level idea of these experiments is to investigate whether blocking undesired third parties would affect the operation of apps and devices.

Official Fitbit app. Specifically, our experiments aim at not only assessing the impact on the application's workflow, but also on fitness data produced by wearables. However, wearable activity trackers are more complex than other IoT devices, which makes it challenging to ensure that every aspect of the application/device remains unaffected. Unlike a smart bulb, which can easily be identified as malfunctioning if it stops turning on/off, wearable devices gather various types of data and do not have a single most important function. As a result, we need to verify whether disabling a third-party would have any impact on fitness data that have already been collected. For example, would disabling `test.com` result in incorrect step counts being recorded? Therefore, we conduct empirical experiments to confirm that filtering rules do not have any impact on the fitness data collected by wearables.

We created two identical Fitbit accounts, with matching information for gender, weight, height, age, and other parameters. Each account was paired with a separate Fitbit Versa 2 device. In the experiment, I *simultaneously* wore both fitness trackers on the same hand, with one having all unwanted third parties disabled according to the blocklists, and the other being used in its off-the-shelf mode without blocking or interception. We then compared the collected data from both trackers to identify any malfunctions. As the devices were not worn in exactly the same position, there was a natural difference in the collected fitness data. To address this issue, we conducted a second round of experiments in which I wore both devices without blocking any domains to establish a baseline difference due to errors and wrist placement. Both rounds lasted five days, from Monday to Friday, during which I tried to engage in as many trackable activities as possible, including various workouts, measuring my heartbeat, and monitoring my sleep. Finally, we compared the *differences* between the two rounds to determine whether blocking the contacted domains significantly affected the discrepancy between the simultaneously worn trackers.

A user study by Chong et al. [24] found that individuals who use wearables consider tracking their steps, sleep, and exercise the primary factors when purchasing a fitness tracker. We use the same metrics to evaluate and compare the results of 2 different devices.

Partner apps of Fitbit. It is not feasible to thoroughly analyze all of the features for each app that we study when we block unwanted third parties. Instead, we only confirm that the

data imported from Fitbit match the corresponding values in the Fitbit cloud. We examine some of the partner apps for Fitbit that we previously studied in Section 3.1.3, selecting the apps that allow users to synchronize their Fitbit activity data, so as to verify whether it is feasible to disable unwanted connections. We use the most recent available versions of the applications (as of July 2022), which are specified in Table 3.5.

	App and version	All Third Parties	# Blocked
	Fitbit v3.18	graph.facebook.com, api.mixpanel.com, decide.mixpanel.com, cdn.optimizely.com, m.stripe.com, mcbs1myt8rhvg1jhw6dlgdpy4fly.device.marketingcloudapis.com, s7.device.marketingcloudapis.com, app-measurement.com, logx.optimizely.com, firebase-settings.crashlytics.com, settings.crashlytics.com, in.appcenter.ms	10/12
Partner Apps	MyFitnessPal v22.15.0	graph.facebook.com, sdk.iad-06.braze.com, z.moatads.com, api2.branch.io, firebase-settings.crashlytics.com, crashlyticsreports-pa.googleapis.com, cdn.branch.io, sdk.split.io, api.segment.io, aax-eu.amazon-adsystem.com, c.amazon-adsystem.com, mads.amazon-adsystem.com, api2.amplitude.com, ads.mopub.com, googleads.g.doubleclick.net, pubads.g.doubleclick.net, auth.split.io, streaming.split.io, events.split.io, d34yn14tavczy0.cloudfront.net, pagead2.googleadservices.com	20/21
	Strava v267.9	graph.facebook.com, sessions.bugsnag.com, api2.branch.io, cdn.branch.io, app.adjust.com, api.iterable.com, events.mapbox.com, api.mapbox.com	7/8
	Runkeeper v13.4	graph.facebook.com, api.iterable.com, launches.appsflyer.com, api2.amplitude.com, crashlyticsreports-pa.googleapis.com, beacons.gcp.gvt2.com	6/6
	Weightloss Running v6.8.13	graph.facebook.com, launches.appsflyer.com, ads.mopub.com, api2.amplitude.com	4/4
	Wokamon v2.17.5	graph.facebook.com, api.share.mob.com, c.data.mob.com, api.exc.mob.com, m.data.mob.com , ms.applovin.com, rt.applovin.com, connect.tapjoy.com, a4.applovin.com, d.applovin.com, rpc.tapjoy.com, placements.tapjoy.com, googleads.g.doubleclick.net, pagead2.googleadservices.com, data.flurry.com	15/15
	Nudge v6.3.3	exp.host, sentry.io, ws-mt1.pusher.com, sockjs-mt1.pusher.com, sock252-mt1.pusher.com	1/5

Table 3.5: Third parties contacted by the studied apps. The domains that are not contained in the blocklists are in blue; while the rest are considered unnecessary and can be disabled.

3.2.3 Analysis of Third Parties

To address Q1, we first describe the third parties that Fitbit and its partner apps communicate with. A complete list of third parties contacted by these apps as part of their operation is provided in Table 3.5. We do not consider the Fitbit API a third party for the partner apps, as it is an expected destination in order to request fitness data. Our findings reveal that both the official Fitbit app and partner applications communicate with a significant number of external domains. In fact, 3 out of the 7 studied apps communicate *only* with undesired third parties. Our results indicate that these unwanted entities primarily consist of advertising providers, tracking/analytics services, and various content delivery networks. Notably, 6 out of 7 apps send data to Facebook, even if users do not use/have the social network credentials.

We further investigate which unnecessary destinations are most frequently contacted

by wearable apps. Figure 3.2 shows the data flows between the studied apps and unwanted third parties, with third parties grouped according to the organizations that operate them. For instance, Google is represented not only by Google ads but also by Crashlytics, an analytics provider owned by Google. The width of the flows in the figure is proportional to the number of second-level domains, where a wider flow indicates a larger number of second-level domains. For example, the flow for apps that contact Branch via both `api2.branch.io` and `cdn.branch.io` is twice as wide as that for Iterable, which is represented by a single domain (`api.iterable.com`). Our results demonstrate that Facebook (Meta) and Google are the most frequently contacted third-party organizations, with 6/7 and 4/7 apps communicating with them, respectively. Furthermore, several companies, including Google, Amazon, and Branch, provide more than one second-level domain per their services. Table 3.6 provides a breakdown of the individual third parties contacted by *multiple* apps, with most domains being contacted by 2 applications.

Third Party	Contacted by Apps
`graph.facebook.com`	Fitbit, MyFitnessPal, Runkeeper Strava, Weightloss Running, Wokamon
`firebase-settings.crashlytics.com`	Fitbit, MyFitnessPal
`crashlyticsreports-pa.googleapis.com`	MyFitnessPal, Weightloss Running
`pagead2.googleadservices.com`	MyFitnessPal, Wokamon
`googleads.g.doubleclick.net`	MyFitnessPal, Wokamon
`*.branch.io`	MyFitnessPal, Strava
`api2.amplitude.com`	MyFitnessPal, Runkeeper
`launches.appsflyer.com`	Runkeeper, Weightloss Running
`ads.mopub.com`	MyFitnessPal, Weightloss Running
`api.iterable.com`	Strava, Runkeeper

Table 3.6: Unnecessary domains contacted by multiple partner apps.

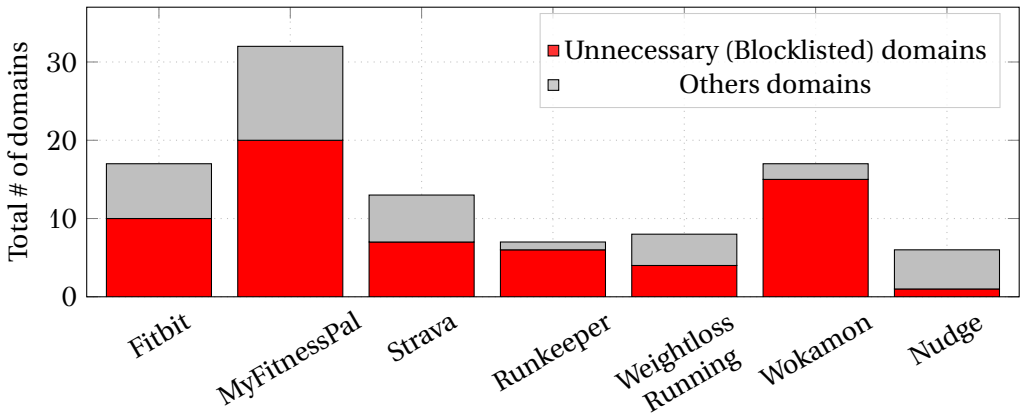

Figure 3.1: The proportion of unnecessary third-party connections to all the contacted domains (including first parties).

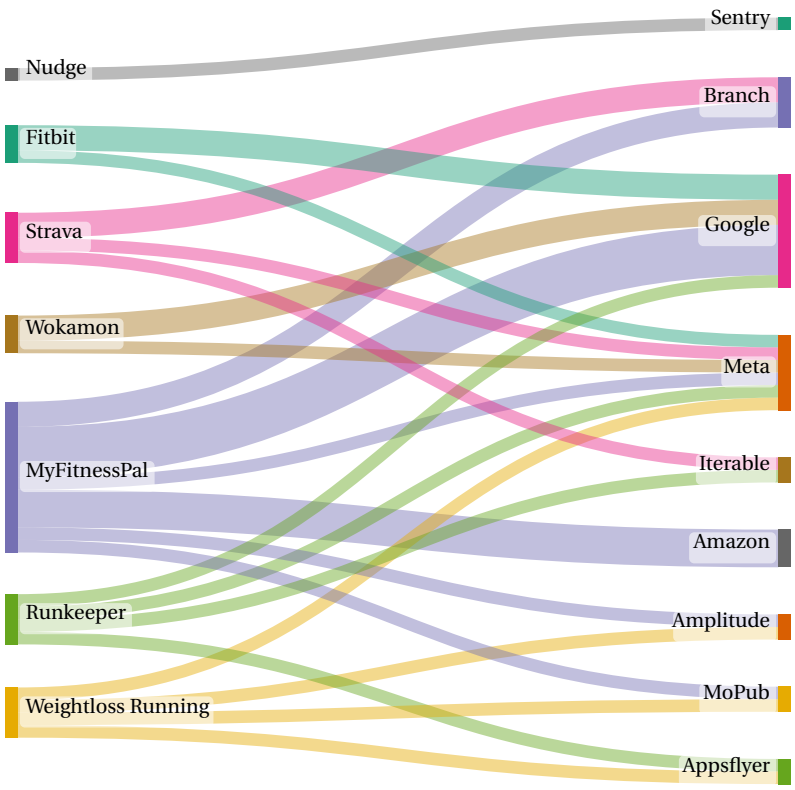

Figure 3.2: Mapping of the Fitbit-associated apps to the companies that provide unnecessary third-party services. The width of the flows corresponds to the number of second-level domains per organization. The most frequently contacted organizations are depicted.

To gain a better understanding for the proportion of all contacted entities that need to be disabled, we present the results for each app individually in Figure 3.1. This comprises all connections, including first parties, the Fitbit API, and other entities that were not reported in Table 3.5. The results show that for almost all applications, the number of unwanted connections exceeds 50%. In other words, at least half of the contacted destinations may be unnecessary or even harmful.

Overall, the obtained results suggest that more than 88% of the third parties are unwanted and, hence, have negative added value for the end users.

3.2.4 Blocking Unnecessary Traffic

In this section, we present the results of our experiments that address Q2. The raw daily cumulative data for both rounds of the experiment are displayed in Table 3.7. We detail daily

Activity	Wearable	Round 1				
		Day 1	Day 2	Day 3	Day 4	Day 5
Steps	1	2184	4869	2960	5140	7685
	2	2110	4862	3019	5035	7706
Distance	1	1650	3700	2120	3900	7710
	2	1600	3690	2060	3820	7060
Sleep Total	1	425	546	408	429	449
	2	424	538	418	447	462
Light Sleep	1	294	317	283	277	303
	2	289	353	274	257	312
REM Sleep	1	98	162	65	71	98
	2	102	113	78	103	106
Deep Sleep	1	33	67	60	81	48
	2	33	72	66	87	44
		Round 2				
Steps	1	2077	6670	4888	2859	3194
	2	2023	6660	4759	2799	3032
Distance	1	1530	7190	3710	2160	2390
	2	1500	6420	3610	2120	2300
Sleep Total	1	427	407	452	470	390
	2	426	399	432	467	403
Light Sleep	1	287	276	319	268	285
	2	276	310	314	252	289
REM Sleep	1	88	71	93	116	60
	2	103	44	78	140	68
Deep Sleep	1	52	60	40	86	45
	2	47	45	40	75	46

Table 3.7: Complete listing of the obtained results. Both wearables are simultaneously worn on the same hand. For the second device in *Round 2* the unnecessary third parties were disabled. Distance is measured in meters; sleep in minutes.

steps, distance, and calories, as well as various type of sleep. Again, during the experiment, I wore both devices simultaneously on the same non-dominant hand. In the first round, both wearables were set up to contact all the default third parties to establish the baseline difference due to varying positions of the trackers. In the second round we disable all the unnecessary connections listed in Table 3.5. Based on visual inspection, there does not appear to be a significant difference between the discrepancies observed in the two rounds. We also observed that concurrently worn trackers sometimes interchange REM and light sleep minutes. However, the total nightly sleep seems to be consistently recorded.

Nevertheless, for the sake of formality, we present statistical analyses to compare the

Activity	RMSE R1	RMSE R2	NRMSE R1	NRMSE R2
Steps	64	99.5	0.011	0.021
Distance	295	350.3	0.048	0.062
Sleep Total	11.5	11.3	0.083	0.141
Light Sleep	19.4	17.7	0.202	0.264
REM Sleep	27.1	19.1	0.279	0.199
Deep Sleep	4.8	8.7	0.089	0.189

Table 3.8: Comparison of Root Mean Square Error (RMSE) and Normalized RMSE (NRMSE) for round 1 (R1) and 2 (R2).

errors between the two rounds of the experiment. In fact, standard statistical tests, such as the t-test or Kolmogorov-Smirnov test, are only able to reject the null hypothesis that the data points are from the same distribution. In other words, it is not feasible to accept the null hypothesis and claim that the data from both rounds are similar, despite the fact that they originate from identical devices on the same hand. Therefore, we present statistical values that can be interpreted for our case in Table 3.8. More specifically, we report the Root Mean Square Error (RMSE) and Normalized Root Mean Square Error (NRMSE) for the daily values of activities between the devices. We separately calculate these metrics for both rounds and estimate the difference between the errors. The obtained results indicate that the highest NRMSE difference is observed for light/REM sleep, whereas the lowest is for the total sleep duration. Our findings suggest that there is no significant discrepancy between the errors of the identical and modified pairs of the devices.

Regarding partner apps, we verified that blocking unnecessary destinations does not affect the correct import of Fitbit data, as the values observed in the partner apps match those in the Fitbit cloud. We experimented with *all* types of fitness data that can be exported from Fitbit for every partner application that we studied.

Overall, it appears that blocking unwanted destinations has no impact on the workflow of the official Fitbit application and the partner apps examined in our study.

3.2.5 Blocklists Ranking

Having verified that domain-based filtering rules do not affect the core functionality of the wearable applications, we proceeded to address Q3, which involves identifying the third-party applications that are considered "the most undesirable."

In Table 3.9 we rank third parties by counting the number of blocklists that include their domains. Essentially, we identify as "the most unwanted" those entities that are included in many blocklists designed to prevent various types of unwanted content. Based on this method, we find Google's DoubleClick and Amazon's AdSystem to be the highest hitting services, as they are present in more than 15 different filtering lists. It is worth

# Occurrences	Third Parties	Collection	Ublock Default
18	`googleads.g.doubleclick.net`	UF	Yes
16	`pubads.g.doubleclick.net`	UF	Yes
15	`aax-eu.amazon-adsystem.com`	UF	Yes
14	`c.amazon-adsystem.com`	UF	Yes
13	`ads.mopub.com`, `mads.amazon-adsystem.com`	UF, UF	No, Yes
11	`z.moatads.com`	UF	Yes
10	`app-measurement.com`	UF	No
9	`api.mixpanel.com`, `pagead2.googleadservices.com`, `data.flurry.com`	UF, UF, F	No, Yes, No
8	`decide.mixpanel.com`, `launches.appsflyer.com`, `d.applovin.com`	UF, UF, UF	No, No, No
7	`api2.branch.io`, `api2.amplitude.com`, `events.mapbox.com`,	UF, UF, UF	No, Yes, Yes
	`m.data.mob.com`, `api.exc.mob.com`, `api.share.mob.com`,	UF, UF, UF	No, No, No
	`ms.applovin.com`, `rt.applovin.com`, `*.tapjoy.com`	UF, UF, UF	No, No, Yes
6	`settings.crashlytics.com`, `api.segment.io`,	UF, UF	Yes, No
	`events.split.io`, `cdn.branch.io`, `c.data.mob.com`	UF, F, UF	Yes, No, No
5	`logx.optimizely.com`, `sdk.iad-06.braze.com`,	UF, UF	Yes, No
	`crashlyticsreports-pa.googleapis.com`, `app.adjust.com`	UF, UF	Yes, Yes
4	`cdn.optimizely.com`, `firebase-settings.crashlytics.com`,	F, UF	No, Yes
	`sessions.bugsnag.com`, `api.iterable.com`	UF, UF	No, Yes
3	`sdk.split.io`, `auth.split.io`, `beacons.gcp.gvt2.com`, `a4.applovin.com`	F, F, UF, UF	No, No, Yes, No
2	`graph.facebook.com`	F	No
1	`*.device.marketingcloudapis.com`, `streaming.split.io`, `sentry.io`	F, F, F	No, No, No

Table 3.9: Ranking of the unnecessary third parties based on the number of blocklists containing them. We indicate whether a third party is detected by a collection of blocklists (U = Ublock, F = Firebog, UF = both). We also report whether a third party is blocked by a default installation of uBlock Origin.

noting that all these domains are disabled by default in the Ublock adblocker that runs on millions of computers worldwide. We also find that among the third parties contained in the blocklists, only 8 were present exclusively in the Firebog collection (which is used for more advanced filtering), while all other entities are included in both collections. As a result, simply employing a popular adblocker may immensely help regular users in safeguarding their privacy, while their wearable devices can still function without any issues. Overall, on average wearable third parties are present in 7 blocklists (with a median of 7 as well).

Continuing our investigation into Q3, we examine the most effective blocklists for preventing unwanted connections of wearable applications. The results are presented in Table 3.10, and it comes as no surprise that the highest hitting lists are those that target mobile tracking and advertising (top 4 in the table). Notably, the widely used and well-maintained *EasyList* and *EasyPrivacy* lists contain only 4 unnecessary domains each. This is likely because they are primarily designed to combat *web* advertising and tracking, as opposed to the mobile ecosystem.

We further stress that the proposed approach can be set up by an *average* Fitbit user via adblocking apps.

Blocklist	# Blocked	Collection
bigdargon	38	Firebog
ads-and-tracking-extended	38	Firebog
adaway	36	Firebog
anudeepND	32	Firebog
VeleSila	17	Firebog
AdGuard Mobile Ads	14	Ublock
Peter Lowe's list	14	Ublock
someonewhocares	12	Firebog
RooneyMcNibNug	10	Firebog
jdlingyu	10	Firebog
Dan Pollock's list	10	Ublock
AdGuard Tracking Protection	10	Ublock
neohostsbasic	9	Firebog
winhelp2002	9	Firebog
Perflyst android-tracking	8	Firebog
KOR: List-KR	6	Ublock
POL list	5	Ublock
EasyPrivacy	4	Firebog Ublock
EasyList	4	Firebog Ublock

Table 3.10: Ranking of blocklists based on the number of unnecessary third parties of wearables. Only lists that contain at least 4 different domains are included.

3.2.6 Applicability of Blocking Approach

To summarize, in Publication III, we propose a traffic filtering methodology for wearable applications. Our solution is specifically designed to work for average users of consumer-level wearable applications. Since all studied applications contact at least 1 unnecessary third party, we believe the problem of disabling undesired communication for wearables to be of utmost importance. Any contact with an undisclosed third party may leak potentially sensitive information, making it vital to disable such communications.

Domain-based filtering. Previous studies have suggested that readily available blocklists may not be the optimal solution for some IoT devices [94] and smart TVs [151]. The authors of these works argue that such collections do not cover a significant number of unnecessary third parties, leading to low recall of such entities. However, trying to block everything raises the risk of encountering false-positive domains and can result in improper functioning of the devices and the companion applications. In our case, we prioritize ensuring that the apps function correctly and that the user experience for regular consumers does not deteriorate, even if it means missing potentially blockable entities. Furthermore, since many researchers and maintainers are investigating unwanted mobile

connections, domain-based blocklists likely contain many more unwanted third parties for wearables compared to other types of IoT devices that do not connect through companion apps. Our research findings suggest that, indeed, utilizing exclusively existing filtering lists is appropriate for fitness trackers.

Limitations. In real-world settings it is difficult to guarantee that blocking the traffic destinations intended by the developers will never cause applications to malfunction. Nevertheless, this issue is inherent to all content filtering approaches and involves a tradeoff between maximizing the identification of unnecessary connections and minimizing the potential to compromise the functionality.

3.3 What Can Regular Users Do?

While "jailbreak" apps [42, 45] are a somewhat applicable solution to mitigate attacks on encrypted traffic (RQ1 in Chapter 2), they are not really addressing the problem discussed in this chapter. Since custom jailbreak apps cannot be paired with any partner apps (for one, partner apps only pull data from the official servers of manufacturers), we recommend using the methodology outlined in this chapter.

Indeed, installing adblocker is not just extremely effective at disabling unwanted connections, but also can be set up by an *average* wearable user as shown in Figure 3.3. For example, the depicted adblocker AdAway[5] does not require root access and supports importing various blocklists via `URLs`.

[5]https://adaway.org/

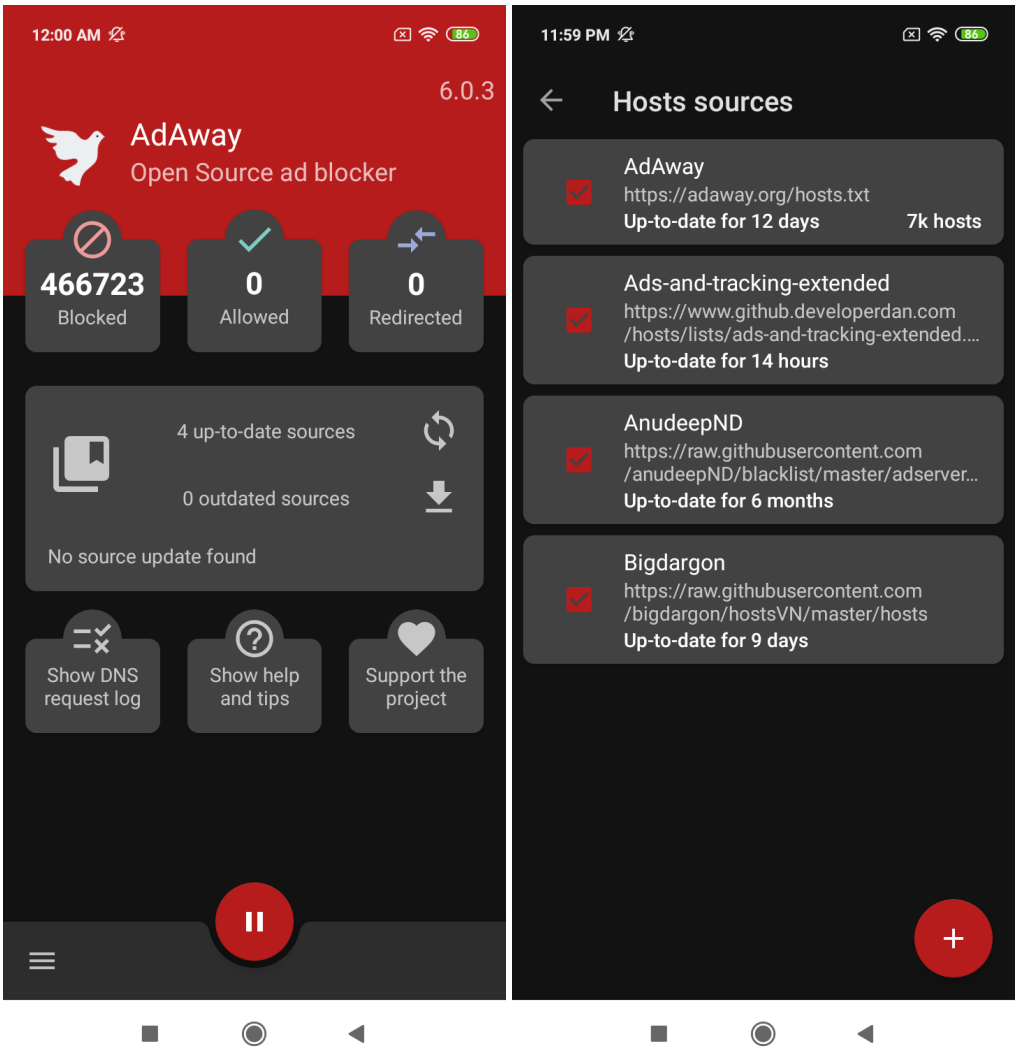

Figure 3.3: AdAway – an open-source adblocker that can be installed and operated by users with limited knowledge of Android. The right screenshot depicts importing 4 highest hitting blocklists for wearables as per Table 3.10 via their corresponding URLs.

Chapter 4
Attacks on Wearable Data

This chapter introduces 2 novel attacks against data collected by consumer wearables and evaluates their effectiveness (Section 4.3). We also review the ML approaches and performance measurements that we utilize for the attacks (Section 4.1). Finally, we discuss the datasets employed in our experiments. This chapter mainly builds on Publication IV, but a preliminary publication of ours on the topic (Publication b) is also addressed.

4.1 Methods and Performance Metrics

Data-driven approaches, which involve the analysis and interpretation of data, often result in more accurate decisions than those based on hand-crafted rules. These methods have been widely discussed in literature.

Classification Metrics. One of the fundamental tasks of data-driven algorithms is to perform a so-called classification of samples – assigning a class label to an example from the domain. In order to estimate how precise the algorithm can classify data samples, several metrics have been proposed over the years. The most fundamental one is called *Accuracy*, which, for *binary settings*, is the ratio of the correct predictions to the total number of predictions (as shown in Equation 4.1).

$$\text{Accuracy} = \frac{\text{\#of correct predictions}}{\text{\#of all predictions}} = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

Where TP are *True Positives*, TN are *True Negatives*, FP are *False Positives*, and FN are *False Negatives*.

Although *Accuracy* is widely used in data science, it may not always provide an accurate measure of an algorithm's success. For instance, in the context of TSA airport checks, predicting everyone as "not carrying anything dangerous" could result in almost 100% accuracy, but this approach could lead to a disaster since it would allow people with explosives, weapons, hazardous materials, and so on, to board.

Therefore, alternative metrics, including *Precision* – Equation 4.2, *Recall* – Equation 4.3,

and $F1$ score – Equation 4.4, have been introduced to address for imbalanced datasets and specific tasks to solve.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

Regression Metrics. In certain ML tasks an example from the domain must be assigned a numerical value instead of a categorical class. These methods are referred to as regression problems in the literature. None of the performance metrics previously described are applicable to these problems.

The most straightforward approach to estimating the effectiveness of a regression model is to calculate Mean Absolute Error (MAE), which is the sum of absolute errors divided by the number of data points. However, MAE has limited practical use since it does not penalize outlier predicted values enough.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{4.5}$$

A more balanced error estimation metric that is widely used in practice is called RMSE. Since RMSE is proportional to the size of the squared error, it is more sensitive to the infrequent large errors compared to MAE.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}. \tag{4.6}$$

In order to compare RMSE values calculated across multiple tasks, the value can be normalized (NRMSE) to provide a dimensionless estimate of error. In our work, we normalize RMSE with respect to difference between the maximum and the minimum values of the true observations, as depicted in Equation 4.7. Note that RMSE can be normalized with respect to different parameters, such as mean, standard deviation, and interquartile range (the difference between chosen percentiles).

$$\text{NRMSE} = \frac{\text{RMSE}}{y_{\max} - y_{\min}} \tag{4.7}$$

For Equations 4.5-4.7, n represents the number of data points, y and $\hat{y}$ are the true (actual observation) and predicted (estimation of observation) values respectively.

We utilize the following ML architectures for the tasks of both de-anonymizing users and inferring their physical parameters.

Random Forest (RF). RF is an ensemble of decision trees, which are organized hierarchically in the form of a tree of queries. Typically, specific noise is added to each decision tree during training in RF.

K-Nearest Neighbors (KNN). KNN is a non-parametric, centroid-based clustering classifier that uses the proximity to the other nearest data points as the basis for prediction. It assigns a group (cluster) to individual data samples.

Support Vector Machines (SVM). SVM is a ML algorithm that supports both classification and regression analysis. It is based on finding a hyperplane in an N-dimensional space to correctly classify as many data points as possible. This hyperplane is chosen based on the maximum distance between data points that are closest to it – so-called support vectors.

Kernel Density Estimation (KDE). KDE is a non-parametric algorithm to smooth a distribution of probability density estimation. Essentially, KDE centers and smooths a chosen kernel function at each data point.

Deep Neural Network (DNN) approaches. Unsurprisingly, we obtained the best results for most of our experiments with DNN approaches. Artificial Neural Networks (ANN) are inspired by biological neural networks, such as those found in mammalian brains. ANNs are a set of connected elements called artificial neurons that are based on the conception of multilayer perceptron. Unlike multilayer perceptron, DNNs use continuous activation functions, which allows differentiation with respect to parameters of the neurons – weights and bias – enabling the training of the whole network. A DNN is an ANN with one or more layers between the input and output layers (as shown in Figure 4.1).

A more complex model utilized in this thesis includes a Recurrent Neural Network (RNN) - based algorithm called Long Short-term Memory networks (LSTM). LSTMs are typically used for deeper processing of sequential data, especially that of long sequences. While the input for regular RNNs at a certain timestamp depends not only on the current input token but also on the hidden state from the previous timestamp, LSTMs employ more sophisticated mechanisms to retain sequential dependencies. Thus, an additional parameter is added to a timestamp cell – called cell state – which retains such dependencies in parallel with the hidden state. Finally, a total of 3 gates decide at every timestamp (i) what should be remembered from the preceding sequence, (ii) what should be learned from the current input, and (iii) what should be passed further to the next timestamp. These gates are:

- *Forget gate* quantifies how much of the information from the previous time step should be retained or "forgotten."
- *Input gate* is used to evaluate the importance of the new data provided as the input.

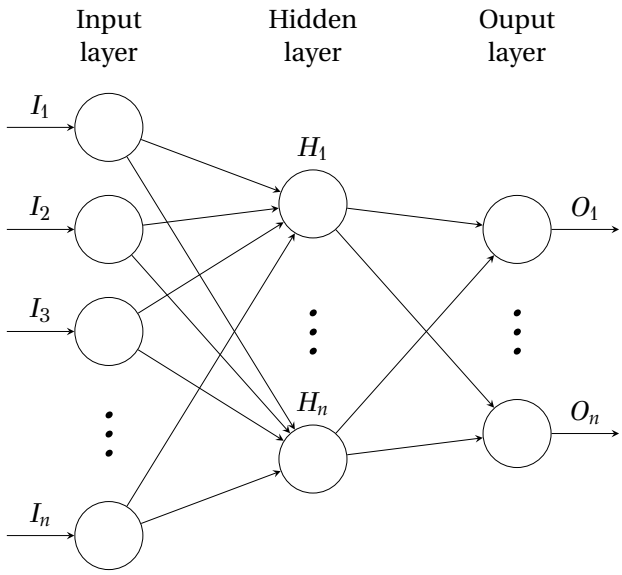

Figure 4.1: Depiction of a deep neural network used for a classification task. Number of features equals the number of neurons in the input layer $N = |(I_1, \ldots, I_n)|$. Number of classes equals the number of neurons in the output layer $N = |(O_1, \ldots, O_n), |$. This example network has 1 hidden layer, which implicitly encodes the impact of each feature. Adding hidden layers may improve the performance of the model, but increases the required computation for training.

- *Output gate* constructs the output.

An example of a LSTM cell is depicted in Figure 4.2.

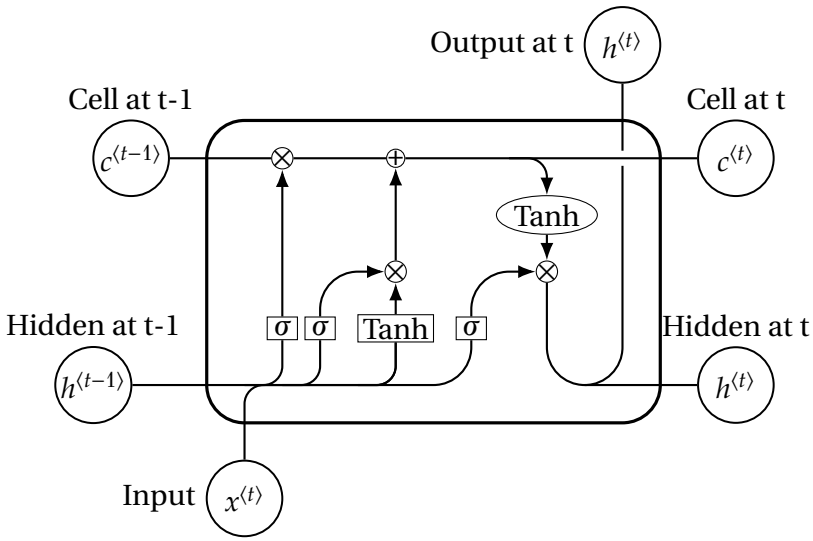

Figure 4.2: LSTM cell. $X^{\langle t \rangle}$, and $h^{\langle t \rangle}$ are the input and output at timestamp t. The hidden and cell states, denoted by h and c, respectively, are calculated at each timestamp and passed throughout the full sequence of inputs.

4.2 Datasets

This section summarizes the 3 open-source, publicly available wearable fitness datasets that we employ throughout this dissertation. Since all the data were collected with Fitbit fitness trackers (although different models of the devices), the published information follows the same format.

Openhumans.[1] The Openhumans dataset was collected by the online data sharing platform Open Humans and consists of data from 40 users who shared their data for a period ranging from 17 to 3509 days. Participant of this dataset are volunteers who have connected their Fitbit accounts to share data from Fitbit activity trackers or other Fitbit devices. Openhumans contains weight and height data of participants, allowing for the calculation of Body Mass Index (BMI). Since gender as such is missing, we reconstruct it from the unambiguous "nicknames" of the users. We discard the data of users who have unisex names for the gender inference part of our experiment. Furthermore, we remove participants who do not have any recorded data.

Crowd-sourced Fitbit datasets.[2] The Crowd-sourced Fitbit datasets dataset (CSFD) was generated via the Amazon Mechanical Turk crowdsourcing platform. This dataset includes 30 Fitbit users who consented to the submission of their fitness data. CSFD includes only weight, and height and has no record of participants' gender. We drop all the empty (0 daily

[1] https://www.openhumans.org/activity/fitbit-connection/
[2] https://zenodo.org/record/53894#.YMoUpnVKiP9

steps) entries, and remove users for whom we could not compute the BMI. The number of recorded days per user for this dataset ranges from 2 to 49 days.

PMData.[3] The PMData dataset was created during a 5-months life logging experiment, counting 16 users. It contains gender, height, weight, and age information for all participants except one (for them only the weight data is missing). Unlike Openhumans and CSFD, this dataset was produced during a controlled experiment, resulting in a similar amount of data for each user. After discarding empty time series, the number of recorded days per participant ranges from 80 to 152 days. Furthermore, PMData contains the data from athletes, rather than regular Fitbit users. All participants had been using the Fitbit Versa 2 wristband.

4.3 Threat Models

We investigate whether an adversary can re-identify a target user in a public wearable dataset, only using information they infer from solely the fitness data without employing any other personal identifiers.

De-anonymization based on physical parameters. The first threat model involves learning physical characteristics from wearable data and comparing the obtained results with real-world information. Specifically, we aim to determine if it is possible to identify the gender of users[4] and whether they are overweight based on their BMI. We chose to use a BMI threshold of 25 for our experiments since people with a BMI over 25 are typically considered overweight, assuming a normal body type. We further investigate the possibility of identifying individuals who are taller than the average male height in Europe, which is 177.6 cm [164]. Henceforth, when we use the term "overweight," we are referring to individuals whose BMI is greater than 25. Similarly, when we use the terms "tall" or "taller," we are referring to people who are above 177.6 cm in height, and when we use the terms "short" or "shorter," we are referring to individuals who are below 177.6 cm. To extract the insights on physical attributes from daily records, we train cross-dataset inference machine learning models, using (i) daily steps, (ii) distance, and (iii) calories as features. The models are trained using datasets described in Section 4.2. By utilizing a limited number of features, we aim to improve the usability of inference models and visualize the obtained results. Our research also examines the number of data samples required to correctly infer personal attributes. After the attacker learns the physical characteristics of all users in the dataset, they proceed to compare them with those of the targeted individual. If there is only one user with the same set of parameters as the victim, the adversary concludes that they are the target. However, if there are multiple users, say k, the attacker can only guess with a

[3] `https://datasets.simula.no/pmdata/`

[4] In this work by gender we imply the binary choice of male/female offered by Fitbit.

probability of $1/k$. Figure 4.3 provides more detailed information on this threat model.

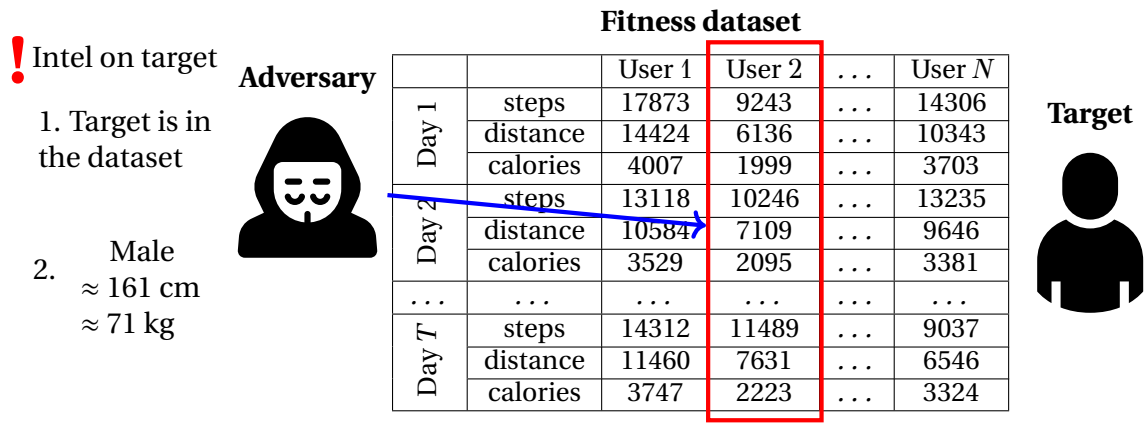

Figure 4.3: In the first threat model we consider, the adversary aims to link a person known to be in an aggregated dataset back to their fitness records. Assuming that the attacker has learned a basic profile of the victim, they infer physical attributes for all the users based on the daily fitness data, and chooses the most matching individual.

De-anonymization based on daily routine. In the second threat model, the adversary aims to de-anonymize users by analyzing their daily activity patterns. This approach differs from the first threat model, as the attacker does not need to know the physical parameters of the target in advance. However, the adversary must have access to external samples of the target's data (which are not part of the anonymized dataset) to re-identify them. These extra samples may be obtained through the target's medical records or through FTSN. For Fitbit, for example, it is possible to follow the activity progress of friends in the dedicated app. Furthermore, a significant number of Fitbit users belong to so-called fitness communities that allow sharing fitness data online. In this case, we also train machine learning inference models, where the final prediction indicates the person who produced the input data sample, effectively de-anonymizing them. For these models the features are wearable data time series of length 24, where each entry represents an hourly tuple (from 00:00 to 23:00) that contains (i) the number of steps taken, (ii) distance covered, (iii) calories burned, and (iv) average heart rate for that hour. These data are combined with information about the day of the week, distinguishing between weekdays and weekends to account for potential changes in routine on Saturdays and Sundays. Our threat models differ significantly, with the first one identifying users based on their identity, and the second one categorizing them based on their activities. The differences between the two are illustrated in Figure 4.4, emphasizing the unique aspects of our second threat model.

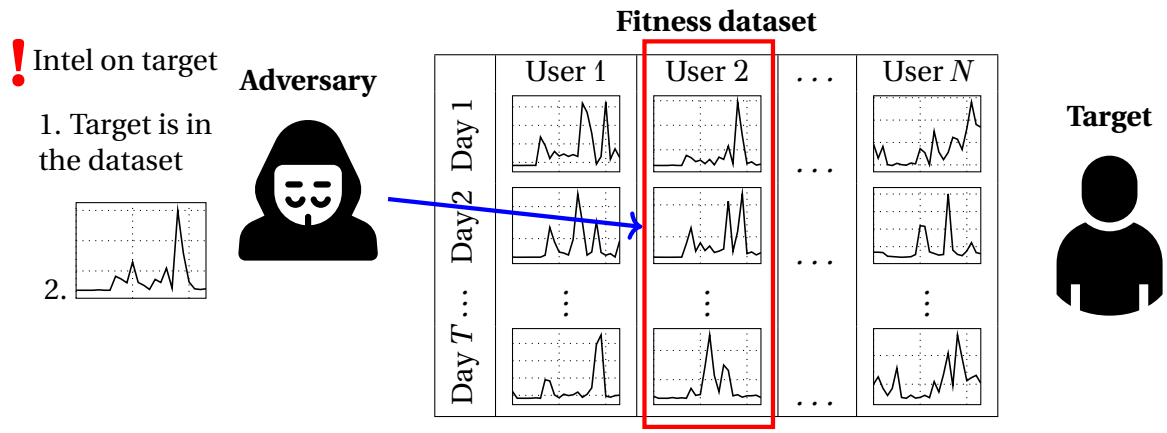

Figure 4.4: In our second threat model, instead of knowing personal attributes of the target, the attacker is in possession of additional victim's fitness samples. E.g., such extra data might be obtainable from social network posts (Fitbit communities). Time series are represented as 1-D for convenience, but are actually 4-D.

4.4 Setup

In this section, we describe the inference models, the hyperparameters used, and the data utilized for both identity- and routine-based inference.

4.4.1 Identity-based inference

We employ several well-known machine learning approaches to infer physical parameters of users. These approaches leverage training data to learn a map that takes the aforementioned features as input and outputs a binary answer (true or false) in response to queries about personal attributes.

In particular, the three binary maps that we learn are:

1. q_{gender}: "Is an individual a male?"
2. q_{BMI}: "Is the BMI of an individual above 25?"
3. q_{height}: "Does the height of an individual exceed 177.6 cm?"

We utilize the following machine learning approaches to learn the maps:

- DNN. We train a 2-layer fully connected deep network with early stopping. We utilize the following hyperparameters for training:

  - Architecture: 120 hidden neurons + ReLU; 60 hidden neurons + ReLU; 2 output neurons + Softmax.
  - Loss: binary cross-entropy.
  - Batch size: 64.
  - Optimizer: Adam, learning rate = 0.001.

- SVM. We utilize the RBF kernel with penalty coefficient $C = 1$ and smoothing parameter $\gamma = 1/(\text{number of features} \times \text{variance of the data})$.

- KNN. We fit the training data with $k = 5$.

User-wise Prediction. All the models that we present throughout this chapter infer binary information from a single daily sample. That is, they are all maps $q : \mathcal{X} \to \{0, 1\}$, where $\mathcal{X}$ is the domain of the features (i.e., all the possible combinations of steps, calories and distance) and $\{0, 1\}$ is the set of possible answers to a binary query (0 if the answer is negative, 1 if it is positive). However, a user producing a time series of daily samples $x = (x^{(1)}, \dots, x^{(T)})$ provides T samples on which a map can be applied. Also, since our models are not 100% accurate sample-wise, a map q will likely provide different predictions for different samples of the same users. A final prediction $\hat{r}$ for a time series x is made according to a majority rule, i.e.,

$$\hat{r} = \arg\max_{r \in \{0,1\}} \left| \left\{ x^{(t)} : q(x^{(t)}) = r \right\} \right| \tag{4.8}$$

In a binary-decision setting, this is equivalent to the simpler criterion

$$\hat{r} = \begin{cases} 1, & \text{if } \frac{1}{T} \sum_{t=1}^{T} \hat{q}(x^{(t)}) > \frac{1}{2} \\ 0, & \text{otherwise} \end{cases} \tag{4.9}$$

meaning that if the average of the binary answers is above 50%, we conclude that the most likely answer is positive, otherwise we conclude that it is negative.

Datasets in use. For gender inference – since the CSFD dataset does not have gender as a ground truth parameter – we utilize only PMData and Openhumans: Openhumans for training and PMData for testing. For the overweight, and tall people detection we train our models on the combination of the Openhumans and CSFD datasets, and test them on PMData.

Training procedure. We employ a 80/20 training/validation split for all the inference models. We apply 5-fold cross-validation when training, and select the best performing model.

4.4.2 Routine-based inference

We train an LSTM neural network that considers the day of the week as an extra categorical input, differentiating between weekdays and weekends. To achieve this, two bits – 01 for weekdays and 10 for weekends – are combined with the LSTM output, as shown in Figure 4.5. During training, we use the following hyperparameters:

- Architecture: as in Figure 4.5.
- Loss: categorical cross-entropy.
- Batch size: 64.
- Optimizer: Adam, learning rate = 0.001.

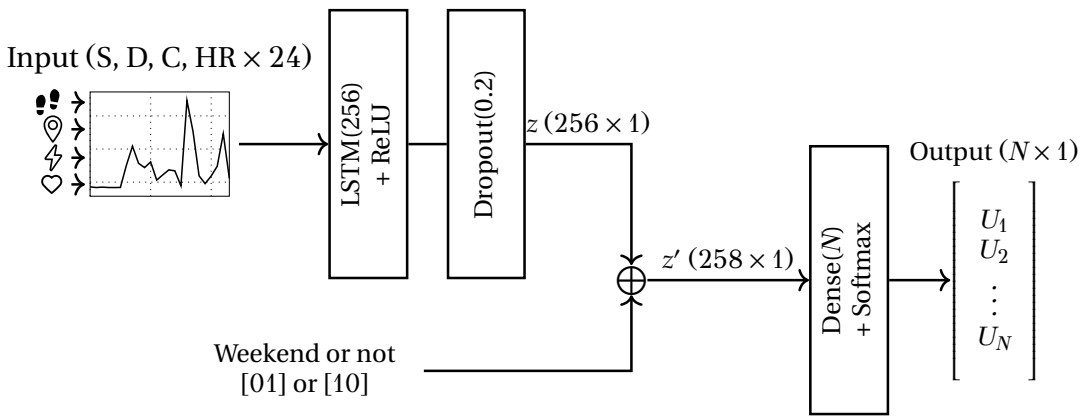

Figure 4.5: LSTM-based architecture for de-anonymization based on activity routine. Input consists of 24 tuples of S: Steps, D: Distance, C: Calories, and HR: Heart Rate that are measured every hour. Two bits are concatenated to the output of the LSTM layer to model weekdays: Monday to Friday (10) or weekends: Saturday and Sunday (01). The output corresponds to the probability of the input routine being produced by every user in the dataset.

Datasets in use. Since PMData has the most even spread of samples between all the users and a suitable number of samples for each user, we choose to use it for re-identifying individuals based on their daily routine.

Training procedure. We opt for an 80/20 split between training and validation data. Additionally, we apply 5-fold stratified cross-validation, selecting the best model.

4.5 Deanonymization Based on Physical Parameters

For the first threat model the corresponding de-anonymization attack goes as follows. We assume an aggregated dataset, consisting of N users $\theta_1, \ldots, \theta_N$, whose personal information is unknown. Each user is identified by a pseudonym, which distinguishes their time series of daily records from the others. The objective of the attacker within this threat model is to find a target user θ^* that they know to be among $\theta_1, \ldots, \theta_N$ and for whom the gender, and approximate values of height and BMI are known. If the user is identified, the attacker may learn some information based on their daily steps, calories and distance, e.g., exercise routines, whether they went to the office on a given day, etc. To do so, the attacker applies the three inference models to each of the N time series $x_1, \ldots, x_N$ present in the dataset. Before any prediction is made, all the users belong to a same *anonymity group* of size N, meaning that an adversary can guess the correct user with probability $1/N$. The adversary leverages the prediction models to answer three binary queries ($q_{\text{gender}}, q_{\text{BMI}}, q_{\text{height}}$), where each query splits a group into 2 subgroups. Therefore, 3 queries divide the dataset into $2^3 = 8$ anonymity groups. Depending on the queries, and on the population of the dataset, the subgroups may vary in size. A lower bound to the size of the largest subgroup is given by $\lceil N/8 \rceil$, implying that if N is greater than 8, it is impossible to identify all the users. Nonetheless, if the target belongs to a minority (e.g., a female in a dataset with prevalence of male users), it might be easy for an adversary to identify them.

4.6 Inference of Physical Parameters

This section summarizes the obtained results, and illustrates the most interesting findings. Tables 4.1, 4.2, and 4.3 depict the inference results for gender, overweight and height detection respectively. In these tables, we present the accuracy achieved by our models on the validation split and the PMData test split, which was not observed during training at any point. We also report the count of users correctly classified in the test split. A user is considered accurately classified if the model can correctly identify over 50% of the samples for that individual, as per Equation 4.9. This metric is crucial as it demonstrates the number of users for whom we can learn their physical parameters. Additionally, we report other metrics of interest, such as Recall, Precision, and $F1$ score for each of the label for all the problems.

It is evident that all the models perform considerably better on the task of gender detection, achieving higher user and sample classification accuracies. For the task of gender inference DNN outperforms all other models across all the observed metrics, reaching the perfect 100% accuracy, and classifying all 16 users correctly. SVM and KNN also perform well, and are both able to classify 14/16 users correctly. However, for the task of profiling overweight people the models do not reach similar higher accuracies. Nevertheless, DNN

Model	Val accuracy	Test accuracy	Labels	F1	Users	User accuracy
DNN	0.925	0.925	Male	0.96	13/13	**1.000**
			Female	0.78	3/3	
KNN	0.91	0.853	Male	0.91	12/13	0.875
			Female	0.63	2/3	
SVM	0.826	0.901	Male	0.95	12/13	0.875
			Female	0.68	2/3	

Table 4.1: Gender inference. *Test accuracy* is computed based on all the samples that have been classified, whereas *user accuracy* indicates whether most of the data samples for each user in the test have been classified accurately.

Model	Val accuracy	Test accuracy	Labels	F1	Users	User accuracy
DNN	0.81	0.731	Overweight	0.75	7/7	**1.000**
			not Overweight	0.71	8/8	
KNN	0.817	0.6	Overweight	0.63	7/7	0.667
			not Overweight	0.54	3/8	
SVM	0.689	0.732	Overweight	0.76	7/7	0.867
			not Overweight	0.67	6/8	

Table 4.2: Detection of overweight users. Similarly to gender inference, the most relevant metric is the user accuracy.

Model	Val accuracy	Test accuracy	Labels	F1	Users	User accuracy
DNN	0.968	0.821	Tall	0.88	12/12	**0.938**
			Short	0.66	3/4	
KNN	0.939	0.654	Tall	0.74	10/12	0.813
			Short	0.49	3/4	
SVM	0.839	0.655	Tall	0.74	9/12	0.688
			Short	0.47	2/4	

Table 4.3: Detection of users beyond the height threshold. Height inference models are unable to attain perfect user classification accuracy, which sets them apart from earlier binary queries.

is still able to classify *all* 15 users correctly, despite having only 73% accuracy of identifying the individual data points. Relatively low sample accuracy might be attributed to the fact that many users from the PMData dataset are very close to the 25 BMI margin, as depicted in Figure 4.6. As for the non-neural network models, KNN performs better than other models on the validation split, but struggles with classifying previously unseen users. SVM

is comparable to DNN on the test dataset when classifying separate data points, but is slightly worse at identifying the actual users.

Regarding detection of tall people, the best results are obtained with the DNN model likewise, achieving 93.8% accuracy. However, for this problem the best model does not attain the perfect classification accuracy, misclassifying one user in the test collection. Traditional machine learning approaches perform reasonable on the validation test, achieving 81.3% user accuracy for KNN, and 68.8% for SVM. Similarly to overweight detection, for this dataset height is also a non-binary parameter. Hence, it is significantly more challenging to classify users whose physiological parameters are close to the classification threshold, as can be observed in Figure 4.6. In general, it is clear that both the amount and quality of the data enables the precise predictions of physiological parameters for users who have not been previously seen during training.

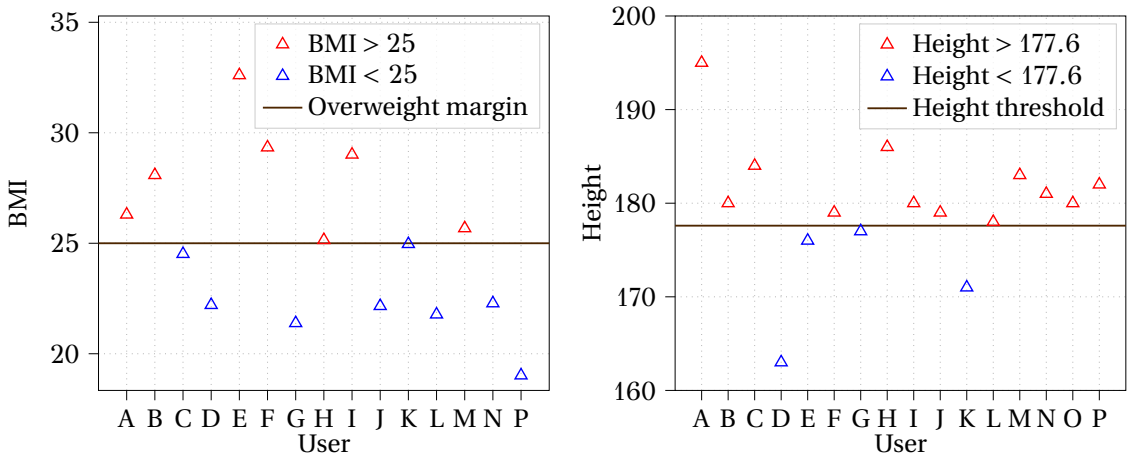

Figure 4.6: BMI of users in PMData (left) and their height (right). For the test set at least 4 users are within less than "1 BMI" of the overweight threshold. Only 4 users (including females) are shorter than the average male height in Europe.

4.6.1 Inference Visualization

As we only use three attributes (steps, distance, and calories) in our models, we can create 3D graphs that show the decision regions for each binary query, indicating the predicted labels for various combinations of these attributes. To make these regions more visible, we evaluate rectangular grids of steps and distance for different fixed calories values. This way, we obtain the "layered" regions that can be observed in Figure 4.7 for the gender model, Figure 4.8 for the BMI model, and Figure 4.9 for the height model, where the layers are evaluated every 250 calories. The axis for each feature ranges from 0 to mean(feature) +

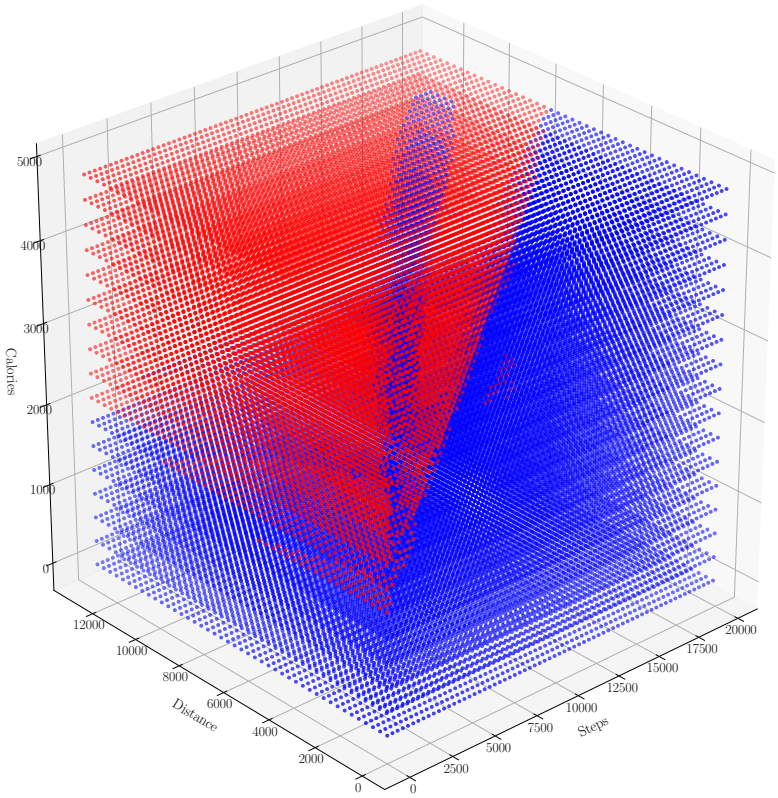

Figure 4.7: Gender inference decision regions. Red color indicates the male areas; blue color corresponds to females. Males tend to burn more calories per same activity, and have a higher ratio of distance to steps.

$2 \times$ stddev(feature).

The Harris-Benedict (HB) equations provide two separate empirical formulas for estimating the daily basal calories burned by females and males [57], which refer to the number of calories required for basic metabolic functions without exercise. Figure 4.10 depicts the basal calories of all the users in the test PMData dataset. It shows that the three females in

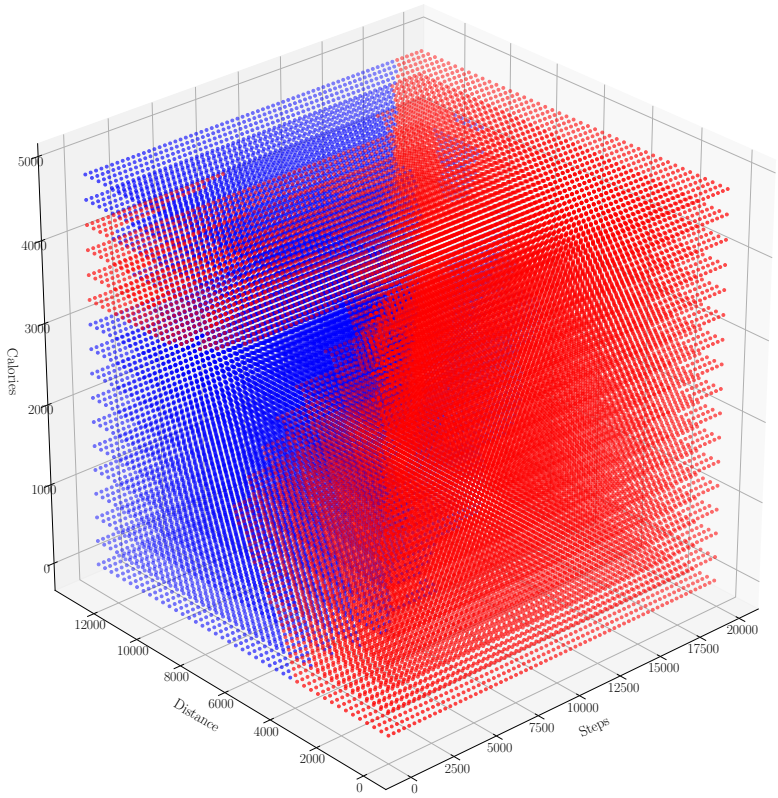

Figure 4.8: Decision regions for the detection of overweight people. Red color corresponds to the overweight predictions; blue color indicates non-overweight areas. Overweight people achieve the same number of daily calories with less daily activity.

the dataset burn significantly fewer calories compared to the males. The gender inference model (Figure 4.7) also conforms to this pattern, with blue points representing females and red points representing males. Indeed, very few male samples fall below the 1500 calorie hyperplane, which is in agreement with the HB equations. Furthermore, it can be observed

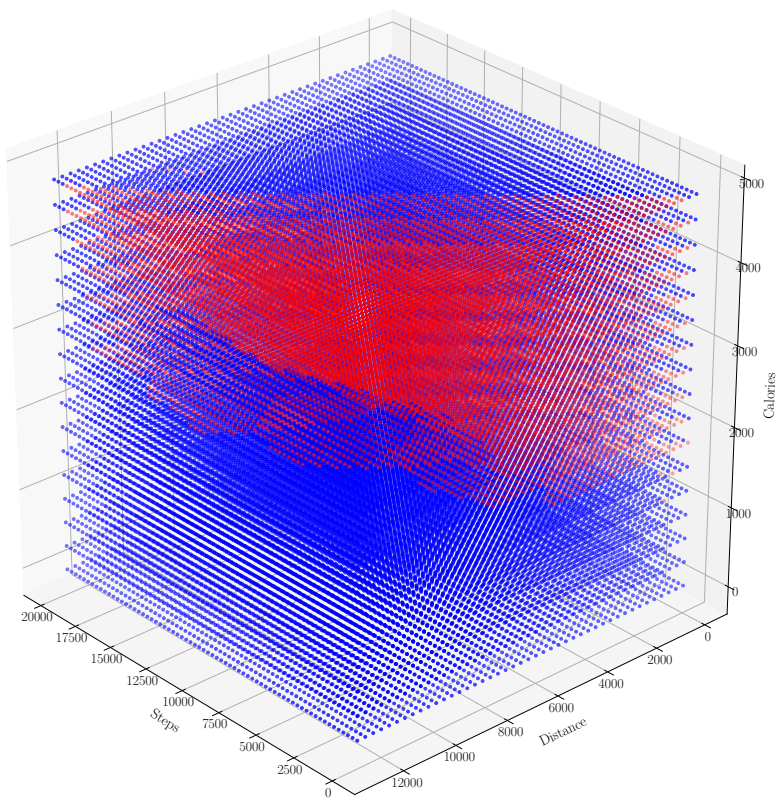

Figure 4.9: Decision regions for detecting people above 177.6 cm. Red regions correspond to taller people, while blue areas represent shorter users. Taller people have bigger stride, and rarely burn less than 2000 calories daily.

that areas with the same number of calories, but a higher ratio of steps to distance, generally correspond to female users. This may suggest that females, being typically lighter, need to take more steps and travel a longer distance to burn the same number of calories as males.

Regarding the detection of overweight users, the decision regions for the inference model are depicted in Figure 4.8. We use the red color to illustrate overweight people

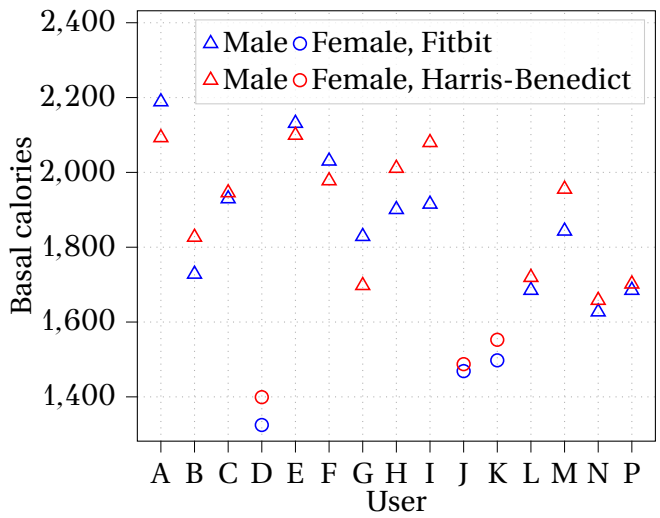

Figure 4.10: Depiction of the Harris-Benedict equations for the studied test dataset PMData. Blue points correspond to the basal calories as recorded in the dataset, while red calories are calculated from the equation. It appears that the estimation of basal calories for Fitbit closely follows HB. Since the users 4, 10, and 11 are females, no males appear to burn less than 1600 calories per day (even when they take no steps).

and the blue color to represent non-overweight users. It can be observed that, for the same number of burned calories, non-overweight people tend to take more steps/distance. This can be explained by the fact that heavier people burn more calories according to the HB equations, and as a result, they have to do less exercise to achieve the same daily number of calories. Furthermore, the image shows that the reason the labels are not almost perfectly separated by a single hyperplane is due to a number of red (overweight) outliers for daily calories that exceed 3500. We believe that this is because there are not enough non-overweight users in the training data who consistently burned that many calories.

The decision regions for detecting tall people are illustrated in Figure 4.9. As the chosen threshold of 177.6 cm represents the average height of males in Europe, it is expected for them to be outnumbered by other groups, such as shorter males and females who are below 177.6 cm. Hence, the volume of shorter people significantly exceeds the areas of their taller counterparts. The image suggests that taller people tend to cover more distance with the same number of daily steps, which appears to be appropriate. Furthermore, as taller people are expected to weigh more and therefore burn more calories, the inference model has estimated that they rarely burn less than 2000 calories per day, even with minimal daily activity. This assumption is heavily reinforced by empirical data.

4.6.2 Incomplete Records Deanonymization

As expected, the accuracy of revealing personal attributes depends on the number of available fitness samples per user. In the previous analyses, the adversary utilizes all the data per user to perform the de-anonymization attack. In practice, however, users in a wearable dataset may have as little as 10 daily fitness records. We set out to investigate whether our attack still performs well in case of *limited data*. To do so, we run a 1000-round Monte Carlo simulation for different number of samples per user. We still utilize the PMData dataset for this experiment. The experimental pipeline goes as follows:

- We restrict the number of fitness samples per user to the interval $I = \{1, \ldots, \bar{T}\}$, where

$$\bar{T} = \min_{i=1,\ldots,N} T_i = 80 \tag{4.10}$$

  is the number of records for the user with the least amount of data.

- We randomly draw T samples per user for each value of $T \in I$, and predict them. We utilize the majority rule on those T predictions to establish the final estimated value for that user as in Equation 4.9 (i.e. choose the most frequent prediction for that user). We repeat the procedure for $n = 1000$ rounds and average the obtained results.

- We repeat the above process for every user in the dataset.

The obtained results are illustrated in the Figures 4.11. The classification results for the individual labels are depicted in red and blue, while the combined (total) accuracies are represented by the black lines.

Moreover, it appears that, overall, increasing the number of samples per user leads to higher user classification accuracies, as expected. Furthermore, it seems that to achieve higher identification accuracies for the minority labels of the test dataset (females, non-overweight, and short), more samples for such users are required. Such a tendency holds across all three personal attribute inferences, and can be explained by the fact that the test dataset consists of athletes who tend to produce time series containing more daily activity and, hence, more steps taken, calories burned, etc. Nevertheless, for all attributes except height, the identification accuracy eventually reaches 100%.

The obtained figures do not appear smooth, with final accuracies slightly declining for every even value of samples per user. These spikes are caused by our approach to defining a successful classification of users. That is, we count a user to be identified correctly only if strictly more than half of their samples are accurately classified. Thus, ties between correct and incorrect predictions are interpreted as a misclassification. As ties are impossible for odd-numbered sample counts, the accuracy does not decline in such cases.

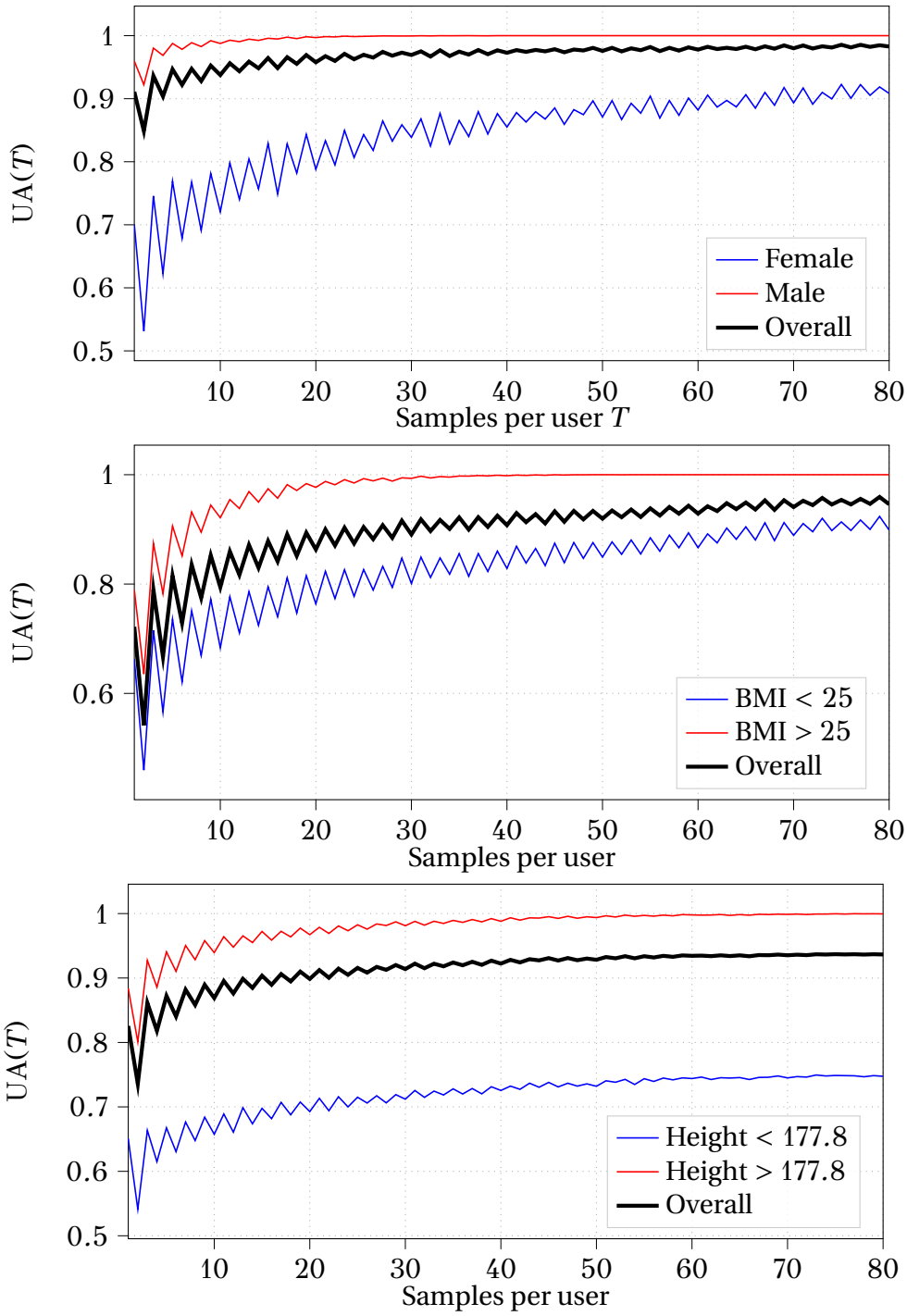

Figure 4.11: Accuracy of predicting gender (top), overweight users (middle), and height (bottom) with limited samples per user .

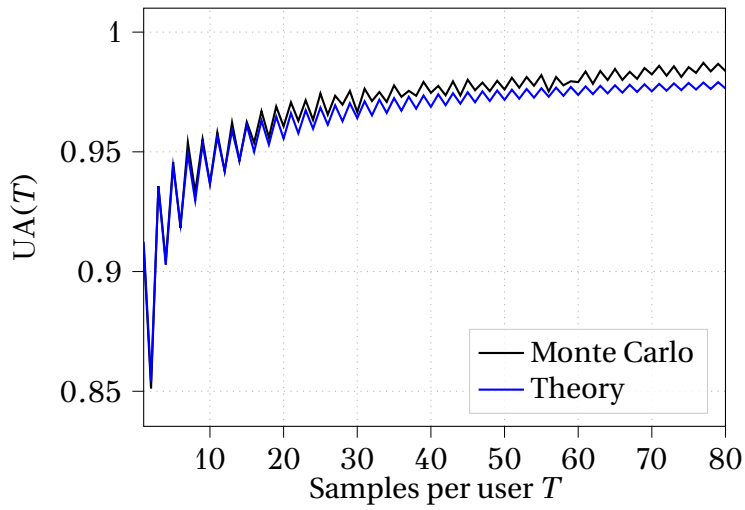

Figure 4.12: Comparison between theoretical analysis and Monte Carlo simulation for gender prediction. The empirical results show a strong correlation with the theoretical estimation.

Theoretical analysis. We show that the behavior of the incomplete-samples curves is correct by comparing them with a theoretical analysis of the majority rule. The objective of this analysis is to estimate the probability for an adversary to correctly predict a binary characteristic using the majority rule, given a time series of T samples. We assume that for user θ_i the predictions are independent and have all probability p_i. The value of p_i is estimated as the fraction of samples from that user that are correctly classified. However, this assumption does not hold in our Monte Carlo simulation, since we have a finite number of samples per user, and drawing a sample changes the distribution of the remaining ones. Nonetheless, it still provides a good approximation.

For user θ_i, the probability of correctly predicting the entire time series is the probability of correctly predicting more than half of the samples. This probability is given by

$$\sum_{t=\lfloor T/2+1 \rfloor}^{T} \binom{T}{t} p_i^k (1-p_i)^{T-t} \tag{4.11}$$

that is the complementary cumulative distribution of a binomial random variable computed at $T/2$, i.e., $\Pr(\mathrm{Bin}(T, p_i) > T/2)$. The overall accuracy is calculated by averaging the accuracy for all users, assuming they are chosen with equal probability. Figure 4.12 shows a comparison between curve (as in Equation 4.11) and the empirical results obtained from the Monte Carlo simulation for gender.

4.6.3 Utilizing Additional Fitness Features.

One may wonder what is the reason behind not using more features for inference of personal attributes from fitness data. Indeed, all three datasets that are employed in this work share a number of additional characteristics, such as heart rate, sleep, and 4 types of daily activity minutes, ranging from very active to sedentary. While the daily activity level might not be directly correlated to the first two of the above features, it can be clearly linked to the latter ones. The primary reason for not utilizing them is to make our inference models more adaptive and usable, as mentioned earlier, by focusing on the most relevant features for the task. Furthermore, most of the online posts that share fitness data do not contain any information other than triplets of steps, distance, and calories. In fact, even when the users share such information, it is never provided as 4 separate activity minutes features. Regarding daily minutes, shareable data consists of either the sum of "very" and "moderately" active minutes or is represented by so-called "zone" minutes, which are calculated based on specific rules[5]. Moreover, while these features are common to all data collections, it is worth noting that some participants have no records of daily activity minutes, making this aspect of the data unusable.

In this section, we enrich our inference models by incorporating additional input features and discuss the resulting outcomes. However, we emphasize that our primary findings pertain to the simpler models described earlier. Table 4.4 illustrates the impact of adding features that are consistent across all studied datasets on validation, test, and user accuracies.

We train the neural network models using the same architecture and incorporating three additional sets of features: (i) all three types of active minutes, (ii) their sum, and (iii) the sum of very and moderately active minutes. The obtained results suggest that adding more features leads to higher validation accuracies. However, this trend is not consistently observed for the test data, possibly because the users in PMData tend to be more active, as depicted in Figure 4.13. There, very active minutes typically correspond to intense workouts, while lightly active minutes are recorded during regular walks.

It appears that the optimal feature combination for the BMI and height inference does not include any types of daily activity minutes, as the models fail to generalize well on unseen users when such features are included.

[5] `https://help.fitbit.com/articles/en_US/Help_article/1379.htm`

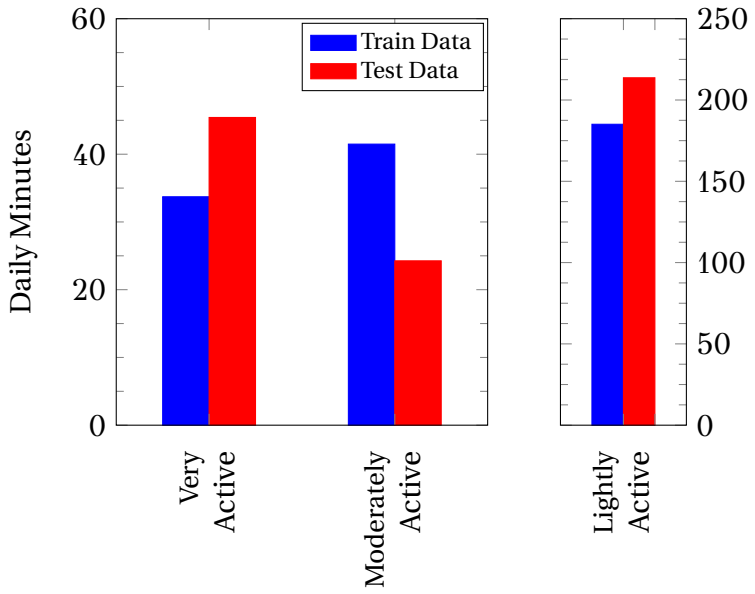

Figure 4.13: Average values of daily minutes for train and test users ordered by the most active first. Lightly active minutes are downscaled by a factor of 4.17.

Attribute	Extra Features	Val acc	Test acc	Labels	Precision	Recall	F1	Users	User acc
Gender	None	0.925	0.925	Male	0.94	0.97	0.96	13/13	1.000
				Female	0.83	0.73	0.78	3/3	
	V + M	0.942	0.934	Male	0.93	0.99	0.96	13/13	1.000
				Female	0.93	0.71	0.80	3/3	
	V + M + L	0.948	**0.949**	Male	0.97	0.97	0.97	13/13	1.000
				Female	0.88	0.85	0.86	3/3	
	V, M, L	**0.96**	0.939	Male	0.94	0.99	0.96	13/13	1.000
				Female	0.92	0.74	0.82	3/3	
Overweight	None	0.81	**0.731**	OW	0.69	0.8	0.75	7/7	1.000
				not OW	0.65	0.71	0.71	8/8	
	V + M	0.856	0.725	OW	0.73	0.75	0.74	7/7	1.000
				not OW	0.72	0.7	0.71	8/8	
	V + M + L	0.865	0.611	OW	0.64	0.59	0.61	4/7	0.733
				not OW	0.59	0.64	0.61	7/8	
	V, M, L	**0.903**	0.652	OW	0.65	0.73	0.69	7/7	0.867
				not OW	0.57	0.61	0.61	6/8	
Height	None	0.968	**0.821**	Tall	0.87	0.89	0.88	12/12	**0.938**
				Short	0.67	0.64	0.66	3/4	
	V + M	0.971	0.778	Tall	0.92	0.76	0.83	11/12	0.875
				Short	0.57	0.82	0.67	3/4	
	V + M + L	0.959	0.813	Tall	0.87	0.87	0.87	11/12	0.875
				Short	0.67	0.66	0.66	3/4	
	V, M, L	**0.974**	0.759	Tall	0.86	0.8	0.83	11/12	0.875
				Short	0.56	0.66	0.61	3/4	

Table 4.4: Inference results, given additional features: very active minutes (V), moderately active minutes (M), lightly active minutes (L). When we say V + M, we imply that the sum of the following parameters corresponds to a *single* feature. Additional features improve the validation results for all the models, but, generally, decrease the performance on the test data.

4.7 User Deanonymization

We use the PMData benchmark dataset to establish the number of individuals who can be uniquely re-identified. PMData contains ground truth information on gender, height, and BMI for all the users except participant 'O', whose BMI logs are unavailable. Table 4.5 summarizes the personal characteristics of the 16 users in the dataset. However, we exclude participant 'O' due to missing information and apply our method to the remaining 15 users.

The attacker leverages the queries previously introduced ($\hat{q}_{\text{gender}}$, $\hat{q}_{\text{height}}$ and $\hat{q}_{\text{bmi}}$) to split the dataset population in buckets and detect outlier participants. Assuming that the adversary gets the correct results for all queries, they can de-anonymize the minority users in PMData. The results are presented in Table 4.6. For every possible combination of queries (parameters), we display the number of users (k) who share such parameters. While users who share their buckets with other participants are somewhat safe, as the probability of their re-identification is now $1/k$ (which is still higher than a random guess), some individuals may still be uniquely de-anonymized. For example, participant J is the only one who matches the criteria of being a tall non-overweight female. Overall, the attacker is able to de-anonymize 3 minority individuals with 100% probability based on their physical attributes in PMData.

ID	Name	Gender	Height	BMI
p01	A	male	195	26.3
p02	B	male	180	28.4
p03	C	male	184	24.2
p04	D	female	163	22.2
p05	E	male	176	32.6
p06	F	male	179	29.5
p07	G	male	177	21.4
p08	H	male	186	25.1
p09	I	male	180	28.9
p10	J	female	179	22.2
p11	K	female	171	24.9
p12	L	male	178	21.7
p13	M	male	183	25.7
p14	N	male	181	21.9
p15	O	male	180	–
p16	P	male	182	19.3

Table 4.5:
Physical parameters of the users in the PMData dataset. Users are named with alphabet letters for more convenient referencing.

User	$\hat{q}_{\text{gender}}$	$\hat{q}_{\text{height}}$	$\hat{q}_{\text{bmi}}$	#
	male	> 177.6	> 25	6
	male	> 177.6	< 25	4
	female	< 177.6	< 25	2
E	male	< 177.6	> 25	1
G	male	< 177.6	< 25	1
J	female	> 177.6	< 25	1

Table 4.6:
Number (#) of users, sharing sets of physical parameters in the PMData dataset. Those who are re-identified with probability 1 are reported and highlighted.

4.8 De-anonymization Based on Daily Routine

In a preliminary work of ours [96] (Publication b) we establish that users in a wearable dataset can be re-identified based on the combination of daily calories and steps. By utilizing classical ML techniques, including KNN, RF, SVM, and KDE, we were able to achieve ≈ 80% accuracy (Figure 4.14). Although this is a decent number, we set to investigate how adding more features and increasing their granularity would impact user de-anonymization.

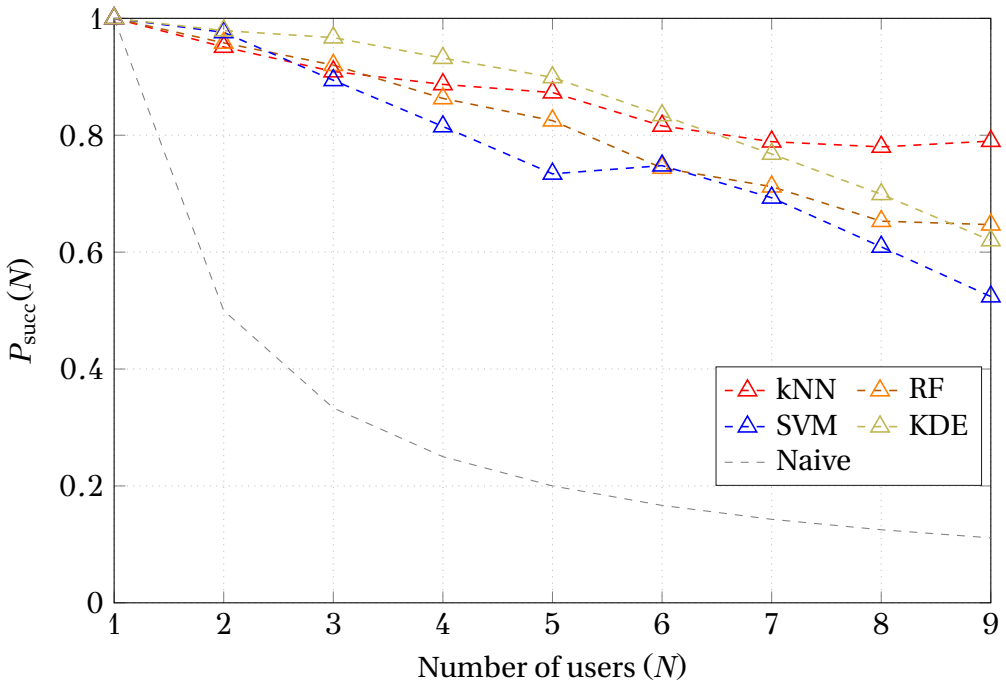

Figure 4.14: Re-identification results for PMData obtained in our previous works with simpler models and less granular data: daily snippets of steps and calories.

By utilizing the architecture from Figure 4.5, we are able to achieve a 93.5% de-anonymization accuracy for the full 16-user PMData. Again, we utilize time series of hourly (i) steps, (ii) distance, (iii) calories, and (iV) average hourly heart rate as features. Furthermore, in Figure 4.15 we report the re-identification results for fewer participants and compare the trend with that of our previous work (Publication b). We run a Monte Carlo simulation, where for every number of users N, ranging from 2 to 15, we perform 10 rounds of the experiment. We randomly select N participants from PMData, and train the inference model. Afterwards, we average the results for each value of N to obtain a final accuracy estimation. Based on the extrapolation of our findings, it can be assumed that de-anonymization attacks may still be successful even with larger datasets.

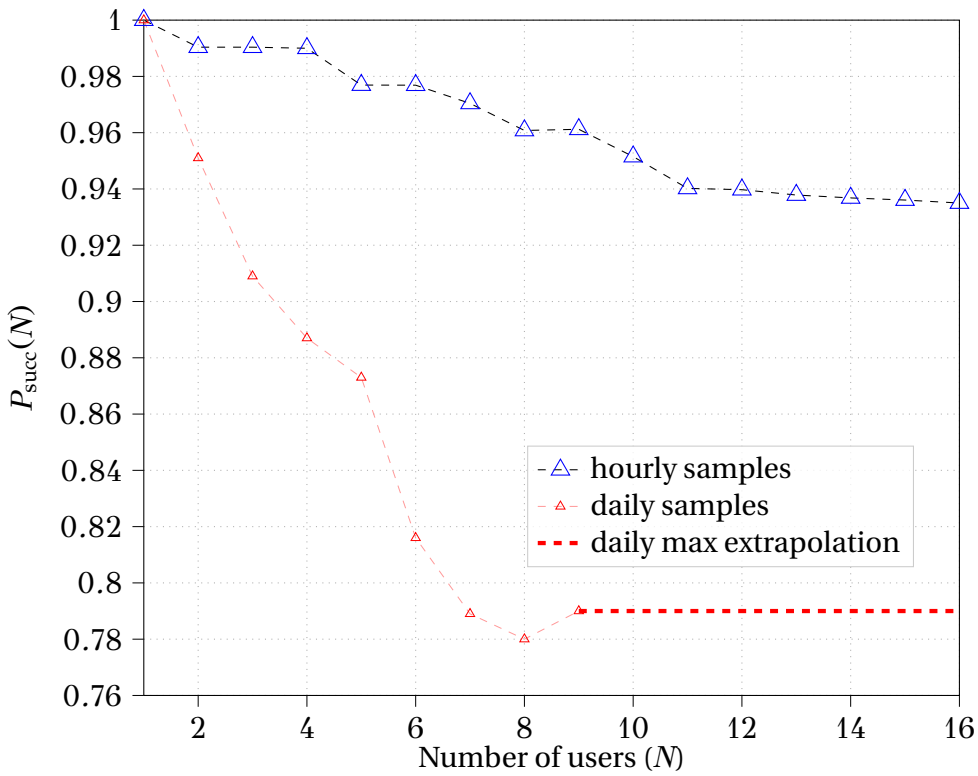

Figure 4.15: De-anonymization of users in the PMData dataset based on daily activity patterns. The LSTM-based model trained on hourly data heavily outperforms the highest performing previous model (KNN with $k = 1$) with the best possible extrapolation for $N = 10, 11, \ldots, 16$.

4.9 What Can regular Users Do?

Since the adversary infers the insights directly from wearable data, traditional anonymization methods that sanitize the direct and indirect identifiers, such as k-anonymity, are not effective against the proposed attacks. A number of recent works [18, 92, 93] incorporated adversarial learning approaches to dynamically sanitize the raw sensor data collected by accelerometer and gyroscope. While the proposed approaches were effective at protecting against inference of physical parameters, the studies were conducted with very limited data and did not evaluate generalization of their models.

We provide detailed guidelines for privacy-preserving wearable data publishing in the next chapter (Chapter 5). In summary, users can choose to opt-out of fitness studies to avoid potential privacy breaches. They can also share their data under pseudonyms when possible and disclose only the most relevant fitness information.

Chapter 5
Privacy-preserving Release of Wearable Data

This chapter suggests defense principles that both regular wearable users and wearable data controllers may adhere to. Summarizing the privacy leaks of wearable data discovered in Chapter 4, we outline practical defense strategies in Section 5.2 We present the wearable dataset collected within our research consortium in Section 5.3. This chapter is mainly based on Publications V-VI in relation to the attack vectors described in Publication IV.

5.1 Background

Data controllers often need to share personal data of private individuals with third parties or even make parts of it public. However, directly disclosing personal data of users can result in ethical concerns, breaches of internal privacy policies, and violations of privacy laws that have been recently enacted. Such laws include the General Data Protection Regulation (GDPR) [27], which came into effect on May 25, 2018, and the California Consumer Privacy Act (CCPA) [105], which went into effect on January 1, 2020. CCPA was amended by the California Privacy Rights Act (CPRA), commonly referred to as CCPA 2.0, on January 1, 2023. These privacy laws guarantee consumers more control over how their personal information is being used. In particular, user may:

- Opt out of the sale for their personal data (CCPA).
- Request their data to be deleted, commonly known as the right-to-be-forgotten (CCPA and GDPR).
- Find out what data are collected on them, the purpose of the collection, and whether these data are shared (all CCPA and GDPR).
- Exercise their right to privacy and not be discriminated against for that (CCPA and GDPR).

- Obtain the collected information in a human-readable, portable format and be able to transmit these data elsewhere without obstruction (CCPA and GDPR).
- Object to any decision made without human interference based on their data (GDPR).

Overall, data protection laws have greatly limited the ability of data controllers to disclose personal data of individuals.

Hence, personal information of users cannot be shared "as is," and data controllers need to find other ways. Typically, the information held by data controllers contains both sensitive data of individuals and information that identifies them. These identifiers can be divided into direct ones, which uniquely identify the individual, such as names, telephone numbers, email addresses, Social Security numbers, etc., and indirect or so-called quasi-identifiers, such as height, weight, eye color, body composition, etc. The naive way to "anonymize" a dataset is to remove all direct identifiers that uniquely identify participants. This process is known as pseudonymization, where all the real names are replaced with pseudonyms. In Figure 5.1 we depict a "toy" example of such a dataset with various types of personal data, including pseudonym (UserID), quasi-identifiers (gender, age, race), sensitive information (vaccine status), and non-sensitive data (number of steps). Note that what may seem like non-important data at first glance may lead to de-anonymization of users. For example, a well-known study by Sweeney established more than 50% of the US population to be uniquely identifiable by only (i) place of birth, (ii) gender, and (iii) date of birth, using the 1990 US Census summary data [142]. Moreover, researches found that a person may be re-identified among 1.5 million individuals based on hourly cellular data, including location, and corresponding carrier antenna information [31]. In particular, only 4 spatio-temporal data points are enough to uniquely identify 95% of users. In another study, De et al. showed that 90% of users may be re-identified based on merely 4 spatio-temporal samples of credit card metadata [32].

Data *de-identification* has been a cornerstone procedure for releasing personal data for the past few decades. De-identification is the process of masking the data to separate them from the individual who is associated with the personal records (data subject). De-identification of already collected information can be achieved through (i) generalizing and (ii) randomizing. In the next sections we review staple approaches for data de-identification that are mentioned throughout our works.

5.1.1 Anonymity

The privacy of the pseudonymized dataset (Figure 5.1) can be easily compromised, if an adversary has access to auxiliary records of a person/persons where quasi-identifiers are present. Naturally, such de-anonymization works only if a target is known to be included in both the original and auxiliary datasets.

Quasi-identifiers

Pseudonym

UserID	Gender	Age	Race	Vaccine	Steps
1	M	29	White	No	8421
2	F	31	Asian	Yes	4256
3	F	32	White	Yes	4311
4	M	25	Black	No	8848

Non-sensitive data

Sensitive data

Figure 5.1: Pseudonymized dataset. In this example, if quasi-identifiers of individuals are known to the attacker (e.g., via auxiliary knowledge) all of them can be re-identified. Note that despite data being marked as "non-sensitive," they may still lead to de-anonymization in some cases. In fact, the steps data may reveal significant insights on users.

***k*-anonymity.** One of the most know techniques to prevent re-identification based on auxiliary data is known as k-anonymity [124, 143]. k-anonymization is a data generalization technique that can be applied once direct identifiers have been suppressed. A dataset is considered to be k-anonymous if information on each participant is indistinguishable from that of at least $k-1$ other people in the dataset. We apply k-anonymity to the dataset from Figure 5.1 in Figure 5.2. Even if an adversary has access to auxiliary university records of dataset participants, since every set of quasi-identifiers is indistinguishable from another record in the table, the dataset is 2-anonymous. The modified dataset contains 2 anonymity sets (equivalence classes).

However, assuming that sensitive attributes for all users in the anonymity set have the same value, the adversary can learn this attribute without fully de-anonymizing the individual. For example, in Figure 5.2, Lea belongs to the equivalence class where all people have been vaccinated – this information can be learned, regardless of which particular record in the anonymity set corresponds to her.

2-anonymous Dataset

UserID	Gender	Age	Race	Vaccine	Steps
1	M	25-32	X	No	8421
2	F	25-32	X	Yes	4256
3	F	25-32	X	Yes	4311
4	M	25-32	X	No	8848

Equivalence classes

Edu.	Gender	Age	Race	Name
M.S	M	29	White	Andy
Ph.D.	F	31	Asian	Lea
Ph.D	F	32	White	Eva
B.S	M	25	Black	Tom

Auxiliary University Records

Figure 5.2: 2-anonymous dataset that does not satisfy l-diversity and t-closeness. Every set of quasi-identifiers can be confused with that of at least another user. Nevertheless, if there is not enough diversity in the sensitive attributes per anonymity set, the adversary is able to learn the vaccine status of the target. Assuming that "Steps" is also a sensitive attribute, the attacker can learn that Lea is relatively inactive ($< 5,000$ steps), which means that t-closeness is not satisfied.

l-diversity. An extension of k-anonymity, l-diversity [89] discusses the potential limitations of the former, addressing the scenarios when there is lack of diversity for the sensitive attributes. l-diversity enforces every anonymity set to contain at least l unique values for the sensitive attribute. A dataset satisfies l-diversity only if every anonymity set is l-diverse. In our example, if every equivalence class is changed to contain both unvaccinated and vaccinated individuals, the dataset becomes 2-diverse.

However, even a dataset that is both k-anonymous and l-diverse may leak sensitive insights. For example, assume that "Steps" becomes a sensitive attribute for the example dataset in Figure 5.1. Now, despite the modified dataset being k-anonymous and l-diverse with respect to steps, the adversary may still be able to extract the possible step range for Lea. With only $4,256 - 4,311$ daily steps, she is well below the normal activity levels.

t-closeness. l-diversity was further extended by t-closeness [81]. This approach ensures that the distance between the distribution of a sensitive attribute in the equivalence class and in the whole dataset does not exceed t – a settable privacy parameter. A dataset satisfies t-closeness if every anonymity sets have t-closeness.

5.1.2 Differential Privacy

For cases when some statistics on the users is released instead of the full dataset, different anonymization techniques need to be applied. If the data controller "honestly" discloses statistical information, the privacy of users may be compromised. For example, if the dataset in Figure 5.1 is controlled by a company, who releases the vaccination data on the employees. Assume that a person known to the adversary (Tom) is furloughed and the new data on vaccination is released (Figure 5.3). In that event, by comparing the outputs for both versions of the dataset, the attacker could learn that Tom is unvaccinated.

Differential privacy (global). To insure the safe release of statistical information on users Differential Privacy (DP) may be applied. DP is a well-established randomization technique for releasing sensitive information [35, 36], once direct identifiers have been suppressed. The main idea of the approach is based on introducing noise to the data to ensure that none of the users individually has a significant impact on the whole dataset. If DP is achieved, the removal or addition of an extra user would not change the result of the queries applied to data beyond a fixed ε coefficient. DP, as such, is independent of quasi-identifiers in the data or any auxiliary knowledge possessed by the adversary, since the anonymized dataset is not being released. Instead, the data controller provides a response to the statistical queries about the data.

More formally, DP can be defined as follows: if a DP algorithm $\mathcal{A}$ is the function applied by the data controller when disclosing information, and for any pair of Datasets $D_1, D_2 \in \mathcal{D}$ – where D is the original dataset, and D_1, D_2 differ by at most a single user – the output of

the algorithm A should satisfy

$$\Pr[\mathcal{A}(D_1) \in O] \leq e^{\varepsilon} \Pr[\mathcal{A}(D_2) \in O], \ \forall O \subseteq \text{Range}(\mathcal{A}). \tag{5.1}$$

The parameter ε is called the *privacy budget* and regulates the amount of noise been applied to the output of the query.

Hospital Records (June 2021)

Name	**Vaccine**	**Steps**
Andy	No	8421
Lea	Yes	4256
Eva	Yes	4311
Tom	No	8848

Query: How many people are unvaccinated?
Output: 2
Output DP: 2 ± noise ≈ **1.5**

Hospital Records (July 2021)

Name	**Vaccine**	**Steps**
Andy	No	8421
Lea	Yes	4256
Eva	Yes	4311

Attacker knows Tom is removed
(he withdrew consent/recovered)

Query: How many people are unvaccinated?
Output: 1
Output DP: 1 ± noise ≈ **1.5**

Figure 5.3: The dataset is held by a trusted entity who reports statistical queries about the data. Assuming the adversary knows that an individual is no longer present in the dataset, if the data reported as is, the attacker can infer their vaccination status. When DP is applied, noisy outputs become indistinguishable, preserving the privacy of the participants.

In Figure 5.3 we show how adding noise to the output of the queries may hide the absence of a user in the dataset. DP is heavily applied in practice, e.g., the US Census Bureau adopted differential privacy for the latest 2020 Census [1].

5.2 Privacy-preserving Wearable Data Publishing

Since activity trackers are constantly worn and monitor various parameters of users, they collect vast amounts of fitness data. In the previous chapter, we demonstrated that the

simple metrics of "just" daily steps, calories, and sleep duration captured by wearables could be used by a smart adversary to compromise the privacy of users. At present, data collected by wearable devices tend to become available to the general public. There is, thus, a need to ensure that if such data are released, the potential privacy exposure is negligible.

5.2.1 Types of Wearable Data Release

Dataset release. The uptake of wearables has proven them to be valuable tools in medical studies [56, 114], and activity research [44, 60, 108, 145]. In the course of such experiments researches may fully or partially release the collected data to the community. Typically, fitness wearable information is comprised into the so-called *lifelogging datasets*, where participants wear fitness trackers for a fixed number of days. The term *lifelog* can be interpreted as a recording of one's live. While lifelogs originated in the form of diaries [148], for the past few decades, wearable devices in general and fitness trackers in particular have been prevalent. Typically, lifelogging studies aim to collect as many various types of data as possible. In addition to regular wearable data, researchers may collect other information, including mood and stress questionnaires, injuries reports, food intake, and many more [145]. It is worth mentioning that wearables are also starting to be used in various randomized trials with activity intervention, where researchers collect metrics on 2 distinct user groups [15, 74, 78].

Social network release. However, disclosing wearable fitness data in "bulks" (i.e., datasets of multiple users) is not the only way to share them. In fact, such activity information may be directly provided by the users who produced it. For example, individual data samples can be shared with the world in many FTSN, such as Strava, Fitbit, etc. Given the unprecedented increase in usage of these fitness applications during the COVID-19 pandemic [139], many more users started sharing their wearable data online through FTSNs. These platforms are mainly used by activity enthusiasts to receive feedback on their progress and encouragement from like-minded peers. Another popular platform for individuals to share their activity data is through thematic *fitness communities*, where a strong atmosphere of comradery and friendship is preserved. For example, Fitbit maintains a wide range of communities for various demographic groups, including senior users, pregnant women, and healthy diet followers, etc. Naturally, since thousands of daily snippets for fitness data are posted there, it may be a perfect place for adversaries to harvest data and target users. Indeed, a number of previous works have identified severe privacy leaks in "conventional" social networks [20, 49, 50, 59, 83, 144, 162, 163]. These papers have utilized various features, including publicly available data, group membership, and "likes" to identify various undisclosed attributes. Furthermore, more recent works have managed to compromise privacy of specifically FTSNs. In particular, Dhondt et al. and Hassan et al. managed to breach the route anonymization algorithm of the Strava FTSN [33, 58]. They

showed that an adversary may learn the anonymized endpoints of an activity, which are likely to be either a user's home address or workplace. There is, thus, a need to protect activity information shared online.

Release via crowdsourcing. Finally, wearable data may be disclosed through crowdsourcing platforms [48, 98]. The main difference between the crowdsourcing and dataset disclosure is that the former allows for the recruitment of more participants for the study and may implement additional privacy measures when collecting users data, such as *randomized response* [157].

5.2.2 Wearable Data Release. Common Misconceptions

Wearable fitness trackers collect a wide variety of information, and it may be tempting to disclose all of it, envisioning that someone might find a different use for the data. However, it should be noted that what has not yet been shared can always be published in the future. What has already been released, on the other hand, is public forever.

On-the-surface inference. Naturally, some insights may be directly observed just from the routine of users. Depending on the aggregation of the information, the attacker might acquire various sensitive information about the participants.

- *Daily records.* When the dataset contains entries that are collected every 24 hours (e.g., as in [108]), the adversary can distinguish between target's active and sedentary days. Such insights may be utilized to track the high-level activity and routine changes.

- *Hourly records.* Observing the data samples that are an aggregation of hourly activity (as in e.g., [44]) may help the adversary to distinguish periods of activity and inaction during a day. These findings can be exploited to reveal a wide variety of sensitive habits for the targeted users.

- *Minute-by-minute records.* Typically, the data that can be gathered with wearables like Fitbit and Apple Watch are recorded every minute. It is important to note that the vast majority of the available public wearable datasets [44, 145] simply release all the data that have been collected by the devices, which are "minute-by-minute." Since such format allows the adversary to infer active periods with even higher precision, they are likely to learn even **more** sensitive information in typical cases. Figure 5.4 highlights the differences in activity levels during morning hours between hourly and minute-by-minute samples of user's data.

At first glance, it does not appear difficult to mitigate the above-mentioned threats. However, in practice, the most obvious approaches might not be very effective at protecting

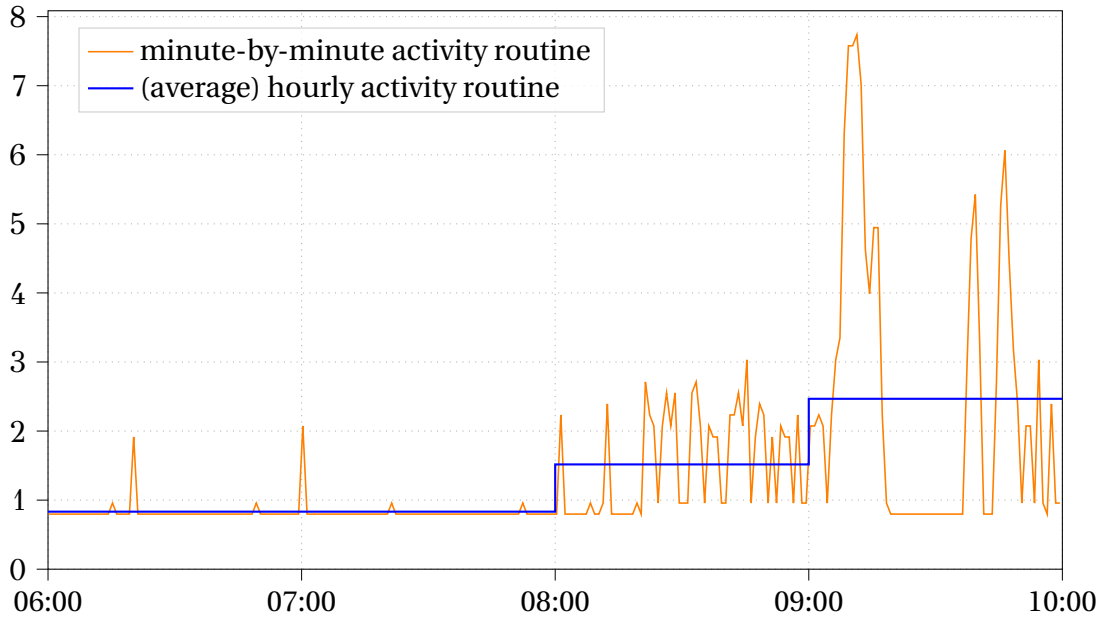

Figure 5.4: Morning routine of a participant in the CSFD dataset [44] based on burned calories. The calories are collected hourly (orange line) or every minute (blue line). For the hourly routine the average number of calories per minute over that hour is presented. While the wake-up time is detected around 8 a.m. for both cases, the higher granularity of the data provides deeper insights into the morning routine of the user.

privacy of the dataset's participants. In order to effectively address the privacy risks, the following common misconceptions must be dispelled. We present them from the standpoint of an *inexperienced data controller* who wants to make their research data public.

Fallacy I: *I will remove direct identifiers of the participants, such as name, phone number, email, etc., and release it. Surely, it is enough to protect their privacy.*
Although pseydonimization is a necessary first step to protect any sensitive data, an adversary who possesses any auxiliary information on a known participant can easily link it to their activity records. Indeed, an adversary may de-anonymize a significant number of users based on indirect physical identifiers.

Fallacy II: *Well, then I will also discard all physical and demographic attributes (age, weight, height, etc.), in case the attacker knows some users. Now, the protection level has to be sufficient.*
Some of the data produced by the fitness trackers depend on physical parameters of the wearer. For example, the number of burned calories heavily correlates with weight, gender,

height, etc. (Chapter 4). Therefore, it may be feasible to reconstruct those characteristics directly from fitness data.

Fallacy III: *OK, in that event, I will remove **all** quasi-identifiers and release **only** activity data and no other information. At last, my dataset is fully protected.*
Although removing quasi-identifiers definitely increases the overall privacy level of the data, some insights still may be inferrable. Since fitness data themselves carry a wealth of information about the person who produced them, it might be possible to fingerprint users based solely on their activity information. In fact, as shown in Chapter 4, individuals may be re-identified based on solely their activity records an no other information.

5.2.3 Wearable Data Release. Guidelines

In this section, we provide practical guidelines to protect wearable data when releasing them via fitness datasets. However, before diving deep into the specific procedures, a general principle should be outlined: every dataset, when published, should contain only the *required* amount of information for the task. This principle is known as "Data minimization" – a cornerstone of data science. This approach, applied to the information release, implies not oversharing more than is needed. For example, for an experiment that correlates watching horror movies and stress levels, users' height might not be of the utmost importance. It is worth noting that entities who collect personal data of users should adhere to the data minimization principle, according to both GDPR and CCPA. Therefore, all the irrelevant information should be discarded from a dataset before any kind of anonymization is applied.

Microdata release. Microdata are the information that is collected once per user, usually at the start of the study. Such data are typically represented by indirect quasi-identifiers or sensitive attributes.

- *Quasi-identifiers* should be anonymized via generalization (k-anonymity) approaches as indicated in Section 5.1. Typically, quasi-identifiers need to be generalized/suppressed until they are shared by at least k users.
- *Sensitive attributes* should be secured by ensuring that both l-diversity and t-closeness are satisfied (Section 5.1).

Note that data controllers need to put special effort into anonymizing individuals who belong to minority groups, as they are the most likely to be re-identified.

Time-series data. "Time-series data" are the wearable information that is sampled with a certain frequency, and that are associated with a particular timestamp. For consumer

wearable trackers such data include steps, calories, distance, workout, sleep, etc. We suggest the following techniques to reduce the possibility for inference of sensitive insights:

- *Data aggregation.* For many fitness studies, it may be feasible to report generalized values instead of the detailed ones, e.g., steps: $12345 \rightarrow 10000+$; BMI: $29.87 \rightarrow$ overweight; height: $193 \rightarrow$ tall. That also applies to e.g., releasing cumulative steps per day, instead of disclosing their number every minute.

- *Data sanitization.* Data controllers may attempt to modify the existing data samples to retain the useful utility information and lose user-specific traits. Unlike DP, such methods apply changes (noise) directly to the data and not the output of the queries. Since this method introduces changes to the original data, it may not always be applicable. Sanitization of sensor data has been introduced in previous works [18, 92, 93]. And while they reported promising results for both preventing re-identification of users and preserving utility of the accelerometer/gyroscope data, it remains unclear whether the proposed solutions generalize to other datasets.

- *Data generation.* It may be achievable to disclose synthetic data samples that appear to be "similar" to the original ones. Such techniques may be used if generated data are indistinguishable for the purpose of the data release. For example, generation of Fitbit fitness samples, using ML, have been investigated in [66]. The authors utilized a Generative Adversarial Network (GAN) to create fitness samples based on a real-world Fitbit dataset. The work, however, does not present any evidence that the produced data are resilient to de-anonymization attacks or provide any theoretical/practical privacy guarantees.

- *Regulating the dataset population.* One of the most simple and yet effective ways to reduce the success rate of de-anonymization attacks is to expand the total number of participants in the dataset. Increasing the size of the dataset reduces the probability of naive guessing.

The optimal strategy for disclosing activity information likely involves complementing the traditional anonymization approaches with wearable-specific techniques proposed in this section.

5.3 Lifesnaps

To show that the proposed anonymization techniques work in practice, the RAIS consortium has released its own lifelogging dataset, called Lifesnaps, containing wide-variety of fitness data collected by wearable devices. Since in the course of this dissertation we do not focus on the "quality" of the lifelog data, we only address the privacy aspects of our dataset. Other

details of Lifesnaps are extensively discussed in Publication VI. In this dissertation, we briefly discuss the distribution of demographic parameters, since outliers may be uniquely de-anonymized under some conditions as indicated in Chapter 4. We conduct 2 rounds of the experiment, recruiting more than 70 diverse users located in various parts of Europe. All the users voluntarily contributed to the project and have the ability to withdraw their consent for data processing at any time. A demographic comparison between Lifesnaps and other publicly available wearable datasets is depicted in Table 5.1. Lifesnaps has more unique users than all of the other datasets combined. While the only other dataset to contain all physical attributes is PMData, its distribution does not seem to reflect real-world trends in gender statistics.

Dataset	Openhumans [108]	CSFD [44]	PMData [145]	Lifesnaps [167]
Users	31	13	16	71
Males	18	-	13	42
Females	13 (42%)	-	3 (19%)	29 (41%)
Overweight	15	9	7	15
Not Overweight	16 (52%)	4 (31%)	8 (53%)	54 (78%)
Age $\leq$ 29	-	-	8	35
Age > 29	-	-	8	34

Table 5.1: Distribution of demographics and physical parameters of the employed datasets and that of Lifesnaps. Overall, Lifesnaps appears to be more balanced and well-represented.

Our commitment to the participants was to protect their privacy and sensitive information, so we carefully anonymize the dataset before publishing it. In the process, we adhere to the following principles: (i) minimizing the probability for successful re-identification of users by real-world adversaries, (ii) maximizing the amount of retained data that are of *use* to the researchers and practitioners, (iii) abiding by the principles and recommendations of GDPR in regards to the handling of personal information, and (iv) following the established anonymization practices and principles. In doing so, we followed several guiding principles:

- We try to minimize the probability of participant re-identification while preserving as much relevant data as possible. We protect our dataset against various real-world attack vectors and adversaries. We specifically focus on retaining information that may be used by researchers and medical practitioners.
- We comply with GDPR principles and recommendations, regarding the management of personal information. In particular, before anonymizing the data, we store them in secure university servers and proprietary cloud services. We delete original data

of the participants upon withdrawal of their consent. The consent is valid for two years unless it is revoked by users. Since anonymized data are not subject to GDPR regulations, they can be stored for an indefinite period of time.

- We follow the established anonymization practices and principles, which we describe in more detail below.

To start the anonymization process, we pseudonymize the users and remove all direct identifiers. We then remove some of the quasi-identifiers that are overly sensitive and of limited interest to the research community, such as ethnicity, country, and timezone. Next, we aggregate the physical parameters of the users to achieve at least 2-anonymity under the strongest attack model. However, the aggregation ranges were not only selected to preserve dataset anonymity but also to align with meaningful real-world categories.

In particular, we categorized the participants into two groups based on their age: individuals under 30, referred to as young adults, and those above 30. Instead of revealing the exact height and weight of the participants, we disclosed their BMI rounded off to the nearest integer. In addition, we replaced the extreme BMI values with ranges, such as underweight or overweight, to protect the privacy of the outlier users. Only the gender of the participants was the quasi-identifier released without any modification. Using the quasi-identifiers (gender, age, and BMI) present in the released version of the data, we demonstrate that our dataset achieves 2 to 12-anonymity, depending on the strength of the adversary we consider. The summary of the defense mechanisms applied is presented in Table 5.2.

Technique	Previous datasets	Lifesnaps
Pseudomization	+	+
k-anonymity	-	+
Data minimization	+	+
Data aggregation	-	+
Data sanitization	-	-
Data generation	-	-
Dataset size	16-33	71

Table 5.2: Anonymization techniques utilized in Lifesnaps. We do not apply data sanitization (as defined earlier) or generation to increase the penetration of our dataset. We utilize the rest of our recommendations put forth in Publication V.

Threat model. In order to ensure that the anonymity standards are upheld, it is necessary to confirm that a potential *realistic* adversary cannot identify any of the participants. For

Intelligence

User	Gender	Age	Distinct parameters
John Jones	m	44	-
Lea Smith	f	25	overweight
Joe Brown	m	29	extremely tall
Anna Green	f	34	-
...	...	...	...
Kim Davis	f	19	disabled

Adversary

Joe Brown = ?

Type	User 1	User 2	...	User N
Profile	data	data	...	data
Steps	data	data	...	data
Calories	data	data	...	data
Sleep	data	data	...	data
IPIP	data	data	...	data
TTEM	data	data	...	data
...	...	...	...	...
BREQ	data	data	...	data

LifeSnaps

Figure 5.5: In the threat model we consider, the attacker has acquired a list of all the individuals in the dataset, along with their ages and physical descriptions. The adversary aims to link the participants (or even a single user) back to their data. Since we do not disclose the participants' height and weight, their physical descriptions have significantly less utility in de-anonymizing them. Lifesnaps achieves 12-anonymity under the relaxed threat model and at least 2-anonymity under the strongest (most favorable to the attacker) one. For more details on *types* of data (e.g., BREQ) please refer to [167].

example, if the attacker possesses information that Anna Green, a participant in the study, recorded precisely 23, 451 steps on a specific day, it would be a straightforward task to re-identify her. In such scenarios, modifying quasi-identifiers cannot maintain anonymity. The only available option is to introduce random perturbations to the time-series data, which would negatively impact their utility.

Instead, we adopt a realistic yet strong threat model. Suppose that the attacker has acquired a list of all individuals in the dataset and has found their birthdates through publicly available sources, as shown in Figure 5.5. Furthermore, the adversary has surveilled

all of the participants from the list either online or in public spaces, and has learned their physical parameters and some distinctive traits, such as being very tall or overweight. The privacy of Lifesnaps could be compromised if the attacker is able to de-anonymize even *a single* user. Thus, we ensure the anonymity of each individual to safeguard the entire dataset. If the adversary only has access to the essential quasi-identifiers (age and gender), our dataset satisfies unconditional 12-anonymity. Moreover, since we do not release precise information on the height and weight of the participants, the attacker cannot directly use the physical parameters observations (column "Distinct parameters" in Figure 5.5) to de-anonymize the users, rendering such insights useless.

Indeed, we show that the attacker cannot utilize the auxiliary information on physical appearances of the individuals to link them with the public BMI values. Given that BMI is dependent on both height and weight, the adversary must estimate both attributes to compute BMI (as per Equation 5.2). Assuming the adversary can make educated guesses for these parameters, with an error range of $\pm 5kg$ for weight and $\pm 5cm$ for height, the resulting error range for BMI can be determined using error propagation (as per Equation 5.3).

$$\text{BMI} = \frac{\text{weight}}{\text{height}^2} \tag{5.2}$$

$$\epsilon_{\text{BMI}} = \frac{\Delta\text{weight}}{\text{weight}} + \frac{2 * \Delta\text{height}}{\text{height}} \tag{5.3}$$

To illustrate, consider a person with a height of 170 cm and a weight of 70 kg, whose $\lfloor \text{BMI} \rfloor$ value is 24. Due to the error interval of ±5 cm for height and ±5 kg for weight, the range of possible values for BMI is relatively large, spanning almost from underweight to obese, and includes 8 integer values in the interval $I = \{21, 22, 23, \mathbf{24}, 25, 26, 27, 28\}$. Thus, it is evident that the adversary cannot gain significant insights on the users due to error propagation. Even if the attacker could accurately estimate the height and weight of the participants, which is highly unlikely, the best they could do is to reduce the anonymity factor k to a minimum of 2.

5.4 What Can Regular Users Do?

We urge users who disclose their wearable data in any form to be mindful of their privacy. Summarizing the previous paragraphs, we outline simple to implement but yet effective techniques to limit sensitive exposure when sharing wearable data.

Sharing wearable data online. Perhaps the most effective way to disclose personal wearable data online is to (i) post it under a pseudonym (Section 5.1). Indeed, an adversary may have very limited use of such de-identified data as auxiliary information. Nevertheless, if pseudonymization is not an option, there are other courses to bolster one's privacy.

Individuals who post their fitness data online on FTSNs may also consider (ii) reducing the granularity and modality of shared data by posting fewer fitness parameters that are aggregated over periods of time. For example, when an individual wants to share a particular training session, they might publish only their daily steps, instead of posting hourly data on steps, calories, and heart rate. Indeed, the fitness community will be made aware of the person's activity levels in both cases without compromising the user's privacy. A number of FTSNs (e.g., Fitbit) allows users to customize the fitness values they want to share with the community in great detail. Another strategy for sharing information is to (iii) reduce the frequency and number of online posts. As demonstrated in Chapter 4, the sensitive inference capabilities increase with more data samples per target. Finally, regular users may consider sharing only their "anomalous" data (iv), i.e., posting non-typical activity records rather than their daily routine. Such behavior is consistent with the spirit of FTSNs, which encourage users to share their greatest achievements or setbacks. Additionally, this approach heavily skews the distribution of a user's wearable records, making it more difficult for adversaries to utilize such data to the same extent as normal wearable samples.

Sharing wearable data to datasets. Participating in various fitness studies that involve consumer wearables may be associated with significant privacy concerns. As regular users only wear the devices and submit the generated data, they have little control over them. All processing and release of data are in the hands of data controllers. Nevertheless, users may choose not to participate in studies that do not guarantee a sufficient level of privacy. We outline a set of inquires users should make before agreeing to join a fitness study:

- *Public/private dataset.* The most important question is whether the collected data will be made public at the end of the study. Naturally, if only the aggregated results are planned to be published, it dramatically reduces the chances of exposing private information (although it does not prevent it completely).

- *Shared data.* In cases where some portions of fitness data will be made available, it is sensible to learn which activity data are planned to be included in the release version. For instance, for a stress study, it may be sensible to opt-out if, besides the actual stress, all other activity information collected by a wearable is released (full routine, demographics, injuries, etc.).

- *Anonymization efforts.* Another vital piece of information concerns the anonymization techniques used by the data controller. As we report previously, simply removing unique identifiers of users is not enough to claim *anonymity* and may be a reason not to join the study. However, if the data controller is committed to enhancing privacy by utilizing state-of-the-art techniques such as k-anonymity, differential privacy, etc., users may proceed.

- *Dataset population.* A reasonable question to ask the experiment organizers is how

many individuals are/will be participating. As shown in our experiments, the re-identification accuracy tend to sharply drop with the increase of the dataset population. It is evident that an individual is much likely to be identified in a dataset of 20 compared to that of $1,000$.

- *Data erasure.* Finally, users need to know what happens to their data if they withdraw from the study. Naturally, if their data are not deleted (which goes against GDPR and CCPA), it may be sensible not to participate.

Sharing wearable data via crowdsourcing. Although sharing wearable data with crowdsourcing platforms is largely outside the scope of this dissertation, we still provide some insights on how to limit the exposure of sensitive information. In general, all the guidelines for privacy-preserving release of datasets apply. In addition users may choose the crowdsourcing platform that supports Local Differential Privacy (LDP) and randomized response when collecting data from users [98]. If this is the case, users do not need to fully trust the platform, since the combination of LDP and randomized responses ensures that the data controller cannot fully recover fitness data beyond a certain threshold. If such data are later released to the public, LDP provides mathematical guarantees against sensitive inference.

Chapter 6
Conclusion

We conclude the dissertation by summarizing our contributions to the research questions presented in the introduction, which include: (i) security and (ii) privacy of consumer wearables, (iii) attacks on wearable data, and (iv) privacy-preserving release of such data. Moreover, we outline the directions for future work in relation to the published studies.

Overall, this thesis outlines the potential risks that regular users of consumer wearables may face when using their devices out-of-the-box. We emphasize the importance of this problem by demonstrating the significant privacy leaks that may occur with these devices and the data they generate. Furthermore, we present *practical* solutions and techniques that can be easily employed by an average owner of a fitness tracker. Finally, we caution *all* tech consumers to remain vigilant and always research the privacy aspects of any device or service associated with wearable technology.

6.1 Synopsis of Contributions

In this thesis, we explored the privacy and security issues associated with consumer wearable devices. We studied not only physical devices themselves but also the various services associated with them and the data they collect. We consider these problems from the standpoint of an *average* fitness tracker user who uses the devices in the "out of the box" mode, following the official instructions of the manufacturer and installing official companion and partner applications. Furthermore, many typical consumers participate in fitness social communities and share the data collected by their devices, which can lead to significant information leaks due to the inherent security and privacy concerns associated with the "default" use of wearables.

Our research shows that users may unwittingly provide significant sensitive insights to third parties who are often unknown and undesired by using companion and partner apps provided by popular smartband vendors. Installing these apps is often mandatory in order to access the full functionality of the devices. Moreover, we have identified that several well-known wearables are vulnerable to the traffic analysis attacks, allowing the passive

adversary to extract fitness information and other private data on regular consumers. We present and evaluate defense techniques against these types of leaks and demonstrate that it is feasible to avoid privacy exposure with minimal effort. The suggested approaches do not require extensive skills and may be utilized by people with limited technical expertise.

Furthermore, in our works, we extensively analyze the data collected by wearables that are often exposed to the general public. We demonstrate that uncontrollable and unsanitized release of such data may greatly compromise the privacy of the users. We introduce several novel attacks against fitness data and have proposed ways to mitigate them. Our attacks show that users can be de-anonymized solely based on their activity records without requiring any non-wearable auxiliary data. We also demonstrate that sensitive physical attributes, such as gender, BMI, and height, can be inferred from even a minimal amount of fitness data. Therefore, we emphasize that participating in fitness studies, even with a large number of participants, can result in significant privacy breaches for wearable device users.

We present and dispel common misconceptions about privacy-preserving information sharing and suggested a complete methodology of fitness data release. As part of the RAIS project, we have published the Lifesnaps dataset, which includes a lifelogging study of $N = 71$ users who wore wearable devices. Lifesnaps contains highly granular wearable data, validated surveys, and ecological momentary assessments, making it significantly more valuable to the research community than other open datasets. In line with principles proposed in our research, we thoroughly anonymized the dataset and made it publicly available.

6.1.1 Security of Wearables (RQ1)

Although attacks on fitness trackers are extensively discussed in the literature, the vast majority of them are not applicable to average users of such devices. They are typically based on unrealistic threat models, short-lived, and involve only specific models of fitness trackers. Instead, in Publication I, we propose, perform, and evaluate a novel attack against wearable devices that may severely compromise the privacy of regular consumers.

The attack is based on the analysis of encrypted traffic to find patterns for specific activities and measurements performed by users. Our attack is (i) passive, making it almost impossible to detect, (ii) device model-agnostic, and (iii) can be executed remotely. Furthermore, unlike other attacks that require distinct modifications to the device or the phone (e.g., root), our threat model allows adversaries to specifically target out-of-the-box users who use their smartbands “as is” [70]. The attacker takes the form of an ISP located somewhere along the path between the phone running a companion application and the cloud. This ISP is honest but curious, meaning it accepts and delivers IP packets to the appropriate destination (the cloud) but attempts to infer information from encrypted

traffic. The adversary launches the attack when the companion application attempts to synchronize activity data with the manufacturer's servers.

We show that such an ISP is able to glean significant insights on the regular users of wearables, including (i) the frequency of heart rate and weight measurements, (ii) the length and duration of workouts, (iii) periods of absence for sleep and low activity levels, and (iv) extreme values of BPM and weight. In Publication I, we empirically demonstrate that flagship devices (as of 2020) from at least two well-known manufacturers of consumer fitness trackers, Xiaomi and Samsung, are susceptible to the proposed attack.

6.1.2 Privacy of Wearables (RQ2)

The ubiquitous data collection of consumer wearable devices has long raised concerns about the privacy of users. While users expect to share their data with the official device vendors, they are likely to oppose relinquishing any sensitive information to undisclosed third parties. To our surprise, despite the continuously rising number of device manufacturers and models, no comprehensive analysis of third-party connections for wearables had been done previously. In our works, we investigate which third parties are being contacted by well-known brands of wearables via their companion applications as well as the most popular partner apps. We demonstrate that such third-party connections may not be anticipated by regular users of fitness trackers. These connections include advertisement services, analytics providers, various external APIs, and even social networks, as we discuss in Publications II-III.

We also analyze the data that are being sent to such unexpected entities. Our findings suggest that private information may indeed be disclosed when users simply use their wearable devices [69]. In some cases, this information may be extremely sensitive, including precise location, demographic information (such as age and gender), lifestyle factors (such as sedentary behavior), email addresses, and Android Advertising IDs (AAIDs). The "less sensitive" shared data contain phone model, SIM carrier, and connection history. We conclude that unanticipated entities can glean sensitive information on regular users who use their wearable devices normally, as discussed in Publication II.

To combat the above privacy threats we investigate whether such unwanted connections may be reduced or altogether prevented. Inspired by other works in the field of IoT we examine whether it is feasible to disable the undesired third-party connections of wearable devices without hindering their essential functionality. In our research, we demonstrate that disabling connections to the domains that are present in well-known blocklists does not hinder the correct synchronization of fitness data by Fitbit devices [71]. We perform extensive empirical evaluation and verify that the most important activity metrics for regular users of wearables [24] remain intact. In particular, regular users of Fitbit trackers can accurately track their step count, covered distance, workouts, duration and quality of

sleep when blocking unwanted third-parties. Our analysis suggests that at least 88% of the connections for the official Fitbit application and 6 partner apps are contained in credible blocklists and can be safely disabled. We further discover that *all* studied apps contact undesired third parties. To combat this, we suggest a blocking methodology that can be employed by regular users of wearables. Our proposed approach involves using mobile filtering services, also known as adblockers, and identifying the most effective blocking lists for wearable applications (Publication III). This methodology can be easily utilized by average fitness tracker users with limited technical competency.

6.1.3 Attacks on Wearable Data (RQ3)

Regular users tend to not only use their wearables but also utilize related accessory services. Fitness trackers, in particular, have become essential tools in various medical studies [15, 23, 56, 74, 78, 114]. Furthermore, users of wearables may join the specialized fitness social networks within the corresponding companion apps. Consequently, wearable data may be shared with the research community and the general public. In this thesis, we investigated whether existing fitness datasets and information sharing techniques are adequate or can be compromised. In particular, readily available fitness activity datasets employ conventional approaches to protect the privacy of the participants by removing personal identifiers, including the full name, e-mail address, and sensitive attributes. Unfortunately, applying only such direct methods, although seemingly appropriate, may not be enough to protect the owners of wearable data. We proposed several novel attacks against data collected by wearable fitness trackers (Publication IV). These attacks enable adversaries to de-anonymize users of fitness datasets and infer their demographic parameters "solely" from activity data and no other information. We show that a user's gender, BMI, and height can be inferred from Fitbit samples by using a minimal number of features, namely activity information [72]. Furthermore, by utilizing daily snippets of steps, distance, calories, and average heart rate, the attacker is able to re-identify users in public wearable datasets with a 93.5% probability. We demonstrate that the de-anonymization rate reaches 100% for individuals with distinct physical attributes, such as very tall or obese people.

6.1.4 Privacy-preserving Release of Wearable Data (RQ4)

Since attacks that specifically target wearable IoT data exist (Publication IV), we set out to investigate ways to minimize the probability of privacy leaks. Furthermore, since conventional anonymization techniques may not be enough to preserve the privacy of wearable data (Publication IV), we research other common misconceptions about information sharing (Publication V). We explain why removing participants' personal identifiers and sanitizing their physical attributes may not be enough to fully protect their anonymity. We present common approaches for privacy-preserving wearable data disclosure (Publication

V). These methods include: data (i) minimization, (ii) aggregation, (iii) sanitization, and (iv) generation [97]. Combined with the conventional approaches, the proposed techniques limit the possibility of inference from wearable data. We further emphasize the importance of resampling/subsampling, as well as reducing granularity when publishing fitness samples.

Finally, we demonstrate the practical privacy-preserving wearable data release by publishing our own lifelogging dataset, Lifesnaps [168] (Publication VI). The data were collected over a four-month period in an unobtrusive study, with $N = 71$ unique users wearing Fitbit fitness trackers. Lifesnaps is geographically distributed across Europe, providing valuable insights into activity dynamics in various countries during the COVID-19 pandemic. Compared to its publicly available counterparts, Lifesnaps contains more activity types and higher granularity, making it superior in terms of data volume. We used previously proposed anonymization techniques to protect the privacy and anonymity of all participants. We show that Lifesnaps is more resilient to both conventional and fitness-specific attacks compared to other publicly available lifelogging datasets.

6.2 Directions for Future Work and Research

This thesis has investigated a wide range of privacy aspects associated with consumer wearable devices, with a specific focus on assessing the impact of wearable technology on the privacy of regular users. However, several elements could be further explored. For example, the traffic analysis attack mentioned in Section 2.2 was evaluated only in laboratory settings, without assessing it in the wild. Prior research has indicated that such attacks could be executed on a much larger scale by utilizing data collected by real-world ISPs [121, 122, 165]. In these works the authors obtained access to the actual traffic of operating ISPs and were able to run statistical queries to identify and analyze the encrypted packets of interest. Conducting our attack from the standpoint of an actual "honest but curious" ISP would further facilitate our findings and raise awareness for out-of-the-box device usage (Publication I). However, conducting experiments of this nature requires ongoing collaborations with actual Internet providers, which may prove extremely challenging due to privacy protection laws, internal policies, and ethical considerations. Therefore, a more adequate continuation of the work done in Publication I would involve collecting greater volumes of data in laboratory settings and attempting to distinguish the activity packets from other traffic generated by IoT devices.

To mitigate the traffic analysis of encrypted wearable data we propose several well-established techniques in Publication I. Their implementation could be a promising direction for future research. Previous research has shown that traffic padding techniques can be effective in protecting against inference in web-based applications [11, 84], and we believe such approaches would be appropriate for wearable apps. However, the main challenge

does not lie in implementing the defense methodology itself but in ensuring that it can be used by regular users of wearables.

Another research area that is worth exploring concerns preventing third-party connections of wearable applications (Publications II, III). Although we have shown that regular users can effectively use ad-blockers to prevent privacy leaks (Publication III), it remains an open question whether this approach puts additional strain on mobile phones and can be utilized as a long-term solution. Several articles have investigated the traffic and energy consumption of mobile applications in general [79, 107], as well as the built-in advertisements in particular [53, 54, 110]. The above works have detected a significant increase of power consumption for the applications that display advertisement to their users. Moreover, Papadopoulos et al. established up to an 8% increase in Internet traffic of mobile applications when displaying advertisement [110]. Potential future research includes assessing the CPU utilization and the energy consumption of the proposed blocking approach compared to running the default version of the application. It should be noted that the regular versions of wearable applications contact not only advertisement providers but also all unwanted third parties (Publications II, III). Furthermore, our aim is to estimate the reduction in traffic volume of wearable applications when they do not send any requests to undesired third parties. Additionally, considering a broader set of partner applications and prominent device manufacturers, especially those that have grown in popularity since 2020, provide an encouraging direction for future research. Finally, we believe that creating and maintaining our personal blocklist of unnecessary wearable domains could be of great value for the research community and fitness tracker userbase. This blocklist would specifically contain URLs of unwanted third parties that are being contacted by the most popular companion and partner applications of well-known vendors. We plan to actively maintain our wearable blocking list based on changes in third-party connections and the release of new models/vendors.

With regard to the attacks against wearable data (Publication IV), exploring other possible attack vectors seems to be a sensible continuation of our research. This includes not only utilizing other fitness parameters as features, but also inferring more robust insights about the users. One of the key limitations of our studies on de-anonymization is the small number of users present in the fitness dataset. In fact, previous studies employing fitness trackers have comprised populations of thousands [103], tens of thousands [116], hundreds of thousands [115, 117], and even millions of individuals [172]. Naturally, none of the aforementioned studies made their datasets publicly available. Furthermore, obtaining such large data collections usually requires ongoing partnerships with wearable vendors, which is not trivial to arrange. We plan to extend our research to explore privacy-preserving ways of releasing wearable data within *fitness communities*. As our previous study has shown that several undisclosed insights on users can be inferred directly from their activity snippets (Publication IV), it is crucial to identify appropriate online sharing strategies to

limit privacy exposure.

The data protection techniques proposed in Publication V include data sanitization, which constitutes a suitable course for future research. Several previous articles have proposed techniques for sanitizing wearable samples by employing data-driven machine learning solutions [18, 66, 92, 93]. However, it remains unclear whether these approaches generalize to other datasets and samples due to the insufficient volume of utilized training data. Therefore, we plan to investigate other sanitization techniques, including non-data-driven LDP [155, 166, 171]. LDP solutions have already been utilized to protect the privacy of general IoT data in previous studies [4, 10, 75, 123]. Our aim is to identify the appropriate ways of applying LDP to protect the data generated by consumer wearable trackers.

Bibliography

[1] John M Abowd. The US Census Bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2867–2867, 2018.

[2] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and Selcuk Uluagac. Peek-a-boo: I see your smart home activities, even encrypted! In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 207–218, 2020.

[3] Alessandro Acquisti and Jens Grossklags. Privacy and rationality in individual decision making. *IEEE security & privacy*, 3(1):26–33, 2005.

[4] Sharmin Afrose, Danfeng Daphne Yao, and Olivera Kotevska. Measurement of Local Differential Privacy Techniques for IoT-based Streaming Data. In *2021 18th International Conference on Privacy, Security and Trust (PST)*, pages 1–10. IEEE, 2021.

[5] Ahmet Aksoy and Mehmet Hadi Gunes. Automated iot device identification using network traffic. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2019.

[6] Arash Alavi, Gireesh K Bogu, Meng Wang, Ekanath Srihari Rangan, Andrew W Brooks, Qiwen Wang, Emily Higgs, Alessandra Celli, Tejaswini Mishra, Ahmed A Metwally, et al. Real-time alerting system for COVID-19 and other stress events using wearable data. *Nature medicine*, 28(1):175–184, 2022.

[7] Abdulmajeed Alqhatani and Heather Richter Lipford. " There is nothing that I need to keep secret": Sharing Practices and Concerns of Wearable Fitness Data. In *SOUPS@ USENIX Security Symposium*, 2019.

[8] Ahmed Alshehri, Jacob Granley, and Chuan Yue. Attacking and protecting tunneled traffic of smart home devices. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, pages 259–270, 2020.

[9] Eduardo B Andrade, Velitchka Kaltcheva, and Barton Weitz. Self-disclosure on the web: The impact of privacy policy, reward, and company reputation. *ACR North American Advances*, 2002.

[10] Arno Appenzeller, Nick Terzer, Erik Krempel, and Jürgen Beyerer. Towards Private Medical Data Donations by Using Privacy Preserving Technologies. In *Proceedings of the 15th International Conference on PErvasive Technologies Related to Assistive Environments*, pages 446–454, 2022.

[11] Noah Apthorpe, Danny Yuxing Huang, Dillon Reisman, Arvind Narayanan, and Nick Feamster. Keeping the smart home private with smart (er) iot traffic shaping. *arXiv preprint arXiv:1812.00955*, 2018.

[12] Noah Apthorpe, Dillon Reisman, and Nick Feamster. A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic. *arXiv preprint arXiv:1705.06805*, 2017.

[13] Noah Apthorpe, Dillon Reisman, Srikanth Sundaresan, Arvind Narayanan, and Nick Feamster. Spying on the smart home: Privacy attacks and defenses on encrypted iot traffic. *arXiv preprint arXiv:1708.05044*, 2017.

[14] Orlando Arias, Jacob Wurm, Khoa Hoang, and Yier Jin. Privacy and security in internet of things and wearable devices. *IEEE Transactions on Multi-Scale Computing Systems*, 1(2):99–109, 2015.

[15] Yang Bai, Ryan Burns, Nancy Gell, and Wonwoo Byun. A randomized trial to promote physical activity in adult pre-hypertensive and hypertensive patients. *Journal of Sports Sciences*, 40(14):1648–1657, 2022.

[16] Rebecca Balebako, Richard Shay, and Lorrie Faith Cranor. Is your inseam a biometric? evaluating the understandability of mobile privacy notice categories. *CMU, Tech. Rep. CMU-CyLab-13-011*, 2013.

[17] George Dean Bissias, Marc Liberatore, David Jensen, and Brian Neil Levine. Privacy vulnerabilities in encrypted HTTP streams. In *International Workshop on Privacy Enhancing Technologies*, pages 1–11. Springer, 2005.

[18] Antoine Boutet, Carole Frindel, Sébastien Gambs, Théo Jourdan, and Rosin Claude Ngueveu. DYSAN: Dynamically sanitizing motion sensor data against sensitive inferences through adversarial networks. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 672–686, 2021.

[19] Pew Research Center. About one-in-five Americans use a smart watch or fitness tracker. `https://www.pewresearch.org/fact-tank/2020/01/09/about-one-in-five-americans-use-a-smart-watch-or-fitness-tracker/`, 2020, January 9. Online; Retrieved May 30, 2022.

[20] Abdelberi Chaabane, Gergely Acs, Mohamed Ali Kaafar, et al. You are what you like! information leakage through users' interests. In *Proceedings of the 19th annual network & distributed system security symposium (NDSS)*. Citeseer, 2012.

[21] Shuo Chen, Rui Wang, XiaoFeng Wang, and Kehuan Zhang. Side-channel leaks in web applications: A reality today, a challenge tomorrow. In *2010 IEEE Symposium on Security and Privacy*, pages 191–206. IEEE, 2010.

[22] Heyning Cheng and Ron Avnur. Traffic analysis of SSL encrypted web browsing. *Project paper, University of Berkeley*, 1998.

[23] Man Lai Cheung, Ka Yin Chau, Michael Huen Sum Lam, Gary Tse, Ka Yan Ho, Stuart W Flint, David R Broom, Ejoe Kar Ho Tso, and Ka Yiu Lee. Examining consumers' adoption of wearable healthcare technology: The role of health attributes. *International journal of environmental research and public health*, 16(13):2257, 2019.

[24] Kimberly PL Chong, Julia Z Guo, Xiaomeng Deng, and Benjamin KP Woo. Consumer perceptions of wearable technology devices: retrospective review and analysis. *JMIR mHealth and uHealth*, 8(4):e17544, 2020.

[25] Michelle M Christovich. Why Should We Care What Fitbit Shares-A Proposed Statutroy Solution to Protect Sensative Personal Fitness Information. *Hastings Comm. & Ent. LJ*, 38:91, 2016.

[26] Jiska Classen, Daniel Wegemer, Paul Patras, Tom Spink, and Matthias Hollick. Anatomy of a vulnerable fitness tracking system: Dissecting the fitbit cloud, app, and firmware. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 2(1):1–24, 2018.

[27] Intersoft Consulting. General Data Protection Regulation GDPR. `https://gdpr-info.eu/`, 2018, May 25. Online; Retrieved May 30, 2022.

[28] Kate Crawford, Jessa Lingel, and Tero Karppi. Our metrics, ourselves: A hundred years of self-tracking from the weight scale to the wrist wearable device. *European Journal of Cultural Studies*, 18(4-5):479–496, 2015.

[29] Brian Cusack, Bryce Antony, Gerard Ward, and Shaunak Mody. Assessment of security vulnerabilities in wearable devices. 2017.

[30] George Danezis. Traffic Analysis of the HTTP Protocol over TLS, 2009.

[31] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3(1):1–5, 2013.

[32] Yves-Alexandre De Montjoye, Laura Radaelli, Vivek Kumar Singh, and Alex "Sandy" Pentland. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science*, 347(6221):536–539, 2015.

[33] Karel Dhondt, Victor Le Pochat, Alexios Voulimeneas, Wouter Joosen, and Stijn Volckaert. A Run a Day Won't Keep the Hacker Away: Inference Attacks on Endpoint Privacy Zones in Fitness Tracking Social Networks. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 801–814, 2022.

[34] Yujie Dong, Adam Hoover, Jenna Scisco, and Eric Muth. A new method for measuring meal intake in humans via automated wrist motion tracking. *Applied psychophysiology and biofeedback*, 37(3):205–215, 2012.

[35] Cynthia Dwork. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pages 1–12. Springer, 2006.

[36] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[37] Hossein Fereidooni, Jiska Classen, Tom Spink, Paul Patras, Markus Miettinen, Ahmad-Reza Sadeghi, Matthias Hollick, and Mauro Conti. Breaking fitness records without moving: Reverse engineering and spoofing fitbit. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 48–69. Springer, 2017.

[38] Hossein Fereidooni, Tommaso Frassetto, Markus Miettinen, Ahmad-Reza Sadeghi, and Mauro Conti. Fitness trackers: fit for health but unfit for security and privacy. In *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pages 19–24. IEEE, 2017.

[39] Kaja Fietkiewicz and Aylin Ilhan. Fitness tracking technologies: Data privacy doesn't matter? The (un) concerns of users, former users, and non-users. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2020.

[40] The Firebog. The Big Blocklist Collection. `https://firebog.net/`. Online; Retrieved May 30, 2022.

[41] Timothy Fraser, Lee Badger, and Mark Feldman. Hardening COTS software with generic software wrappers. In *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, volume 2, pages 323–337. IEEE, 2000.

[42] Freemyband. Free my band. `https://www.freemyband.com/`. Online; Retrieved August 25, 2022.

[43] Frida. Dynamic instrumentation toolkit for developers, reverse-engineers, and security researchers. `https://frida.re`, 2023. Online; Retrieved September 30, 2022.

[44] Robert Furberg, Julia Brinton, Michael Keating, and Alexa Ortiz. Crowd-sourced Fitbit datasets 03.12.2016-05.12.2016, May 2016.

[45] Gadgetbridge. Gadgetbridge. `https://gadgetbridge.org/`. Online; Retrieved August 25, 2022.

[46] GeoIP. GeoIP Lookup Tool. `https://geoip.com/`, 2023. Online; Retrieved September 30, 2022.

[47] Konstantinos Georgiou, Andreas V Larentzakis, Nehal N Khamis, Ghadah I Alsuhaibani, Yasser A Alaska, and Elias J Giallafos. Can wearable devices accurately measure heart rate variability? A systematic review. *Folia medica*, 60(1):7–20, 2018.

[48] Lodovico Giaretta, Ioannis Savvidis, Thomas Marchioro, Šarūnas Girdzijauskas, George Pallis, Marios D Dikaiakos, and Evangelos Markatos. PDS 2: A user-centered decentralized marketplace for privacy preserving data processing. In *2021 IEEE 37th International Conference on Data Engineering Workshops (ICDEW)*, pages 92–99. IEEE, 2021.

[49] Neil Zhenqiang Gong and Bin Liu. You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 979–995, 2016.

[50] Neil Zhenqiang Gong and Bin Liu. Attribute inference attacks in online social networks. *ACM Transactions on Privacy and Security (TOPS)*, 21(1):1–30, 2018.

[51] Google. HTTPS encryption on the web. `https://transparencyreport.google.com/https/overview?hl=en`, 2023, February 19. Online; Retrieved February 25, 2023.

[52] Rohit Goyal, Nicola Dragoni, and Angelo Spognardi. Mind the tracker you wear: a security analysis of wearable health trackers. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 131–136, 2016.

[53] Jiaping Gui, Ding Li, Mian Wan, and William GJ Halfond. Lightweight measurement and estimation of mobile ad energy consumption. In *Proceedings of the 5th international workshop on green and sustainable software*, pages 1–7, 2016.

[54] Jiaping Gui, Stuart Mcilroy, Meiyappan Nagappan, and William GJ Halfond. Truth in advertising: The hidden cost of mobile ads for software developers. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 1, pages 100–110. IEEE, 2015.

[55] Naman Gupta, Vinayak Naik, and Srishti Sengupta. A firewall for internet of things. In *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, pages 411–412. IEEE, 2017.

[56] Mostafa Haghi, Kerstin Thurow, and Regina Stoll. Wearable devices in medical internet of things: scientific research and commercially available devices. *Healthcare informatics research*, 23(1):4–15, 2017.

[57] J Arthur Harris and Francis G Benedict. A biometric study of human basal metabolism. *Proceedings of the National Academy of Sciences of the United States of America*, 4(12):370, 1918.

[58] Wajih Ul Hassan, Saad Hussain, and Adam Bates. Analysis of Privacy Protections in Fitness Tracking Social Networks-or-You can run, but can you hide? In *27th USENIX Security Symposium (USENIX Security 18)*, pages 497–512, 2018.

[59] Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. Preventing private information inference attacks on social networks. *IEEE Transactions on Knowledge and Data Engineering*, 25(8):1849–1862, 2012.

[60] Steven G Hershman, Brian M Bot, Anna Shcherbina, Megan Doerr, Yasbanoo Moayedi, Aleksandra Pavlovic, Daryl Waggott, Mildred K Cho, Mary E Rosenberger, William L Haskell, et al. Physical activity, sleep and cardiovascular health data for 50,000 individuals from the MyHeart Counts Study. *Scientific data*, 6(1):1–10, 2019.

[61] Andrew Hilts, Christopher Parsons, and Jeffrey Knockel. Every step you fake: A comparative analysis of fitness tracker privacy and security. *Open Effect Report*, 76(24):31–33, 2016.

[62] Andrew Hintz. Fingerprinting websites using traffic analysis. In *International workshop on privacy enhancing technologies*, pages 171–178. Springer, 2002.

[63] Guannan Hu and Kensuke Fukuda. Toward detecting iot device traffic in transit networks. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 525–530. IEEE, 2020.

[64] IDC. Wearable Devices Market Share. `https://www.idc.com/promo/wearablevendor`, 2022, December 21. Online; Retrieved February 30, 2023.

[65] IDC. Wearables Deliver Double-Digit Growth for Both Q4 and the Full Year 2021, According to IDC. `https://www.idc.com/getdoc.jsp?containerId=prUS48935722`, 2022, March 8. Online; Retrieved February 30, 2023.

[66] Sana Imtiaz, Muhammad Arsalan, Vladimir Vlassov, and Ramin Sadre. Synthetic and private smart health care data generation using GANs. In *2021 International Conference on Computer Communications and Networks (ICCCN)*, pages 1–7. IEEE, 2021.

[67] Pankush Kalgotra, Uzma Raja, and Ramesh Sharda. Growth in the development of health and fitness mobile apps amid COVID-19 pandemic. *Digital Health*, 8:20552076221129070, 2022.

[68] Andrei Kazlouski, Thomas Marchioro, Harry Manifavas, and Evangelos Markatos. Do you know who is talking to your wearable smartband? *Integrated Citizen Centered Digital Health and Social Care*, page 142, 2020.

[69] Andrei Kazlouski, Thomas Marchioro, Harry Manifavas, and Evangelos Markatos. Do partner apps offer the same level of privacy protection? The case of wearable applications. In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pages 648–653. IEEE, 2021.

[70] Andrei Kazlouski, Thomas Marchioro, Harry Manifavas, and Evangelos P Markatos. I still See You! Inferring Fitness Data from Encrypted Traffic of Wearables. In *HEALTH-INF*, pages 369–376, 2021.

[71] Andrei Kazlouski, Thomas Marchioro, and Evangelos Markatos. I just wanted to track my steps! Blocking unwanted traffic of Fitbit devices. In *Proceedings of the 12th International Conference on the Internet of Things*, pages 96–103, 2022.

[72] Andrei Kazlouski., Thomas Marchioro., and Evangelos Markatos. What your Fitbit Says about You: De-anonymizing Users in Lifelogging Datasets. In *Proceedings of the 19th International Conference on Security and Cryptography - SECRYPT,*, pages 341–348. INSTICC, SciTePress, 2022.

[73] Daniel Kelly, Kevin Curran, and Brian Caulfield. Automatic prediction of health status using smartphone-derived behavior profiles. *IEEE journal of biomedical and health informatics*, 21(6):1750–1760, 2017.

[74] Hee Jin Kim, Kang Hyun Lee, Jung Hun Lee, Hyun Youk, and Hee Young Lee. The Effect of a Mobile and Wearable Device Intervention on Increased Physical Activity to Prevent Metabolic Syndrome: Observational Study. *JMIR mHealth and uHealth*, 10(2):e34059, 2022.

[75] Jong Wook Kim, Jong Hyun Lim, Su Mee Moon, Hoon Yoo, and Beakcheol Jang. Privacy-preserving data collection scheme on smartwatch platform. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–4. IEEE, 2019.

[76] Bih-Hwang Lee, Ervin Kusuma Dewi, and Muhammad Farid Wajdi. Data security in cloud computing using AES under HEROKU cloud. In *2018 27th Wireless and Optical Communication Conference (WOCC)*, pages 1–5. IEEE, 2018.

[77] Myeonggeon Lee, Kyungmook Lee, Jaewoo Shim, Seong-je Cho, and Jongmoo Choi. Security threat on wearable services: Empirical study using a commercial smartband. In *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pages 1–5. IEEE, 2016.

[78] Sang-Ho Lee, Yeongmi Ha, Mira Jung, Seungkyoung Yang, and Won-Seok Kang. The effects of a mobile wellness intervention with Fitbit use and goal setting for workers. *Telemedicine and e-Health*, 25(11):1115–1122, 2019.

[79] Ding Li, Shuai Hao, Jiaping Gui, and William GJ Halfond. An empirical study of the energy consumption of android applications. In *2014 IEEE International Conference on Software Maintenance and Evolution*, pages 121–130. IEEE, 2014.

[80] Ming Li, Shucheng Yu, Kui Ren, and Wenjing Lou. Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings. In *International conference on security and privacy in communication systems*, pages 89–106. Springer, 2010.

[81] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd international conference on data engineering*, pages 106–115. IEEE, 2006.

[82] Marc Liberatore and Brian Neil Levine. Inferring the source of encrypted HTTP connections. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 255–263, 2006.

[83] Jack Lindamood, Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. Inferring private information using social network data. In *Proceedings of the 18th international conference on World wide web*, pages 1145–1146, 2009.

[84] Wen Ming Liu, Lingyu Wang, Pengsu Cheng, Kui Ren, Shunzhi Zhu, and Mourad Debbabi. PPTP: Privacy-preserving traffic padding in web-based applications. *IEEE Transactions on Dependable and Secure Computing*, 11(6):538–552, 2014.

[85] Xiangyu Liu, Zhe Zhou, Wenrui Diao, Zhou Li, and Kehuan Zhang. When good becomes evil: Keystroke inference with smartwatch. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1273–1285, 2015.

[86] Yang Liu and Zhenjiang Li. aleak: Context-free side-channel from your smart watch leaks your typing privacy. *IEEE Transactions on Mobile Computing*, 19(8):1775–1788, 2019.

[87] Deborah Lupton. *The quantified self*. John Wiley & Sons, 2016.

[88] Kelvin Ly and Yier Jin. Security studies on wearable fitness trackers. In *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE*, 2016.

[89] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3–es, 2007.

[90] Anindya Maiti, Ryan Heard, Mohd Sabra, and Murtuza Jadliwala. Towards inferring mechanical lock combinations using wrist-wearables as a side-channel. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pages 111–122, 2018.

[91] Anindya Maiti, Murtuza Jadliwala, Jibo He, and Igor Bilogrevic. (Smart) watch your taps: Side-channel keystroke inference attacks using smartwatches. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 27–30, 2015.

[92] Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. Protecting sensory data against sensitive inferences. In *Proceedings of the 1st Workshop on Privacy by Design in Distributed Systems*, pages 1–6, 2018.

[93] Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. Mobile sensor data anonymization. In *Proceedings of the international conference on internet of things design and implementation*, pages 49–58, 2019.

[94] Anna Maria Mandalari, Daniel J Dubois, Roman Kolcun, Muhammad Talha Paracha, Hamed Haddadi, and David Choffnes. Blocking without breaking: Identification and mitigation of non-essential iot traffic. *arXiv preprint arXiv:2105.05162*, 2021.

[95] Anna Maria Mandalari, Roman Kolcun, Hamed Haddadi, Daniel J Dubois, and David Choffnes. Towards automatic identification and blocking of non-critical iot traffic destinations. *arXiv preprint arXiv:2003.07133*, 2020.

[96] Thomas Marchioro., Andrei Kazlouski., and Evangelos Markatos. User Identification from Time Series of Fitness Data. In *Proceedings of the 18th International Conference on Security and Cryptography - SECRYPT,*, pages 806–811. INSTICC, SciTePress, 2021.

[97] Thomas Marchioro, Andrei Kazlouski, and Evangelos Markatos. How to Publish Wearables' Data: Practical Guidelines to Protect User Privacy. *Studies in Health Technology and Informatics*, 294:949–950, 2022.

[98] Thomas Marchioro, Andrei Kazlouski, and Evangelos P Markatos. Practical Crowdsourcing of Wearable IoT Data with Local Differential Privacy. In *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation*, pages 275–287, 2023.

[99] Yair Meidan, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis. In *Proceedings of the symposium on applied computing*, pages 506–509, 2017.

[100] David B Meinert, Dane K Peterson, John R Criswell, and Martin D Crossland. Privacy policy statements and consumer willingness to provide personal information. *Journal of Electronic Commerce in Organizations (JECO)*, 4(1):1–17, 2006.

[101] Ralph C Merkle. Secure communications over insecure channels. *Communications of the ACM*, 21(4):294–299, 1978.

[102] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. Iot sentinel: Automated device-type identification for security enforcement in iot. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2177–2184. IEEE, 2017.

[103] Tejaswini Mishra, Meng Wang, Ahmed A Metwally, Gireesh K Bogu, Andrew W Brooks, Amir Bahmani, Arash Alavi, Alessandra Celli, Emily Higgs, Orit Dagan-Rosenfeld, et al. Pre-symptomatic detection of COVID-19 from smartwatch data. *Nature biomedical engineering*, 4(12):1208–1220, 2020.

[104] Hooman Mohajeri Moghaddam, Gunes Acar, Ben Burgess, Arunesh Mathur, Danny Yuxing Huang, Nick Feamster, Edward W Felten, Prateek Mittal, and Arvind Narayanan. Watching you watch: The tracking ecosystem of over-the-top tv streaming devices. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 131–147, 2019.

[105] State of California Department of Justice. California consumer privacy act (ccpa). `https://oag.ca.gov/privacy/ccpa`, 2023, February 15. Online; Retrieved February 30, 2023.

[106] County of Santa Clara Office of the District Attourney. Suspect in Fitbit Murder Dies in Custody. `https://countyda.sccgov.org/news/news-release/suspect-fitbit-murder-dies-custody`. Online; Retrieved September 30, 2022.

[107] Wellington Oliveira, Renato Oliveira, and Fernando Castor. A study on the energy consumption of android app development approaches. In *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, pages 42–52. IEEE, 2017.

[108] OpenHumans. Open Humans Fitbit Connection. `https://www.openhumans.org/activity/fitbit-connection`, February 2016.

[109] OWASP. Certificate and Public Key Pinning. `https://owasp.org/www-community/controls/Certificate_and_Public_Key_Pinning`, 2018. Online; Retrieved February 25, 2022.

[110] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P Markatos. The cost of digital advertisement: Comparing user and advertiser views. In *Proceedings of the 2018 World Wide Web Conference*, pages 1479–1489, 2018.

[111] Abhinav Parate, Meng-Chieh Chiu, Chaniel Chadowitz, Deepak Ganesan, and Evangelos Kalogerakis. Risq: Recognizing smoking gestures with inertial sensors on a wristband. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 149–161, 2014.

[112] Radio Perlman. An overview of PKI trust models. *IEEE network*, 13(6):38–43, 1999.

[113] Portswigger. What do you want to do with Burp Suite? `https://portswigger.net/burp`, 2023. Online; Retrieved September 30, 2022.

[114] Yazdan Ahmad Qadri, Ali Nauman, Yousaf Bin Zikria, Athanasios V Vasilakos, and Sung Won Kim. The future of healthcare internet of things: a survey of emerging technologies. *IEEE Communications Surveys & Tutorials*, 22(2):1121–1167, 2020.

[115] Giorgio Quer, Pishoy Gouda, Michael Galarnyk, Eric J Topol, and Steven R Steinhubl. Inter-and intraindividual variability in daily resting heart rate and its associations with age, sex, sleep, BMI, and time of year: Retrospective, longitudinal cohort study of 92,457 adults. *Plos one*, 15(2):e0227709, 2020.

[116] Giorgio Quer, Jennifer M Radin, Matteo Gadaleta, Katie Baca-Motes, Lauren Ariniello, Edward Ramos, Vik Kheterpal, Eric J Topol, and Steven R Steinhubl. Wearable sensor data and self-reported symptoms for COVID-19 detection. *Nature Medicine*, 27(1):73–77, 2021.

[117] Jennifer M Radin, Nathan E Wineinger, Eric J Topol, and Steven R Steinhubl. Harnessing wearable device data to improve state-level real-time surveillance of influenza-like illness in the USA: a population-based study. *The Lancet Digital Health*, 2(2):e85–e93, 2020.

[118] Andrew Raij, Animikh Ghosh, Santosh Kumar, and Mani Srivastava. Privacy risks emerging from the adoption of innocuous wearable sensors in the mobile environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 11–20, 2011.

[119] Jingjing Ren, Daniel J Dubois, David Choffnes, Anna Maria Mandalari, Roman Kolcun, and Hamed Haddadi. Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach. In *Proceedings of the Internet Measurement Conference*, pages 267–279, 2019.

[120] Jakob Rieck. Attacks on fitness trackers revisited: A case-study of unfit firmware security. *arXiv preprint arXiv:1604.03313*, 2016.

[121] Said Jawad Saidi, Anna Maria Mandalari, Hamed Haddadi, Daniel J Dubois, David Choffnes, Georgios Smaragdakis, and Anja Feldmann. Detecting consumer IoT devices through the lens of an ISP. In *Proceedings of the Applied Networking Research Workshop*, pages 36–38, 2021.

[122] Said Jawad Saidi, Anna Maria Mandalari, Roman Kolcun, Hamed Haddadi, Daniel J Dubois, David Choffnes, Georgios Smaragdakis, and Anja Feldmann. A haystack full of needles: Scalable detection of iot devices in the wild. In *Proceedings of the ACM Internet Measurement Conference*, pages 87–100, 2020.

[123] Munshi Saifuzzaman, Tajkia Nuri Ananna, Mohammad Jabed Morshed Chowdhury, Md Sadek Ferdous, and Farida Chowdhury. A systematic literature review on wearable health data publishing under differential privacy. *International Journal of Information Security*, pages 1–26, 2022.

[124] Pierangela Samarati. Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[125] Allen Sarkisyan, Ryan Debbiny, and Ani Nahapetian. WristSnoop: Smartphone PINs prediction using smartwatch motion sensors. In *2015 IEEE international workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2015.

[126] Mustafizur R Shahid, Gregory Blanc, Zonghua Zhang, and Hervé Debar. IoT devices recognition through network traffic analysis. In *2018 IEEE international conference on big data (big data)*, pages 5187–5192. IEEE, 2018.

[127] Katie Shilton. Four billion little brothers? Privacy, mobile phones, and ubiquitous data collection. *Communications of the ACM*, 52(11):48–53, 2009.

[128] Jaewoo Shim, Kyeonghwan Lim, Jaemin Jeong, Seong-je Cho, Minkyu Park, and Sangchul Han. A case study on vulnerability analysis and firmware modification attack for a wearable fitness tracker. *IT Converg. Pract*, 5(4):25–33, 2017.

[129] Similarweb. Effortlessly Analyze Your Competitive Landscape. `https://www.similarweb.com/`, 2023. Online; Retrieved September 30, 2022.

[130] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. Classifying IoT devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing*, 18(8):1745–1759, 2018.

[131] Arunan Sivanathan, Hassan Habibi Gharakheili, and Vijay Sivaraman. Inferring iot device types from network behavior using unsupervised clustering. In *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, pages 230–233. IEEE, 2019.

[132] Arunan Sivanathan, Daniel Sherratt, Hassan Habibi Gharakheili, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. Characterizing and classifying IoT traffic in smart cities and campuses. In *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 559–564. IEEE, 2017.

[133] Snort. Snort - Network Intrusion Detection & Prevention System. `https://www.snort.org/`, 2023. Online; Retrieved September 30, 2022.

[134] Statista. Number of adblock users worldwide from 2013 to 2019. `https://www.statista.com/statistics/435252/adblock-users-worldwide/`, aug 2020. Online; Retrieved February 25, 2023.

[135] Statista. Wearables unit shipments worldwide from 2014 to 2021. `https://www.statista.com/statistics/437871/wearables-worldwide-shipments/`, 2022, March 22. Online; Retrieved May 30, 2023.

[136] Statista. Wearables unit shipments worldwide by vendor from 1st quarter 2014 to 3rd quarter 2022. `https://www.statista.com/statistics/435933/quarterly-wearables-shipments-worldwide-by-vendor/`, 2023, February 16. Online; Retrieved February 30, 2023.

[137] Nili Steinfeld. "I agree to the terms and conditions":(How) do users read privacy policies online? An eye-tracking experiment. *Computers in human behavior*, 55:992–1000, 2016.

[138] Raoul Strackx and Frank Piessens. Fides: Selectively hardening software application components against kernel-level or process-level malware. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 2–13, 2012.

[139] Strava. Strava releases 2020 Year In Sport Data Report. `https://blog.strava.com/press/yis2020/`. Online; Retrieved September 30, 2022.

[140] Orathai Sukwong, Hyong Kim, and James Hoe. Commercial antivirus software effectiveness: an empirical study. *Computer*, 44(03):63–70, 2011.

[141] Qixiang Sun, Daniel R Simon, Yi-Min Wang, Wilf Russell, Venkata N Padmanabhan, and Lili Qiu. Statistical identification of encrypted web browsing traffic. In *Proceedings 2002 IEEE Symposium on Security and Privacy*, pages 19–30. IEEE, 2002.

[142] Latanya Sweeney. Simple demographics often identify people uniquely. *Health (San Francisco)*, 671(2000):1–34, 2000.

[143] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[144] Cong Tang, Keith Ross, Nitesh Saxena, and Ruichuan Chen. What's in a name: A study of names, gender inference, and gender behavior in facebook. In *International Conference on Database Systems for Advanced Applications*, pages 344–356. Springer, 2011.

[145] Vajira Thambawita, Steven Alexander Hicks, Hanna Borgli, Håkon Kvale Stensland, Debesh Jha, Martin Kristoffer Svensen, Svein-Arne Pettersen, Dag Johansen, Håvard Dagenborg Johansen, Susann Dahl Pettersen, et al. Pmdata: a sports logging dataset. In *Proceedings of the 11th ACM Multimedia Systems Conference*, pages 231–236, 2020.

[146] Theguardian. Greek helicopter pilot found guilty of murdering British wife Caroline Crouch. `https://www.theguardian.com/uk-news/2022/may/16/greek-helicopter-pilot-who-killed-british-wife-caroline-crouch-guilty`. Online; Retrieved September 30, 2022.

[147] Theguardian. Fitness tracking app Strava gives away location of secret US army bases. `https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases`, 2018, January. Online; Retrieved May 30, 2022.

[148] The New York Times. Robert Shields, Wordy Diarist, Dies at 89. `https://www.nytimes.com/2007/10/29/us/29shields.html`. Online; Retrieved September 30, 2022.

[149] Huong Ly Tong, Carol Maher, Kate Parker, Tien Dung Pham, Ana Luisa Neves, Benjamin Riordan, Clara K Chow, Liliana Laranjo, and Juan C Quiroz. The use of mobile apps and fitness trackers to promote healthy behaviors during COVID-19: A cross-sectional survey. *PLOS Digital Health*, 1(8):e0000087, 2022.

[150] uBlock Origin. uBlock Origin - Free, open-source ad content blocker. `https://ublockorigin.com/`, aug 2022. Online; Retrieved February 25, 2023.

[151] Janus Varmarken, Hieu Le, Anastasia Shuba, Athina Markopoulou, and Zubair Shafiq. The tv is smart and full of trackers: Measuring smart tv advertising and tracking. *Proceedings on Privacy Enhancing Technologies*, 2020(2), 2020.

[152] Jessica Vitak, Yuting Liao, Priya Kumar, Michael Zimmer, and Katherine Kritikos. Privacy attitudes and data valuation among fitness tracker users. In *International Conference on Information*, pages 229–239. Springer, 2018.

[153] Chen Wang, Xiaonan Guo, Yingying Chen, Yan Wang, and Bo Liu. Personal PIN leakage from wearable devices. *IEEE Transactions on Mobile Computing*, 17(3):646–660, 2017.

[154] He Wang, Ted Tsung-Te Lai, and Romit Roy Choudhury. Mole: Motion leaks through smartwatch sensors. In *Proceedings of the 21st annual international conference on mobile computing and networking*, pages 155–166, 2015.

[155] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. Collecting and analyzing multidimensional data with local differential privacy. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 638–649. IEEE, 2019.

[156] Shulan Wang, Junwei Zhou, Joseph K Liu, Jianping Yu, Jianyong Chen, and Weixin Xie. An efficient file hierarchy attribute-based encryption scheme in cloud computing. *IEEE Transactions on Information Forensics and Security*, 11(6):1265–1277, 2016.

[157] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[158] Connecticut's Official State WEbsite. Jury Convicts Richard Dabate of the December 2015 Murder of His Wife, Connie Dabate. `https://portal.ct.gov/DCJ/Press-Room/Press-Releases/051022Dabate`. Online; Retrieved September 30, 2022.

[159] WEF. Fitness apps grew by nearly 50% during the first half of 2020, study finds. `https://www.weforum.org/agenda/2020/09/fitness-apps-gym-health-downloads/`. Online; Retrieved September 30, 2022.

[160] Whois. Whois Domain Lookup. https://www.whois.com/whois, 2023. Online; Retrieved September 30, 2022.

[161] Wireshark. Go Deep. https://www.wireshark.org/, 2023. Online; Retrieved September 30, 2022.

[162] Pamela J Wisniewski, Bart P Knijnenburg, and Heather Richter Lipford. Making privacy personal: Profiling social network users to inform privacy education and nudging. *International Journal of human-computer studies*, 98:95–108, 2017.

[163] Gilbert Wondracek, Thorsten Holz, Engin Kirda, and Christopher Kruegel. A practical attack to de-anonymize social network users. In *2010 ieee symposium on security and privacy*, pages 223–238. IEEE, 2010.

[164] Disabled World. Height Chart of Men and Women in Different Countries. Disabled World. www.disabled-world.com/calculators-charts/height-chart.php, 2017, December 1. Online; Retrieved May 2, 2022.

[165] Guowu Xie, Marios Iliofotou, Ram Keralapura, Michalis Faloutsos, and Antonio Nucci. Subflow: Towards practical flow-level traffic classification. In *2012 Proceedings IEEE INFOCOM*, pages 2541–2545. IEEE, 2012.

[166] Mengmeng Yang, Lingjuan Lyu, Jun Zhao, Tianqing Zhu, and Kwok-Yan Lam. Local differential privacy and its applications: A comprehensive survey. *arXiv preprint arXiv:2008.03686*, 2020.

[167] Sofia Yfantidou, Christina Karagianni, Stefanos Efstathiou, Athena Vakali, Joao Palotti, Dimitrios Panteleimon Giakatos, Thomas Marchioro, Andrei Kazlouski, Elena Ferrari, and Šarūnas Girdzijauskas. LifeSnaps, a 4-month multi-modal dataset capturing unobtrusive snapshots of our lives in the wild. *Scientific Data*, 9(1):663, Oct 2022.

[168] Sofia Yfantidou, Christina Karagianni, Stefanos Efstathiou, Athena Vakali, Joao Palotti, Dimitrios Panteleimon Giakatos, Thomas Marchioro, Andrei Kazlouski, Elena Ferrari, and Šarūnas Girdzijauskas. LifeSnaps: a 4-month multi-modal dataset capturing unobtrusive snapshots of our lives in the wild, July 2022. Version 2: Conversion from 7z to zip for easier uncompressing Version 3: Deletion of unnecessary files Version 4: Updates CSV files after bug fix in the export process.

[169] YoungPTone. Husband detects wife's pregnancy from abnormal Fitbit data. https://np.reddit.com/r/fitbit/comments/445ppj/hr_reading_consistently_high_last_few_days/, 2016, February. Online; Retrieved May 30, 2022.

[170] Qiaoyang Zhang and Zhiyao Liang. Security analysis of bluetooth low energy based smart wristbands. In *2017 2nd International Conference on Frontiers of Sensors Technologies (ICFST)*, pages 421–425. IEEE, 2017.

[171] Yang Zhao, Jun Zhao, Mengmeng Yang, Teng Wang, Ning Wang, Lingjuan Lyu, Dusit Niyato, and Kwok-Yan Lam. Local differential privacy-based federated learning for internet of things. *IEEE Internet of Things Journal*, 8(11):8836–8853, 2020.

[172] Guokang Zhu, Jia Li, Zi Meng, Yi Yu, Yanan Li, Xiao Tang, Yuling Dong, Guangxin Sun, Rui Zhou, Hui Wang, et al. Learning from large-scale wearable device data for predicting epidemics trend of COVID-19. *Discrete Dynamics in Nature and Society*, 2020, 2020.