1-1-2020

Behavioral Privacy Risks and Mitigation Approaches in Sharing of Wearable Inertial Sensor Data

Nazir Saleheen

BEHAVIORAL PRIVACY RISKS AND MITIGATION APPROACHES IN SHARING OF WEARABLE INERTIAL SENSOR DATA

by

Nazir Saleheen

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

Major: Computer Science

The University of Memphis

August 2020

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Dr. Santosh Kumar for mentoring me to grow as a research scientist. His consistent guidance, encouragement, and inspiration helped me to pursue my research. I would like to thank my committee members, Dr. Paul Balister, Dr. Nirman Kumar, Dr. Mani Srivastava, and Dr. Deepak Venugopal, for their constructive comments and suggestions, and serving as my committee members. I am immensely fortunate to have had them as an advisor.

My appreciation also extends to my lab members for their encouragement and generous support. Countless thanks and appreciation to Dr. Amin Ahsan Ali, Dr. Md. Mahbubur Rahman, Dr. Hillol Sarker, Dr. Monowar Hossain, Dr. Sudip Vhaduri, Rummana Bari, Soujanya Chatterjee, Sayma Akhter, Md. Azim Ullah, Shiplu Hawlader, Mithun Saha, Rabin Banjade, and Sameer Neupane.

A special thanks to my parents: Md. Golam Gous and Salma Sultana, who always support and guide me, for all of the sacrifices that they have made for me. Without their support, I would have never come this far. I would like to thank my brother Nazir Arefeen and my sister Rabeya Sultana for their endless care and love.

My work in graduate school would not have been possible without the unconditional support from my wife, Sayma Akther. She has always stood beside me like a rock and has taken care of every aspect of our life. She puts me at my ease, and for that, I consider myself a lucky man.

This work was supported by the National Science Foundation and by the National Institutes of Health.

Finally, I would like to thank the University of Memphis for providing me with an excellent research platform as well as the constant support throughout my academic career.

ABSTRACT

Saleheen, Nazir. Ph.D. The University of Memphis. August 2019. Behavioral privacy risks and mitigation approaches in sharing of wearable inertial sensor data. Major Professor: Dr. Santosh Kumar

Wrist-worn inertial sensors in activity trackers and smartwatches are increasingly being used for daily tracking of activity and sleep. Wearable devices, with their onboard sensors, provide appealing mobile health (mHealth) platform that can be leveraged for continuous and unobtrusive monitoring of an individual in their daily life. As a result, an adaptation of wrist-worn devices in many applications (such as health, sport, and recreation) increases. Additionally, an increasing number of sensory datasets consisting of motion sensor data from wrist-worn devices are becoming publicly available for research. However, releasing or sharing these wearable sensor data creates serious privacy concerns of the user. First, in many application domains (such as mHealth, insurance, and health provider), user identity is an integral part of the shared data. In such settings, instead of identity privacy preservation, the focus is more on the behavioral privacy problem that is the disclosure of sensitive behaviors from the shared sensor data. Second, different datasets usually focus on only a select subset of these behaviors. But, in the event that users can be re-identified from accelerometry data, different databases of motion data (contributed by the same user) can be linked, resulting in the revelation of sensitive behaviors or health diagnoses of a user that was neither originally declared by a data collector nor consented by the user.

The contributions of this dissertation are multifold. First, to show the behavioral privacy risk in sharing the raw sensor, this dissertation presents a detailed case study of detecting cigarette smoking in the field. It proposes a new machine learning model, called *puffMarker*, that achieves a false positive rate of 1/6 (or 0.17) per day, with a recall rate of 87.5%, when tested in a field study with 61 newly abstinent daily smokers. Second, it proposes a model-based data substitution mechanism, namely *mSieve*, to protect behavioral privacy. It evaluates the efficacy of the scheme using 660 hours of raw sensor

data collected and demonstrates that it is possible to retain meaningful utility, in terms of inference accuracy (90%), while simultaneously preserving the privacy of sensitive behaviors. Finally, it analyzes the risks of user re-identification from wrist-worn sensor data, even after applying mSieve for protecting behavioral privacy. It presents a deep learning architecture that can identify unique micro-movement pattern in each wearer's wrists. A new consistency-distinction loss function is proposed to train the deep learning model for open set learning so as to maximize re-identification consistency for known users and amplify distinction with any unknown user. In 10 weeks of daily sensor wearing by 353 participants, we show that a known user can be re-identified with 99.7% true matching rate while keeping the false acceptance rate to 0.1% for an unknown user. Finally, for mitigation, we show that injecting even a low level of Laplace noise in the data stream can limit the re-identification risk. This dissertation creates new research opportunities on understanding and mitigating risks and ethical challenges associated with behavioral privacy.

TABLE OF CONTENTS

LIST OF FIGURES

Figures **Pages**

LIST OF TABLES

Tables **Pages**

Chapter 1

Introduction

Sensor technologies are becoming popular day by day, especially in the field of mobile health (mHealth). According to the National Purchase Diary Panel (NPD group) report, smartwatch sales were up more than 60 percent in the US last year[1]. These devices make it possible to continuously and remotely monitor individuals in their natural settings. More and more health features are integrating into these devices. As a result, more people are using these wearable technologies as devices begin to mature.

Researchers developed several ubiquitous technologies to detect health outcomes from these wearable inertial sensor data such as smoking [2, 3], eating [4], brushing, and flossing [5]. Activities like stress [6], drug usage [7], alcohol usage, pain, anxiety, opioid usage, or cocaine usage [7] can be detected using variability in the physiology of the body from wearable sensors.

On the one hand, some of the above inferences, such as walking, conversation, eating, are extremely useful in the investigation of behavioral risk factors on health and wellness. But, on the other hand, inferences such as smoking, cocaine usage, and stress may be sensitive to the user and need to be kept private. Thus, we have a conundrum, where the same time series data can be used for making both utility providing inferences (that are desirable) and also sensitive inferences (that need to be protected).

To facilitate the development of these biomarkers, researchers have already released several datasets [8–10] containing raw sensor data. To prevent linkage attack, each released database usually strips the data of any identifiers and is typically anonymized using recommended practices (e.g., using k-anonymity [11], l-diversity [12], and t-closeness [13]) before release. But, they consist of raw sensor data streams assuming low risk of re-identification.

[1] https://www.theverge.com/2019/2/13/18223272/smartwatch-sales-growth-us-npd-group-apple-samsung-fitbit

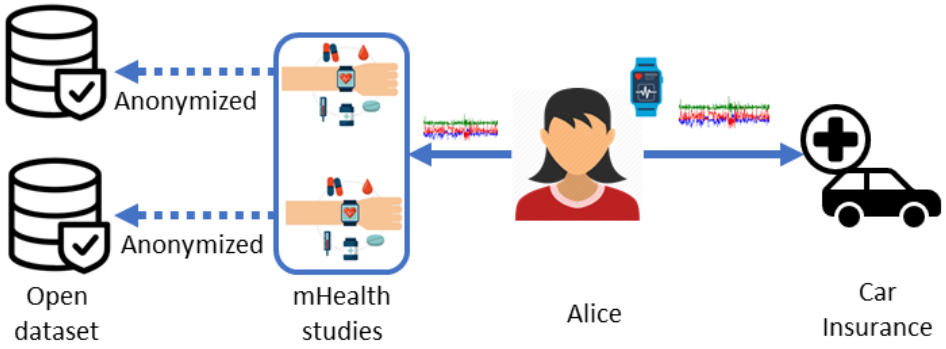


Fig. 1.1: Example scenario for two types of sensor data sharing

1.1 Problem Setting

A typical scenario of data sharing involves two parties: data owner or user and data recipient, where the data owner shares the data, and the data recipient consumes data. Consider a scenario, as shown in Figure 1.1, where Alice participates in several mHealth studies, such as smoking study and opioid study, and shares wrist-worn inertial sensor data along with smoking behaviors, tremor, and opioid usage. By taking the consent from Alice, researchers release these datasets as an open dataset. To achieve anonymization goal, suppose the identifiers (such as name, address, and SSN) and quasi-identifiers (such as gender, birth date, and zip code) are removed or perturbed from the dataset. Suppose Alice also subscribes to be safe and save policy from car insurance. As part of this policy, Alice caries a smartwatch that measures her driving quality, and in return, she gets a low premium. In this scenario, the insurance company has both the identity of Alice and motion sensor data. On the other hand, the released datasets do not contain any identifiable information about Alice except raw sensor data and ground truth (e.g., smoking events, tremor, and opioid usage). Here we can group the various privacy problems of Alice into two broad classes:

- **Behavioral privacy:** In our example, Alice's identity (user-id) is known to the insurance company (data recipient). For this class of problems, the user-id is not concealed in the shared data. Instead, there exists a specific set of sensitive

inferences which the data owner wants to protect and a specific set of activities that can be shared. In our example, the desirable inferences, such as driving quality and amount of driving, form a whitelist which the user wants to share. However, the same sensor data can be used to infer smoking and tremor – inferences sensitive to the user. The sensitive inferences compose the blacklist, which the user wants to keep private. Alice wants to know the behavioral privacy risks: is it possible to detect sensitive behavior like smoking with high accuracy? If yes, then the privacy problem is to design a system that will take as input the whitelist and blacklist of inferences and translate them into privacy actions on the shared sensors such that the conditions on the lists are satisfied.

- **Identity Privacy:** The second concern is identity privacy: can the insurance company learn of Alice's prior history with smoking, pain, tremor, etc., from public datasets that Alice has contributed to previously to help advance science. The data of open datasets are syntactically sanitized by stripping it of personally identifiable information before sharing (using an anonymization technique). Only the unsanitized data is raw sensor data streams. Now the privacy question is: does the raw motion data capture the user's unique character that can be used to e-identify the user? If yes, then the insurance company can de-anonymize the open dataset and learn additional information about Alice.

1.2 Overview of contributions

In this dissertation, we focus on the privacy risks and mitigation approaches while sharing raw sensor data, and make the following primary contributions.

1.2.1 Detection of a Sensitive Behaviors

Chapter 2 presents a detailed case study of detecting cigarette smoking. We propose a new method called *puffMarker* to detect smoking puffs that is sufficiently robust for use in smoking cessation studies. We adopt an explainable modeling approach so as to obtain better interpretability and generalizability. *puffMarker* uses data collected from two

wearable sensors — breathing pattern captured from a respiration sensor and hand gestures captured using 6-axis inertial sensors (3-axis accelerometers and 3-axis gyroscopes) worn on wrists. We observe that during smoking, the hand comes to the mouth, and is immediately followed by a deep inhalation. We propose a novel method of data reduction by identifying windows of data that represent the hand reaching the mouth, being at the mouth, and leaving the mouth. We then develop a machine learning model to classify each remaining candidate window into puff or non-puff and propose two post-processing rules to further limit false alarms. This work demonstrates that sensitive behaviors such as smoking can be detected with high accuracy from wearable motion sensor data.

1.2.2 Protecting Behavioral Privacy while Sharing Raw Sensor Data

Sharing of mobile sensor data, especially physiological data, raise different privacy challenges, that of protecting private behaviors, such as smoking, that can be revealed from time series of sensor data. Existing privacy mechanisms rely on noise addition and data perturbation. But the accuracy requirement on inferences drawn from physiological data, together with well-established limits within which these data values occur, render traditional privacy mechanisms inapplicable.

In Chapter 3, we define a new behavioral privacy metric based on differential privacy and propose a novel data substitution mechanism to protect behavioral privacy. Instead of random substitution of sensitive segments, which can degrade the utility of the overall dataset, we perform the model-based substitution. We employ a Dynamic Bayesian Network model that allows us to search for plausible user-specific candidate segments that satisfy the statistics of the sensitive segment and thus preserve the overall consistency of the shared data. Through experimentation on real-life physiological datasets, we demonstrated that our substitution strategies can indeed be used for preserving the utility of inferences while achieving behavioral privacy.

1.2.3 WristPrint: User Re-identification Risks from Wrist-worn Accelerometry Data

Wrist-worn inertial sensors in activity trackers and smartwatches are increasingly being used for daily tracking of activity and sleep. New research shows that the same sensor data can also be used to detect other daily behaviors such as eating, drinking, brushing, flossing, and some sensitive states such as smoking, drinking, tremors, pain, and drug use. Different datasets usually focus on only a select subset of these behaviors. But, in the event that users can be re-identified from accelerometry data, different databases of motion data (contributed by the same user) can be linked, resulting in the revelation of sensitive behaviors or health diagnoses of a user that was neither originally declared by the data collector nor consented by the user.

In Chapter 4, we formulate the problem of user re-identification from these motion sensor data as an open set learning problem and propose a new consistency-distinction loss function to train a deep learning model. Using 70,600 minutes of sensor data from 353 participants, we show that a known user can be re-identified with a 99.7% true matching rate while keeping the false accept rate to 0.1% for any unknown new user.

1.3 Thesis organization

The remainder of this dissertation is organized as follows: We begin by presenting the case study of cigarette smoking in Chapter2. Chapter 3 presents a solution for protecting behavioral privacy. Finally, Chapter 4 presents user reidentification risks from wrist-worn motion sensor data.

Chapter 2

Detection of a Sensitive Behavior: A Case Study of Cigarette Smoking

Recent researches have demonstrated the feasibility of detecting smoking from wearable sensors, but their performance on real-life smoking lapse detection is unknown. In this chapter, we propose a new model and evaluate its performance on 61 newly abstinent smokers for detecting a first lapse.

2.1 Introduction

Smoking accounts for nearly one of every five deaths in the United States [14, 15]. Smoking cessation rates are improving with advances in treatments and interventions, but are still in single digits. A primary hurdle in achieving a higher success rate is a lack of methods that can intervene or deliver treatment at the right moment when an abstinent smoker is most vulnerable [16]. Advances in mobile technology have created an opportunity to deliver an intervention anytime and anywhere if a potential smoking lapse event can be predicted in advance. However, to find sensor-based predictors of a smoking lapse [17–26] (e.g., rapid rise in stress [27] or proximity to a tobacco outlet), timing of a lapse event (especially the first lapse event, the most clinically relevant event as it usually leads to full relapse [24]) needs to be determined accurately. Then, data mining methods can be applied on the time series of mobile sensor data to identify the antecedents and precipitants of a smoking lapse (in a smoking cessation study). The traditional method of self-reporting a smoking lapse event [24, 28–30] lacks the temporal precision needed to identify predictors in a continuous stream of sensor data.

There have been several recent works on detecting a smoking episode from wearable sensors. They include tracking hand gestures during smoking by inertial sensors worn on the wrist [31, 32] and tracking deep inhalation and exhalation in the breathing pattern via Respiratory Inductive Plethysmography (RIP) sensors [33, 34]. But, their performance is reported mostly on training data collected in supervised settings, with the exception of RisQ [31], which was tested on 4 smokers who wore 9-axis inertial sensors

for 4 hours a day over 3 days. They report detection of 27 smoking episodes (out of 30) and report a false positive rate of 2/3 per day (8 sessions out of 12 person days).

While RisQ is the most promising smoking detection method, it uses a 9-axis wrist sensor, whereas most modern smartwatches (e.g., Microsoft Band, Apple Watch) have only 6-axis inertial sensors, for better energy-efficiency. Since RisQ relies on quaternions that need all 9-axis, it is not clear how to adapt this method to 6 axis. Most importantly, none of the above described methods have been evaluated on data collected in a smoking cessation study and hence their performance for first lapse detection is not known.

In this paper, we propose a new method called *puffMarker* to detect smoking puffs that is sufficiently robust for use in smoking cessation studies. We adopt an explainable modeling approach so as to obtain better interpretability and generalizability. *puffMarker* uses data collected from two wearable sensors — breathing pattern captured from a RIP sensor and hand gestures captured using 6-axis inertial sensors (3-axis accelerometers and 3-axis gyroscopes) worn on wrists. Since a participant may use both hands to smoke, they are provided two wrist sensors to wear, one on each wrist. Both sensors nicely complement each other and hence provide a better detection accuracy. To provide an intuition of the benefit of using these two diverse sensors, we show signals captured during smoking, walking, and eating in Figure 2.1. We only show y-axis for the inertial sensors since it has distinct pattern for hand gesture. We observe that during smoking, the hand comes to the mouth, and is immediately followed by a deep inhalation. During walking, the hand is downwards with a pendulum like movement and respiration is faster. During eating (cereal), the hand comes at the mouth, however deep inhalation, observed during smoking, is absent in such activities [35]. We later describe details of the *puffMarker* model (see Figure 2.4 and its accompanying description).

We train the model on 40 hours of data from 6 regular smokers, where each of the 470 puffs were carefully marked. In 10-fold cross-validation on the training data, the model achieves a recall rate of 96.9%, for a false positive rate of 1.1%. We applied the

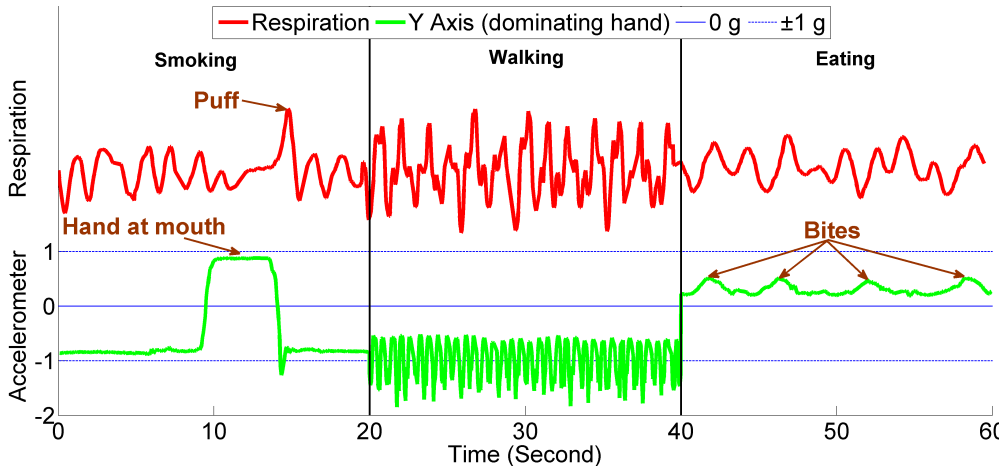


Fig. 2.1: Comparison of respiration and wrist accelerometer (y-axis) signal between smoking, walking, and eating.

puffMarker model to a smoking cessation study with 61 participants, where each participant wore the sensors for one day while smoking ad lib and for 3 days since quitting. Among 61 participants, 33 lapsed within three days (verified by a CO monitor) — 17 lapsed on the first day, 12 on the second day, and 4 on the third day. We apply our model on these data and report 7 key findings.

1. **Recall:** Among 33 lapsers, one is eliminated due to high data loss; Of the remaining 32, first lapse is detected in 28.
2. **False Positives:** When tested on 20 abstinent days from 32 lapsers, only two false episodes are detected. When tested on 84 abstinent days (946 hours) of data from 28 abstainers, false episode per day is limited to 1/6.
3. **Lapse Progression:** The average number of smoking episodes is 1.1 on the lapse day, 2.75 on the day after lapse, and 3.56 on 2 days after lapse.
4. **Puff Count:** A regular smoking session contains an average of 15 puffs, but the first lapse episode contains an average of only 6.5 puffs. Number of puffs in a smoking episode increases to 9.5 puffs on the day after lapse and 11 puffs on 2 days after lapse.
5. **Temporal Inaccuracy of Self-report:** Out of 28 first lapse events detected by

puffMarker, 9 were not self-reported, 15 were reported (an average of 41 minutes) after lapse, and 4 were reported (an average of 12.7 minutes) before lapse.

6. **Recall Inaccuracy:** Nine lapsers who did not self-report, recalled the lapse time upon CO verification in the lab next day. Temporal inaccuracy for these recalls was larger, ranging from 107 minutes before to 205 minutes after.
7. **Hand Pattern:** Among 61 smokers, 33 smoke using the right hand, 18 use the left hand, and 10 use both hands.

In summary, ours is the first work to show that precise moment of first lapse can indeed be detected in a real-life smoking cessation study. Given the critical nature of the first lapse in smoking cessation (as it marks the first event in cessation failure and usually leads to full relapse [24]), this work lays the groundwork for development of mobile-based just in-time intervention for smoking cessation.

2.2 Related Work

We discuss related works that could be considered for detecting first lapse in a smoking cessation study. There have been several recent works on finding methods to detect smoking episodes from sensor data. The E-cigarette can record the timing of puffs and smart lighters can detect when it is lit [36]. Either one can detect the timing of the first lapse, but only if participants remember to use these devices at the time of their first lapse. Alternatively, if the smoking spot is under video surveillance, then the timing of the first lapse can be detected [37], but only if the participant is under surveillance at the time of their first lapse.

For detecting smoking of regular cigarettes without any instrumentation of lighters or being under surveillance, several wearable sensor-based methods have been proposed. They include tracking hand gestures during smoking by inertial sensors worn on the wrist [31, 32] and tracking deep inhalation and exhalation in the breathing pattern via respiratory inductive plethysmography (RIP) sensors [33, 34].

In [33], 161 puffs were collected from 10 participants while they wore a

respiration sensor to capture breathing pattern. A machine learning model obtained a precision of 0.91 and recall of 0.81 in 10-fold cross validation. In [34], 20 participants wore a radio frequency sensor on the wrist and on the collar to track hand reaching mouth. They also wore a respiration sensor to capture breathing pattern. They performed 12 activities, including smoking in different postures. An average precision of 0.87 and a recall of 0.81 was reported.

In [32], 6 participants wore 4 accelerometers (wrist and upper arm of dominant hand, other wrist and ankle) and performed smoking and other activities for a total of 11.8 hours (consisting of 34 smoking episodes or 481 puffs). Recall and precision rates of upper seventies and lower eighties is obtained for a machine learning model.

The above works demonstrate a potential for detection of smoking events via experimentation in controlled setting. A recent work, RisQ [31], reported evaluation of a smoking detection model from wrist sensors in the field environment. In this work, data from 15 volunteers were collected for a total of 17 smoking episodes for training. The smoking episodes included smoking alone, in a group while having a conversation and smoking while walking around. The volunteers wore a 9-axis inertial measurement unit (IMU) on the wrist for an average of 2 hours each. Out of 369 puffs and 5,228 other gestures collected over 28 hours, their model achieves a precision of 0.91 and recall of 0.81. They applied their model on 4 users who wore 9-axis inertial sensors for 4 hours each on 3 days in the field. On this field dataset, they reported a recall rate of 90% (27 out of 30 sessions detected) and a false positive rate or 2/3 episodes per day (8 false sessions in 12 person days).

Although our work builds upon [31] and [33], it makes several novel contributions. First, RisQ [31] uses 9-axis wrist sensor (3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer), whereas our method uses 6 axis IMU (supported by many modern smartwatches such as Microsoft Band and Apple Watch). Since gesture recognition is invariant to the absolute orientation of the subject in the earth's inertial frame, 6 axis IMU

provides sufficient degrees of freedom. Second, prior work [31] relies on generic gesture recognition algorithms based on inertial tracking of the absolute orientation of the wrist, we developed a lightweight recognition algorithm tailored for the smoking gesture with much reduced computational complexity and sampling rate requirements. Third, mPuff [33] classified respiration cycles into puff and non-puff, but has a high false alarm rate; it falsely detects 150 out of 1,000 respiration cycles as puffs, making it ineffective for use in the natural environment. In contrast, our *puffMarker* model falsely detects only 1 out of 1,117 respiration cycles as puffs. Fourth, ours is the first work that was applied to data collected from a real-life smoking cessation study. All other prior works reported their results on only regular smoking training data. Fifth, our work is the first one to detect first lapses, which is most challenging due to significantly smaller number of puffs (45%). Sixth, RisQ and mPuff were both evaluated on only 4 users in the field environment while we evaluate on 61 users, making our work clearly the largest-ever study for sensor-based detection of smoking. Seventh, ours is the first work that combines respiration and wrist movement data and shows how inclusion of wrist movement detection can increase the performance of respiration based detector [33]. Finally, performance of our system (recall of 96.9% and false positive rate of 1.1%) is better than any previously reported work even on training data.

2.3 Data Collection

We describe details of the user study for collecting training data for the *puffMarker* model and the smoking cessation study where the *puffMarker* model was applied.

Wearable Sensor Suite: Participants in both studies wore a wireless physiological sensor suite (AutoSense [38]) underneath their clothes. The wearable sensor suite consisted of two-lead electrocardiograph (ECG), 3-axis accelerometer, and respiration sensors. Participants also wore an inertial sensor on each wrist that includes a 3-axis accelerometer and a 3-axis gyroscope. Each sensor transmitted the sensor data continuously to a mobile phone. AutoSense respiration sensor has it's own battery and it

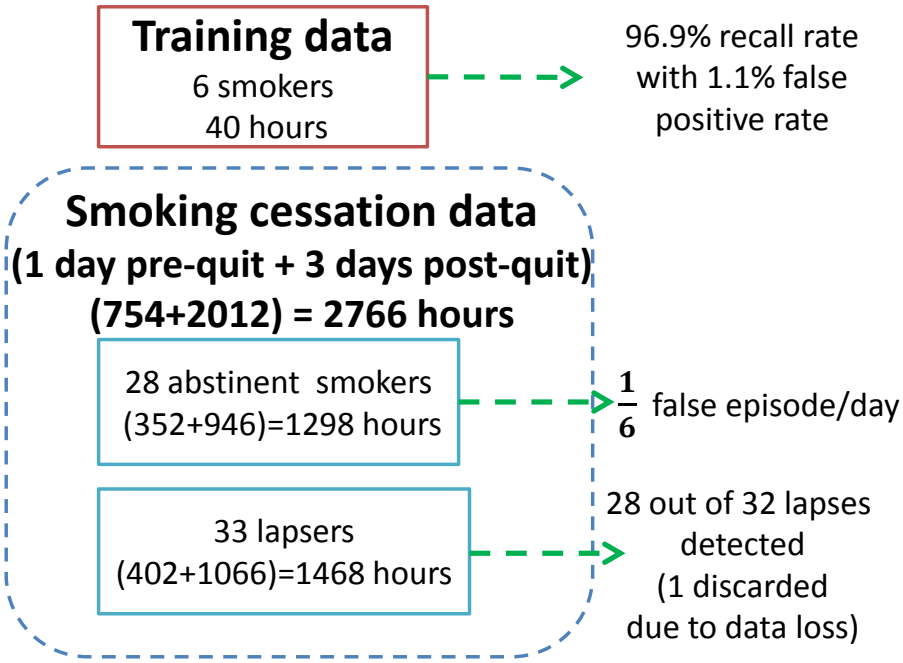


Fig. 2.2: Data Summary of Training and Smoking Cessation Study

lasts for 10 days on a 750 mAh battery. It uses a low powered ANT Radio to connect with the phone. The phone (which collects GPS data continuously and keeps its wireless radio on for data reception) lasts for 13 hours on a single charge. The smartwatch we use lasts 3 days on a 500 mAh battery. The sampling rate for the respiration sensor is 21.3 Hz and that for the accelerometer and gyroscope on smartwatch are 16 Hz for each of the six axes.

Mobile Phone: Participants were given a smart phone to carry. It receives and stores data from sensors on the body and on the phone. It also collects self-reports in response to random prompts which capture characteristics of situational factors associated with smoking. These factors include stress, physical activity levels, posture, places visited, and commuting episodes. In the training study, an observer marks the timing of each puff on the phone. In the smoking cessation study, participants used the phone to report the beginning of smoking episodes by pressing a button.

2.3.1 Data Collection for Model Training

We collected data from 6 daily smokers. They wore the sensors for a total of 40.3 hours in field as they went about their daily lives. Each time they smoked, they were accompanied by an observer who who was instructed to mark (on study phone) a puff when the participant held cigarette between the lips and inhaled smoke. From the marking, we thus obtained the timing when the hand is at the mouth during smoking. This dataset contains 32 smoking episodes (that includes smoking while standing, sitting, walking, and being in a conversation) with 470 puff markings. In 179 instances out of the 470 puff markings, there were wireless data losses and noise due to physical movement or loosening of the respiration belt. We use the remaining 291 puff instances for which we have acceptable respiration and wrist sensor data.

2.3.2 Smoking Cessation Study

Participants: The participants were cigarette smokers who reported smoking 10 or more cigarettes per day for at least 2 years, and who reported high motivation to quit. To qualify, participants had to pass a screening session prior to being enrolled in the study. The screening includes assessment of current medical and mental health status and history of any major medical and psychiatric illness. Screening also includes assessment of smoking behavior, mood, and other behavioral health measures. Participants were excluded if they had ongoing major medical or psychiatric problems and if they had other comorbid psychiatric and substance use problems. Also, participants who were not entertained into the normal day/light diurnal cycle were excluded to control for variation in diurnal physiological activity and behaviors.

Protocol: Once enrolled, the participants picked a smoking quit date. Two weeks prior to their quit date, subjects wore the sensor suite for 24 hours in their natural environment. After completion of the 24 hour monitoring, which we call the pre-quit session, subjects come back to the lab for their second visit. Smoking cessation counseling is provided starting at this second visit to the lab. Then the subjects come back

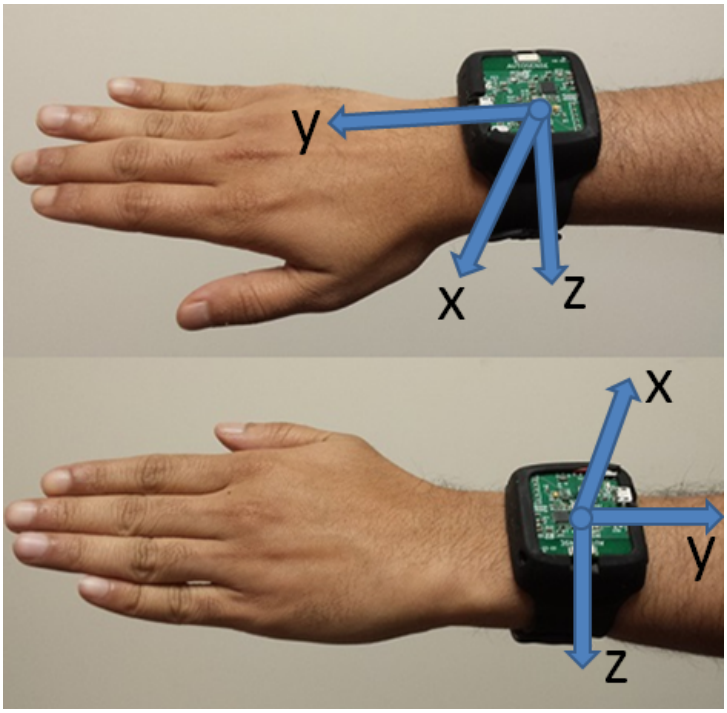


Fig. 2.3: Depiction of mounting of inertial sensors on the wrist and the orientation of their axes.

to the lab on the assigned quit date to attend a counseling session and to begin the 72 hour of monitoring in the field; this we refer to as the post-quit session. They come back to the lab each day to confirm smoking status by capturing an expired breath sample in a carbon monoxide (CO) monitor. During each day of monitoring (24 hour pre-quit and 72 hour post-quit), the participants wear the sensor suite during awake hours, and on the mobile phone, complete 12 Ecological Momentary Assessments (EMAs) (i.e., self-reports) daily.

Data Collected: We collected data from 61 participants. The participants wore the sensor suite for a total of 2,766 hours. A summary of collected data is shown in Figure 2.2.

2.4 Data Processing and Model Development

Respiration, 3-axis accelerometers, and 3-axis gyroscopes provide us with 7 concurrent time series of data that are all time-stamped when they are received on the mobile phone[1]. The x, y and z axes of the accelerometers and gyroscopes on the wristband

[1]We note that timestamping of sensor data upon receipt on the phone does not adversely affect time snchronization needed for our method. Data from sensors are transmitted to the phone tens of times each

are aligned with each other. The directions of the axes of the wristband sensors are shown in Figure 2.3.

2.4.1 Overview of *puffMarker* Model

Figure 2.4 presents an overview of all the data analysis steps. First, we remove outliers and impute missing data. Second, we describe a method to detect hand-at-mouth gestures that segments (i.e., creates windows in) the time series of sensors data. These windows are assessed for representing a puff. Hand gesture during puffing a cigarette is typically composed of three sub-gestures that usually occur in the following sequence — hand moving to the mouth, hand being at mouth while taking a puff, and hand moving away from the mouth.

puffMarker locates puff events by detecting the segments in the wrist sensor time series that contain hand-at-mouth gestures. When an accelerometer is stationary, any accelerometer axis aligned precisely with the earth's downward gravitational field will result in a measurement of $-1g$ in that axis. Due to the mounting of the sensor on the wrist (see Figure 2.3), we observe a positive value on y-axis when the right hand is held in an upward direction, whereas we get a negative value on the y-axis of the left wrist accelerometer.

However, when an accelerometer is moving, an axis will measure the combination of linear acceleration due to movement and the component of gravity in the direction of that axis. The gyroscope axes, on the other hand, measure the rate of change of rotation or angular velocity about the axes. We use the measurements obtained from both the accelerometers and gyroscope to detect hand-at-mouth gestures.

As a third step, we develop criteria to screen out segments (or windows) of data that do not correspond to smoking puffs. Using the detected hand-at-mouth segments, we identify the accompanying respiration cycles that potentially correspond to smoking puffs.

second and therefore the delay from sampling to reception on the phone is of the order of milliseconds. Our time granularity requirements for classification (discussed later) is of the order of 3-7 seconds, duration of hand staying at the mouth. Each respiration cycle is 3-6 seconds long. Hence, millisecond level errors in time synchronization between respiration and inertial sensor data does not adversely affect our model.

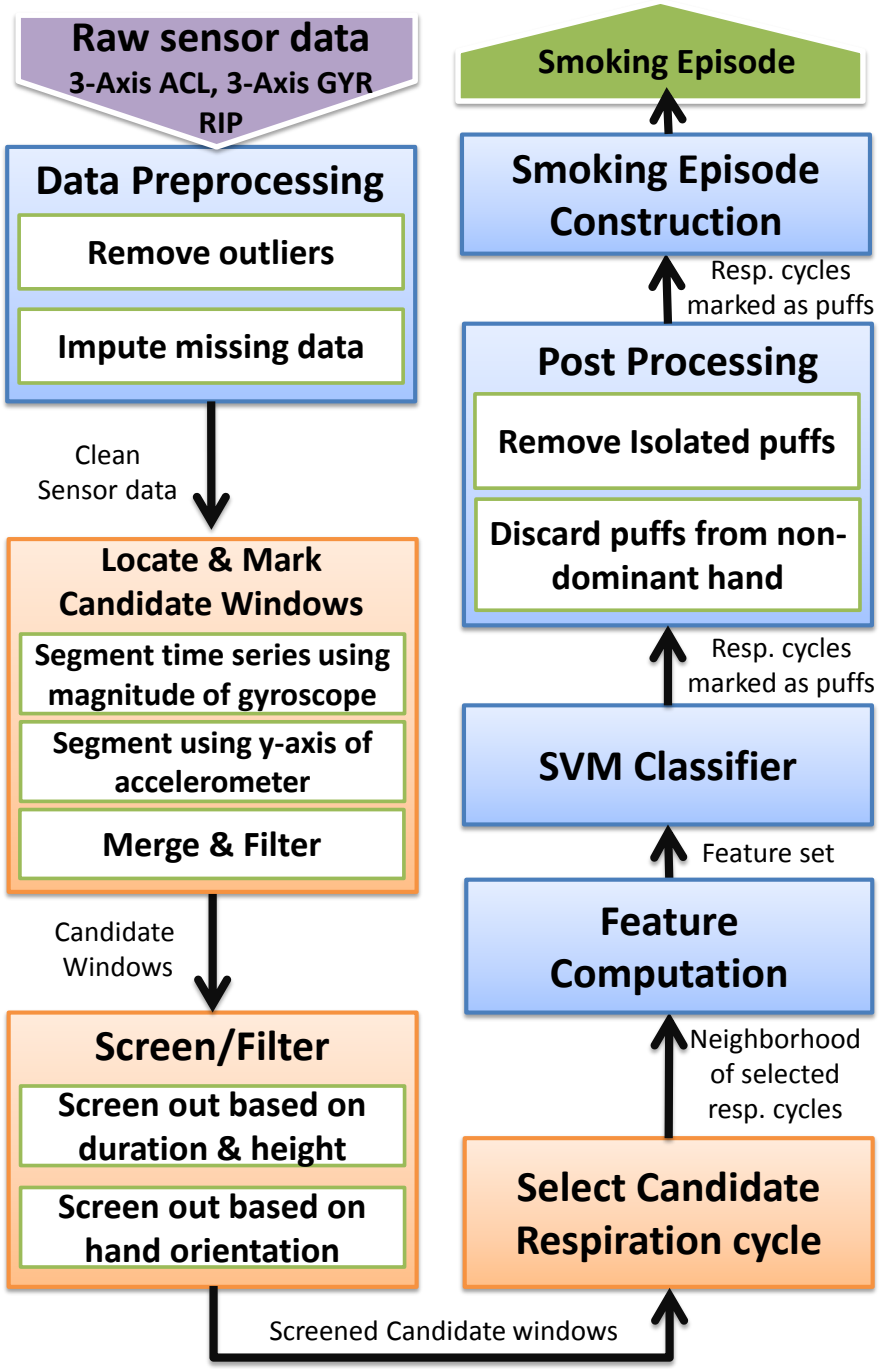


Fig. 2.4: An overview of key modeling steps for detecting smoking puffs and constructing smoking episodes in the *puffMarker* model.

Fourth, we compute features from those candidates' respiration cycles and hand-at-mouth segments that pass the preceding criteria. For classification, we train a support vector machine model that uses the input features to classify the qualifying candidate windows of data into *puff* and *non-puff*.

Fifth, we apply two simple post-processing steps to further reduce false alarms. Isolated puffs that do not fit within the distribution of inter-puff duration in a regular smoking episode are filtered out. Also, the model is applied to data collected from each wrist, but if a participant is observed to use only one hand for smoking during the pre-quit phase, then smoking puffs detected from non-dominant hand are filtered out. Finally, a smoking episode is constructed if sufficient number of smoking puffs (in a cluster) are detected in close vicinity.

2.4.2 Data Preprocessing

Sensor data are collected in the field environment where they are subjected to various sources of noises, losses, and degradation in quality. We develop a series of screening methods to clean the sensor data.

Removing outliers: Windows of sensor data when a participant is not wearing the sensors (determined using methods presented in [39]) are removed from analysis.

Imputation of missing data: In a real-life environment, some data packets are lost in wireless transmission. We interpolate if sample loss is limited to one packet containing 5 samples as described in [39]. Any longer loss burst is not imputed to maintain data quality. After outlier removal and imputation, the data is ready for further processing.

2.4.3 Locating and Marking Windows of Interest

We observe that during cigarette smoke inhalation, the hand must remain stationary at the mouth for few seconds. We look for these stationary moments in the inertial sensor data stream. We argue that for detecting hand-at-mouth gestures, this method of locating puffs is more robust than tracking the hand trajectory, which can vary widely depending on the body posture (i.e., sitting, standing, walking) and hand position.

In order to detect hand-at-mouth gestures, we first segment the sensor data from both wrists to find relatively stationary segments and discard all non-stationary segments. To find the body location when the hand is relatively stationary, we use the magnitude of the gyroscope axes. Any movement of the hand will manifest as a rotation about one of the gyroscope axes and cause the magnitude value to increase independent of the direction of rotation. When there is very little movement of the hand, the gyroscope magnitude will be low since it is not affected by gravity. Therefore, a hand-at-mouth gesture can be detected by finding segments where the gyroscope magnitude time series attains low values and is preceded and followed by peaks. The first peak is due to the hand moving towards the mouth and the second one is due to the hand moving away from the mouth.

It should, however, be noted that simple thresholding on the magnitude values to locate hand-at-mouth segments may not work well in practice. The average magnitude of a hand-at-mouth segment during walking is usually higher than that of stationary segments during standing or sitting. This is because, when a person is walking, the whole body is moving and there will always be some movement of the hands. Even when the hand is at mouth while taking a puff, there is some movement of the wrists due to taking steps.

Therefore, for the hand-at-mouth gestures during walking, we expect to find segments that attain low magnitude values compared to the average magnitude during walking. Moreover, in several instances, we observe that these relatively low amplitude values are sometimes higher than the amplitude of peaks corresponding to the hand movements before and after the hand-at-mouth stationary segments. The amplitude of these peaks depend on the rest position of the hand before and after the puff.

While standing or walking, the hand usually hangs beside the body and, therefore, we observe a larger peak amplitude. On the other hand, while sitting, the hands may be resting on the thighs, or on the armrest of a chair. In these cases, the hand is moving a shorter distance to the mouth and hence the peak amplitudes are lower. Also, sometimes, participants smoke with their hand hanging near the mouth. In such instances, the amount

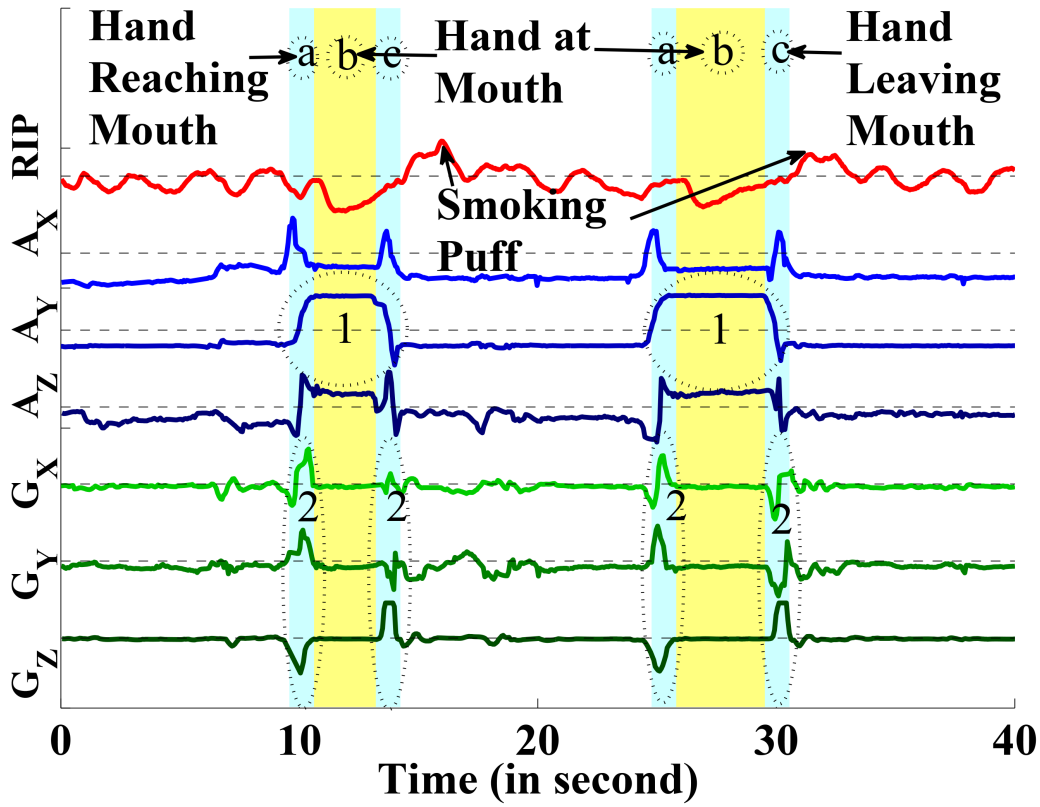


Fig. 2.5: Hand at mouth segment detection. (A_X, A_Y, A_Z) and (G_X, G_Y, G_Z) present the signals of accelerometers and gyroscopes. The circled area 1 represents the effect of y-axis of accelerometer when hand is at mouth and the circled area 2 represents the changes in gyroscope when hand is reaching the mouth and hand is leaving the mouth.

of movement is even lower, producing the lowest peak amplitudes. Therefore, setting the threshold value as high as the amplitude of hand-at-mouth segments during walking does not correctly identify hand-at-mouth segments in several other situations. This necessitates a segmentation method that is adaptive to the current level of movement so as to find segments that are relatively stationary.

We use a procedure that makes use of two moving averages to detect rise and fall in the gyroscope magnitude time series. Such methods are commonly used by investors in stock markets to identify price rise and fall [40]. More specifically, we perform the following steps.

1. We compute a fast (0.8 second window) and a slow (8 second window) moving

average of the gyroscope magnitude. The fast moving average closely follows the dynamic nature of the signal, while the slow moving average represents the level of movement in the neighborhood (see Figure 2.5). The window size of the slow moving average corresponds to the length of a smoking hand gesture length, which usually lasts from 3-7 seconds [41]. Because of this window size, during smoking hand-at-mouth gestures, the slow moving average is computed over windows that always contain the peaks due to the hand moving towards and away from the mouth. Therefore, during these segments, the slow moving average attains higher values compared to the fast moving average. The 0.8 second window size for the fast moving average is empirically chosen so that all segments that contain hand-at-mouth gestures during puffs are detected.

2. We select segments where the fast moving average lies below the slow moving average. These segments are demarcated by two consecutive crossing-over points of the two moving averages. The first of them corresponds to the location after which the fast moving average moves below the slower one and the second crossing-over point corresponds to the location after which the fast moving average rises above the slower one. The selected segments are, therefore, the ones where the magnitude values are relatively lower than the average magnitude of the neighborhood.

While segmenting data using the above method, we also check whether at that time y-axis of accelerometer is in an upward direction or not. Change in the magnitude of the gyroscope indicates that the hand is in motion. If, at that moment, the y-axis of the accelerometer changes from low to high (for the right hand sensor), it indicates the hand is moving in an upward direction. An opposite change in the y-axis of the accelerometer at the end of a gesture segment indicates the hand is moving in a downward direction. Figure 2.6 shows how segmentation is performed using both gyroscope and y-axis of accelerometer. This method of segment identification is designed to include all segments that are likely to contain a hand-at-mouth gesture, but may be over-inclusive.

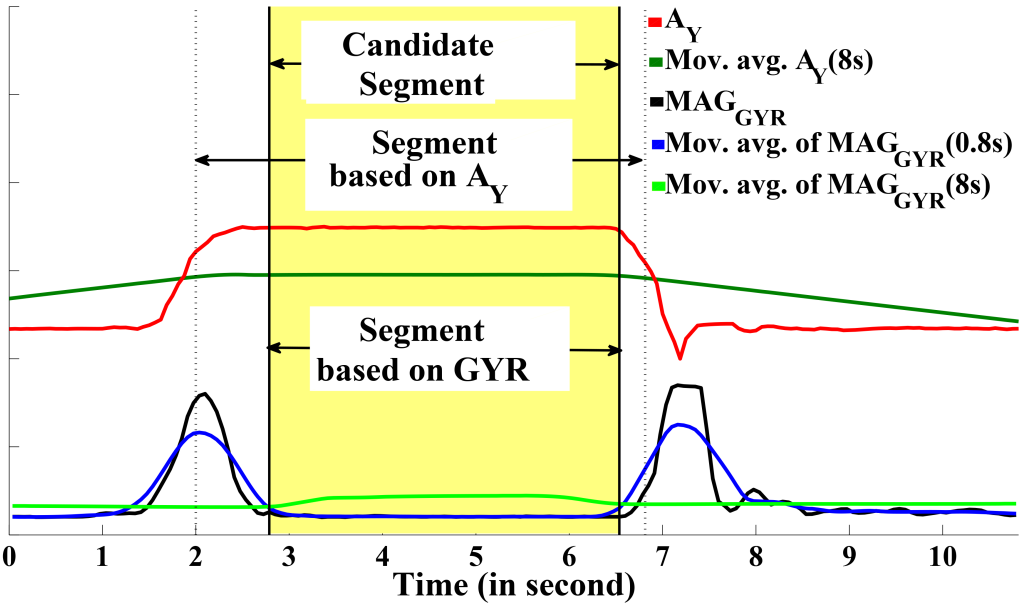


Fig. 2.6: Locating a candidate segment and identifying its boundaries.

2.4.4 Data Reduction via Non-candidate Segment Exclusion

We narrow down our search space by excluding hand-to-mouth segments in the inertial sensor data stream that are unlikely to represent a puff. We employ three methods, all of which are trained (i.e., determining parameters) using training data where each puff was carefully marked. In this dataset, we refer to inertial data segments that correspond to hand-at-mouth gestures when taking a puff as *puff segments*.

Appropriate Degree of Movement? We compute the mean difference between the fast and slow moving averages for each of the *puff segments*. Using the minimum of these distances as a threshold we discard all segments for which the mean distance is lower than this threshold (50 degree/second).

Appropriate Duration of Hand-to-mouth Gesture? We determine the duration of each *puff segments*. Inertial data segments that have a duration more that 3 standard deviations away from the mean duration (less than 0.8 seconds or greater than 5 seconds) are excluded.

Proper Hand Orientation When The Hand At Mouth? When the hand is at or

near the mouth, it may not be for taking a puff, such as when touching the hair, fixing eyeglasses, yawning, etc. To determine whether the hand is properly oriented during a hand-to-mouth gesture (e.g., as it usually is when taking a puff), we determine orientation of the hand by computing the pitch and roll angles.

The pitch and roll angles at a particular orientation indicate the amount of rotation about the x and y axis respectively required to reach the particular orientation from an initial orientation. We assume that in the initial orientation, the hand is kept horizontal with the palm facing down (z-axis is aligned with the gravitational field). Roll and pitch angles can be computed from either accelerometer or gyroscope. However, in the presence of linear acceleration, the orientation angles computed from accelerometers are usually less accurate. On the other hand, roll and pitch angles can be computed by integrating the angular velocity measurements obtained from gyroscope. However, a small error in angular velocity measurement may lead to large integration errors.

Since we are interested in computing the orientation at times when the hand is relatively stationary, relying on only accelerometer measurements is sufficiently accurate for our purpose. For each segment, we compute the average of each axis forming the vector (a_x, a_y, a_z). Following the method proposed in [42], we compute pitch(θ) = $(-a_y)/(-a_z)$ and roll(ϕ) = $a_x/\sqrt{{a_y}^2 + {a_z}^2}$.

We note that pitch and roll may suffer from gimbal lock, which refers to the lack of bi-continuous map between the spherical coordinates (pitch/roll) and the torus surface of the rotations. As far as the classification of the static orientation is concerned, the lack of continuous map is not an issue, the pitch/roll still uniquely defines the orientation and therefore each static orientation can be uniquely classified. The problem appears when dynamic measurements are averaged to get an average orientation in a window of data. This could be remedied going to an intermediate redundant representation such as quaternions to implement the averaging. We have chosen a simpler approach to address this problem. We use a window of measurements for computing pitch and roll. From the

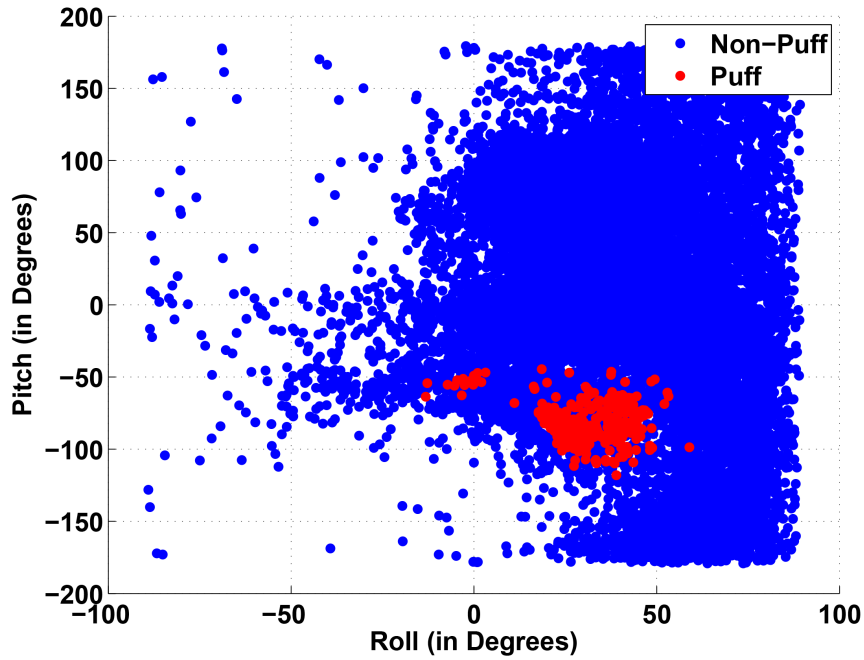


Fig. 2.7: Scatter plot of roll and pitch angles for the *puff segments* and non-*puff segments*.

window, we only take those values that do not suffer from Gimbal lock. Since our window consists of at least 0.8 seconds worth of samples, most of our windows have valid measurements for pitch and roll even if some values are momentarily affected by Gimbal lock due to orientation alignment.

We next describe our method for handling change in hands (left vs. right). Since the direction of the x-axis (and y-axis) of the accelerometer on the left and right wrists are opposite to each other, we first negate the x and y axis measurements of the left wrist sensor. In this way, the corresponding axes point in the same directions in both wrist sensors. By convention, the roll and pitch angles are positive when there is clockwise rotation about the y-axis and x-axis respectively. Therefore, when either arm is lifted from the horizontal position, pitch angles will have the same negative sign. However, inward (or outward) rotation of the left hand is in the opposite direction for an inward (or outward) rotation of the right hand. Therefore, in order to obtain the same sign for roll angles in both hands (that are)mirror images of each other, we negate the sign of the roll angles obtained for the left hand.

To determine proper values of pitch and roll, we compute these values for each *puff segments* and for non-*puff segments* (see Figure 2.7 for a scatter plot). We observe that the roll and pitch angles are slightly correlated with each other. For each segment, we compute the Mahalanabis distance[2] from the distribution of roll and pitch angles of *puff segments*. Mahalanabis distance d is computed by $d = (\mathbf{x} - \mu)S^{-1}(\mathbf{x} - \mu)'$, where $\mathbf{x} = (x_{roll}, x_{pitch})$ is the vector representing the roll and pitch angles of a segment and μ and S are the mean vector and co-variance matrix computed from roll and pitch angles of *puff segments*. Since for *puff segments*, the distance should be lower than other segments, we find a threshold t_d so that all *puff segments* are below it. We set the value of t_d to the largest value from all *puff segments* from training data and use it as a threshold. We discard all segments that have distance greater than $t_d = 10.15$ degree square.

2.4.5 Candidate Respiration Cycle Selection

After filtering out all non-candidate segments, we identify a respiration cycle that corresponds to each candidate hand-to-mouth gesture segment. We find the respiration cycles by computing the peak and valley locations in the respiration signal. Respiration signal reaches the peak once smoke is completely inhaled and exhalation of smoke usually occurs once cigarette is removed from the mouth. For each candidate hand-at-mouth gesture segment detected, we, therefore, select the first respiration cycle whose peak occurs after the end of the segment. This respiration cycle is a candidate for puff, if respiration signal is missing in that segment we use the hand-at-mouth segment as candidate.

A respiration cycle, however, can be associated with two different candidate segments, one from each hand, whose end times are close to each other. To avoid the situation where the training data contains a puff and a non-puff instance that are both

[2] An alternative measure could be Euclidean distance but the Euclidean distance is blind to correlated variables while the Mahalanabis distance takes the co-variances into account, which lead to elliptic decision boundaries in the 2D case, as opposed to the circular boundary in the Euclidean case.

associated with the same respiration cycle but different candidate segments, we only consider the non-smoking regions of the dataset as the source of non-puff instances.

2.4.6 Feature Computation

Respiration Features from [33]: From each respiration cycle, we compute the 17 respiration features presented in [33]. These features capture the characteristics of a respiration cycle and the relative changes in these characteristics. First, *Inhalation Duration*, *Exhalation Duration*, which correspond to the time required to breathe in and breathe out respectively are used as features. The next two features, *IE Ratio* and *Respiration Duration* are defined as the ratio of inhalation duration to the exhalation duration and their sum respectively. *Stretch* is defined to be the difference between the maximum (legitimate) amplitude and the minimum (legitimate) amplitude the signal attains within a respiration cycle.

Forward and backward first differences of a feature are defined as the difference between the value of this feature obtained from current respiration cycle and that from the next cycle and previous cycle, respectively. Since the smoking puff is different than neighboring respiration cycles, the forward and backward first differences of the values of the inhalation duration, exhalation duration, respiration duration, and stretch are also used as features to capture the relative changes in breathing pattern. Ratio of exhalation duration and stretch values to the average of the feature values of neighboring cycles are also used as features that capture the relative change in respiration. Finally, stretch of a respiration cycle is also divided into upper and lower parts with respect to the running mean of the valley amplitude of the respiration signal and these are used as features. The upper stretch magnitude is computed by taking the difference of peak amplitude and running mean value of the valley amplitudes of signal cycles (*ValleyAmplitudeMean*), while the lower stretch magnitude is computed by taking the absolute difference of minimum amplitude in a respiration cycle and *ValleyAmplitudeMean*.

New Respiration Features: In addition to the above features, we propose two

new features. These features are computed from the rate of change signal obtained by taking the first derivative of the respiration signal. The maximum and minimum values that the rate of change signal attains within a respiration cycle are used as features.

Inertial Features: From the candidate hand-at-mouth gesture segments, we compute the mean, median, standard deviation, and quartile deviation of **magnitude of gyroscope**, **pitch**, and **roll**. This gives us a total of 12 features.

2.4.7 Model Development

We extract features from both the candidate segments and the corresponding respiration cycle from the training data. We train a two-class Support Vector Machine (SVM) classifier using this training data to detect puffs.

2.4.8 Post Processing

After obtaining a classification from the SVM model, we conduct two post-processing steps. Our post-processing steps are similar in purpose to the use of random forests and conditional random field used in prior works [31, 32] to construct a smoking episode from individual puffs. We opt for rule-based methods for better explainability of the resulting model.

Remove Isolated Puffs: We call a detected puff an *isolated puff* if no other puff is within two standard deviations of the mean inter-puff duration (i.e., 28 ($\pm$18.6) seconds). An isolated puff is unlikely to be part of a smoking episode.

Discard Puffs from the Non-dominant Hand: We observe that among 61 participants, 33 always smoke using their right hand, 18 smoke using left hand, and only 10 switch hands, sometimes switching hands even within a smoking episode. This points to the utility of using wrist sensors on both hands in a smoking cessation study. But, since majority of the participants smoke using only their dominant hand, puffs detected from their non-dominant hand can be discarded.

Constructing a smoking episode: After removing isolated puffs, we are left with clusters of (2 or more) puffs in the data stream. We use a simple rule-based method to

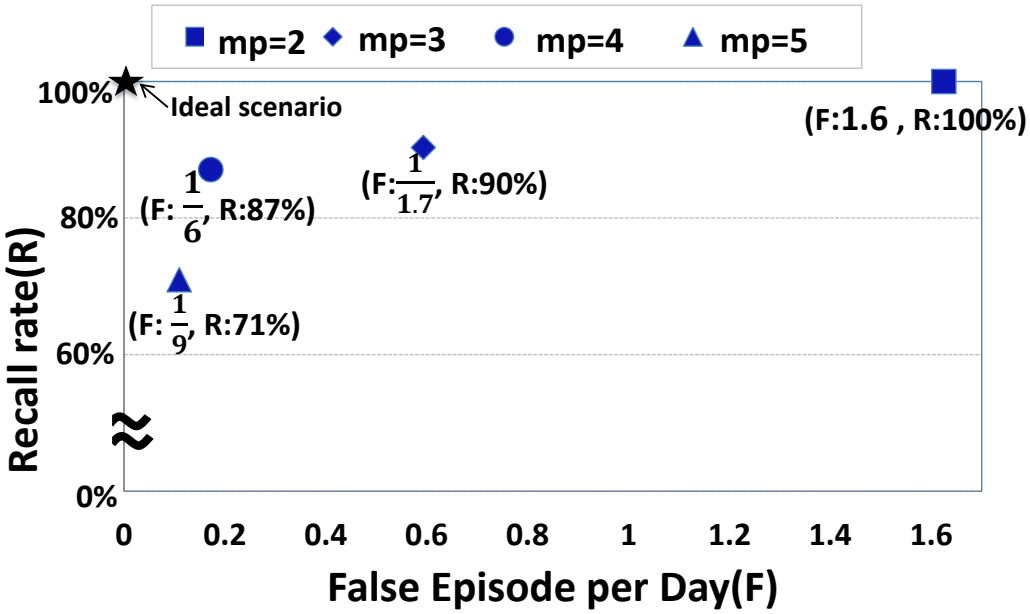


Fig. 2.8: Recall versus false episode per day for different value of *mp*

declare a cluster of puffs as a smoking episode, i.e., if it contains at least *mp* (minimum puff count) puffs. To find an appropriate value for *mp*, we analyze the recall rate for detecting first lapses in lapsers and false episode detection rate in abstinent smokers in our smoking cessation study data. Figure 2.8 presents the recall and false episode per day rates for different values of *mp*. We observe that the best result is achieved when *mp* = 4.

If a cluster of puffs contains at least four puffs then we called it as lapse episode. Figure 2.9 shows how Lapse episode is selected. Here one instance of user data is shown from 11:00 am to 8:00 pm. Self report about lapse was given at 5:50 pm. After applying puff detector we get three cluster of puffs. From this figure we see first and second cluster could not be a smoking episode because of less number of puffs while third one is a lapse episode.

2.5 Evaluation and Application

We now describe the performance of *puffMarker* on both training data and on the smoking cessation data.

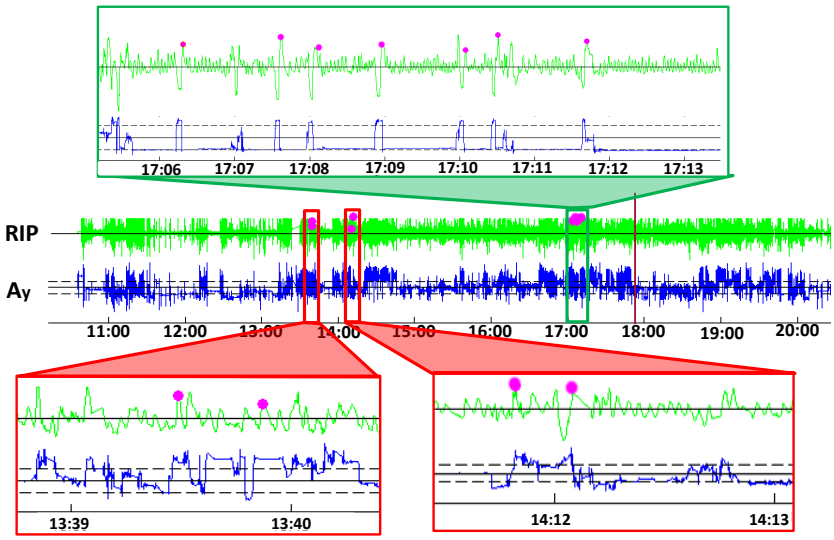


Fig. 2.9: Construction of Lapse episode.

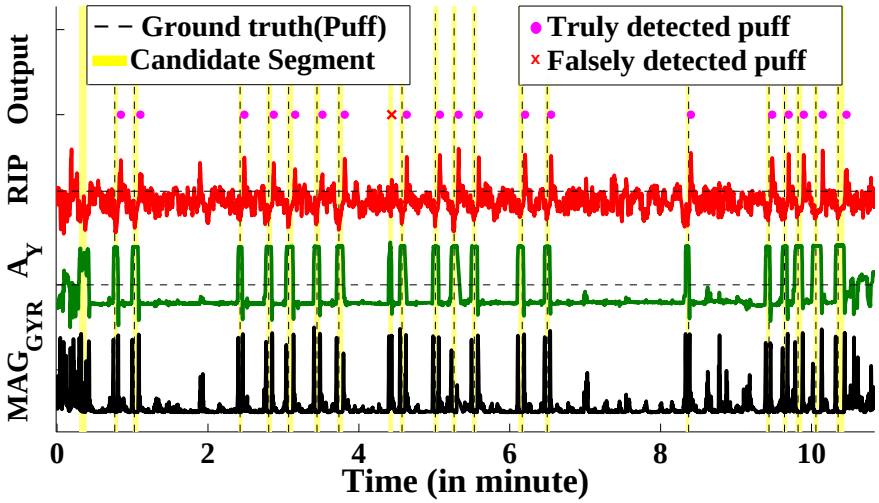


Fig. 2.10: Example of *puffMarker*'s Performance on training data during a smoking episode.

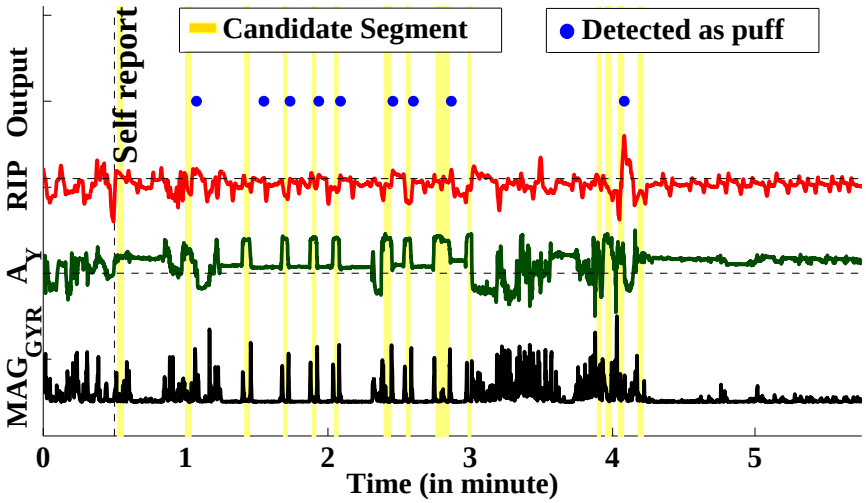


Fig. 2.11: Example of *puffMarker*'s Performance on the smoking cessation data during a smoking episode (for a lapsed participant).

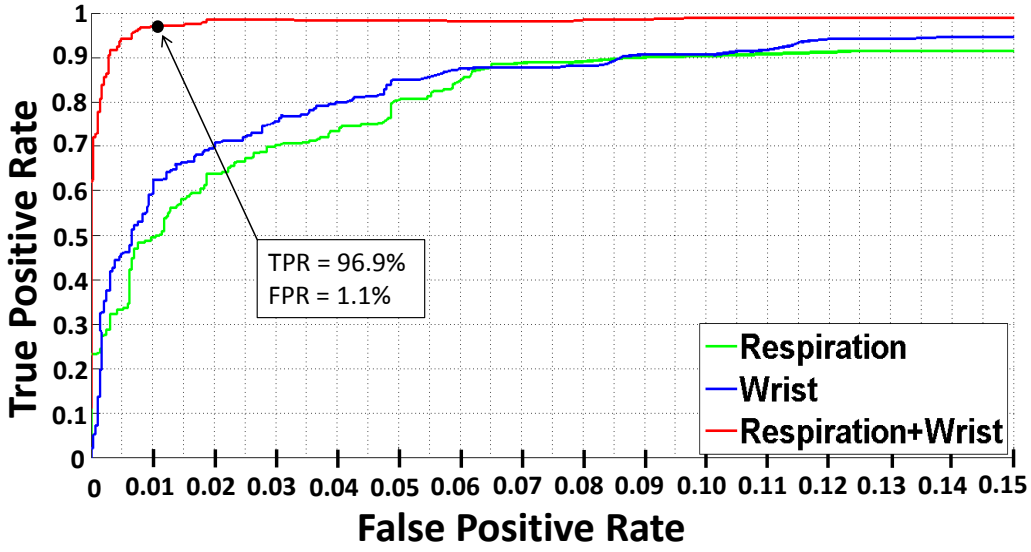


Fig. 2.12: True Positive (recall) Rate vs. False Positive Rate of three classifiers for respiration-only model, wrist-only model, and for the combined model.

Table 2.1: Confusion Matrix for training data using 10-fold cross validation; Recall=96.9%, Precision=87.5%, Accuracy=98.7%, False Positive Rate=1.1%, Kappa=0.91

	Classified as puffs	Classified as non-puffs
puffs	282	9
non-puffs	40	3505

2.5.1 Performance on Training Data

In addition to the *puffMarker* model that uses both respiration and wrist sensors, we also construct a wrist-only model to understand the performance expected if only wrist sensors are used due to its greater convenience and ease of wearing. To understand the improvement in accuracy due to each of the two sensor types, we also analyze the performance if only respiration sensor were used.

Training data for the puff classifier consists of 291 puffs and 44,696 respiration cycles that reduces to 3,545 non-puff cycles after applying our preprocessing steps. We build three different classifiers based on i) respiration-only features, ii) wrist-only features, and iii) features from both sensors.

Figure 2.12 presents the true positive rate of puffs detected versus true positive rate for the classifiers in 10-fold cross-validation. We observe that the performance of wrist-only model is better than that of respiration only. Combining both sensors results into significant improvement and makes it suitable for robust performance in the field setting. We pick an operating point on the ROC curve that is closest to the top left corner. It corresponds to a recall rate of 96.9% and false positive rate of 1.1%. A confusion matrix with various metrics for this operating point is presented in Table 2.1.

We further investigate the generalizability of our combined model by performing leave-one-subject-out cross-validation. In this experiment, the model is learned from five participants' data and evaluated on the sixth. In each experiment, threshold on the score of the SVM classifier is set to a value so that it achieves at least 95% recall rate on the training data. From Figure 2.13, we observe that recall and precision are usually high with

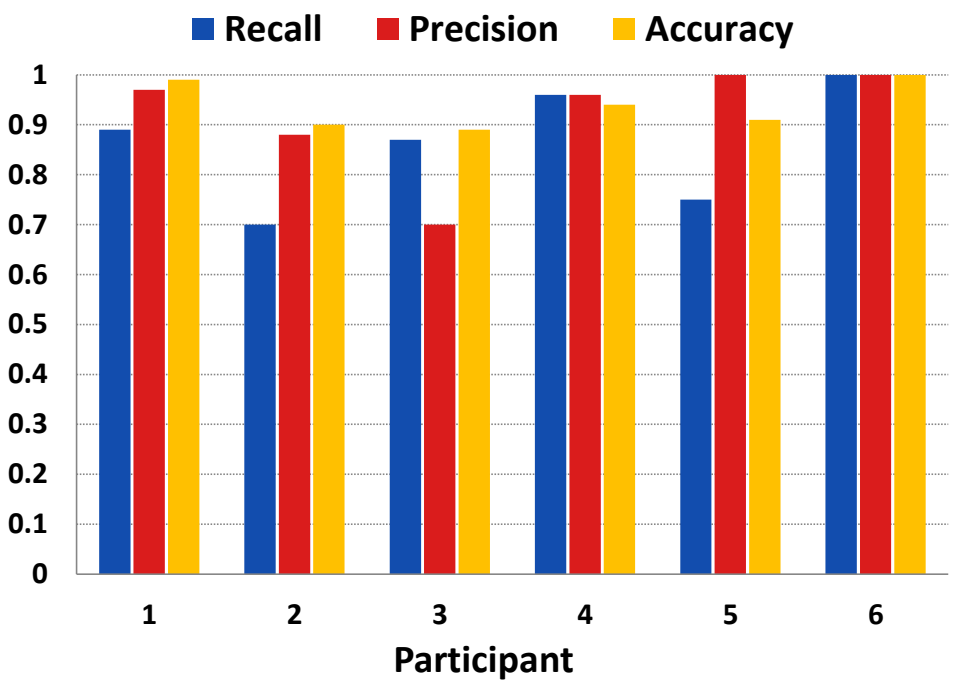


Fig. 2.13: Leave one subject out cross validation on training data.

a minimum value of 0.7. It indicates that our method can generalize to new users. The difference in recall or precision can be attributed to difference in the proportion of confounding activities (e.g., conversation and physical activity) present in each participant's data.

2.5.2 Performance on Smoking Abstinence Data

We collected data on 61 participants in a smoking cessation study in which 33 lapsed within three days (verified by a CO monitor) — 17 lapsed on the first day, 12 on the second day, and 4 on the third day. We apply our model on these data and report several findings that include recall rate for detecting first lapse, false episode per day on abstinence data, lapse progression in lapsers, puff count in first lapse episodes, and temporal inaccuracy in self-report or recall of first lapses.

Detection of First Lapses

The CO report ascertained that among the 61 participants, 33 lapsed during their post quit session; 22 of them self-reported their lapse and 11 mentioned it in their next day

Table 2.2: Comparison between *puffMarker* and Wrist-only model

	puffMarker	Wrist-only
False episode per day	$\frac{1}{6}$	$\frac{1}{1.71}$
# detected lapses	$\frac{28}{32}(87.5\%)$	$\frac{24}{32}(75\%)$

interview. One participant is excluded from analysis because of high ($> 80\%$) sensor data loss around the neighborhood of lapse self-report. Of the 32 lapsers for whom data is available, *puffMarker* detects 28 lapse episodes. We can now derive various novel results on the nature of first lapse that has previously not been known. We report them in the following.

False Episode Detection

To analyze the false positive rate of *puffMarker*, we apply it to the data collected from 28 participants who did not lapse (confirmed by CO testing) during three days of post quit session. From each of these participants, we obtained an average of 11.2 hours of data per day for 3 days, for a total of 946 hours of data. Since the participants did not smoke on these days, all episodes detected by *puffMarker* are false positives. Out of these 28 participants, we get zero false positives for 22 participants, one episode for 2 participants, one episode each on two days for 2 participants, two episodes on two days for 1 participant, and four false episodes in one day for the final participant. We get a total of 14 falsely detected episodes in 84 days, for a false episode per day of 1 every six days.

We also analyze the false episode per day on abstinent days of lapsed smokers. Since 12 participants lapsed on the second day and 4 lapsed on the third day, we have a total of 12+8=20 abstinent days on these participants. We get false positive on only one participant who lapsed on the third day. *puffMarker* detects one episode on the first post-quit day and another episode on the second day.

Performance of Wrist-only Model

In future, it may be desirable to use only wrist sensors in smoking cessation studies due to its convenience. We, therefore, also apply the wrist-only model on the

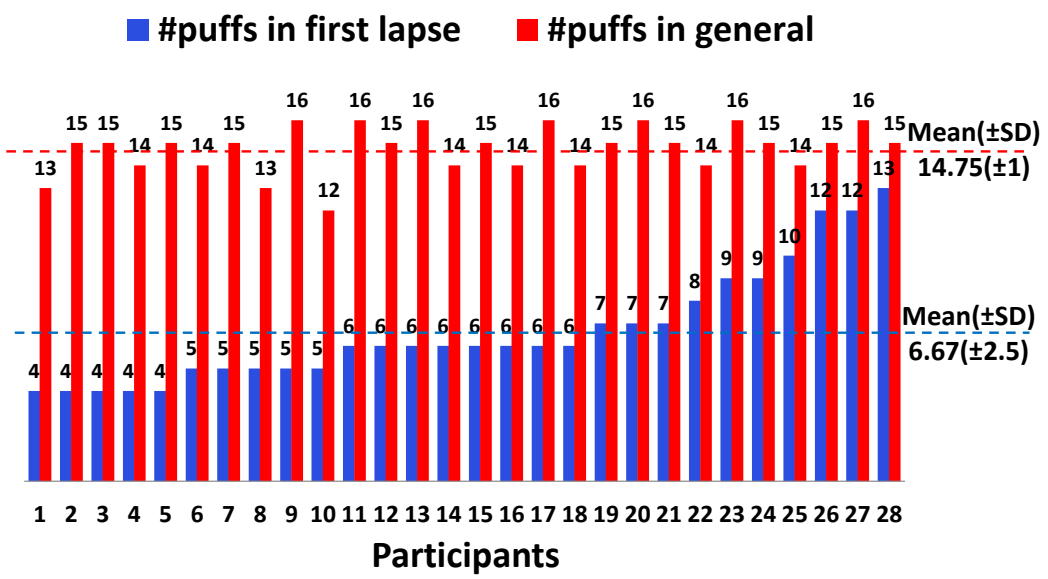


Fig. 2.14: Puff variation in Lapse episode and regular smoking episode

smoking cessation data. Table 2.2 presents the performance of wrist-sensor-only model. We observe that the wrist-only model detects 24 of the 32 lapse events and the number of false episodes detected is 49 (or 1/1.71 per day).

Characterizing the Lapse process

Now that we have a model to detect the first lapse and subsequent lapses in a smoking cessation study, we can get some new insights into the lapse process that were not observable earlier. We report three novel findings. First, we analyze the number of puffs taken during the first lapse smoking event (see Figure 2.14). We find that the number of puffs in the first lapse event after quitting is significantly lower than that during regular smoking episodes. The average number of puffs in first lapse events is 6.67 (± 2.5), whereas the average number of puffs in regular smoking episodes is 14.75 (± 1). This has been suspected by smoking researchers as smokers are trying to resist smoking in the post-quit period, but our data now provides the first objective evidence.

Second, we analyze the number of smoking episodes per day on the lapse day, the day after lapse day, and 2 days after the lapse day. Third, we analyze the number of puffs per episode on these three days. Figure 2.15 shows data for both of these metrics. Lapsers

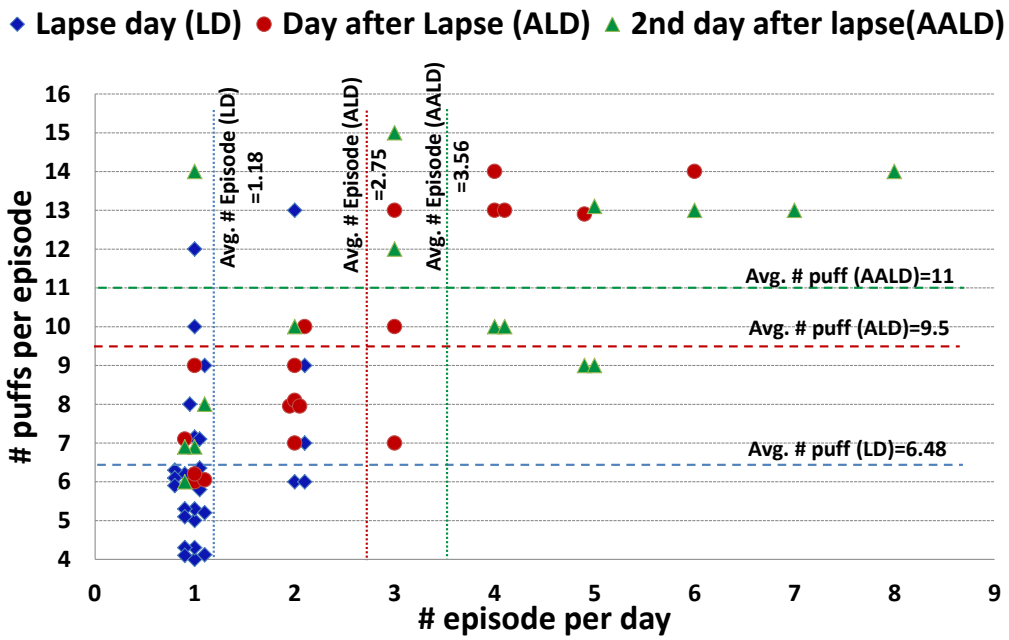


Fig. 2.15: Progression of the lapse process. Number of smoking episodes per day and number of puffs per episode following a lapse are shown for the lapse day, day after lapse, and 2 days after lapse.

smoke an average of 1.18 times on the lapse day, an average of 2.75 times on the day after lapse day, and an average of 3.56 times on the second day after lapse day. For the number of puffs per smoking episode, we observe that the number of puffs per smoking episode is 6.5 on the lapse day, 9.5 on the day after lapse day, and 11 on 2 days after the lapse day.

The above objective data support prior observations (based on self-reports) that once a participant lapses, they gradually increase smoking frequency and eventually relapse fully. It is interesting to observe that not only the smoking frequency increases every day after lapse, but the number of puffs per episode also increases. Consequently, the total number of puffs per day increases rapidly; it is 7.7 on lapse day, 26.1 on day after lapse, and 39.2 on 2 days after lapse. This study only observed abstinent smokers for only 3 post-quit days; longer studies in future may reveal the entire progression process.

Temporal Precision in Self-report or Recall of First Lapse

Accurately locating the timing of the first lapse has been considered critical for the development of interventions. Smoking researchers have suspected that lapsers do not

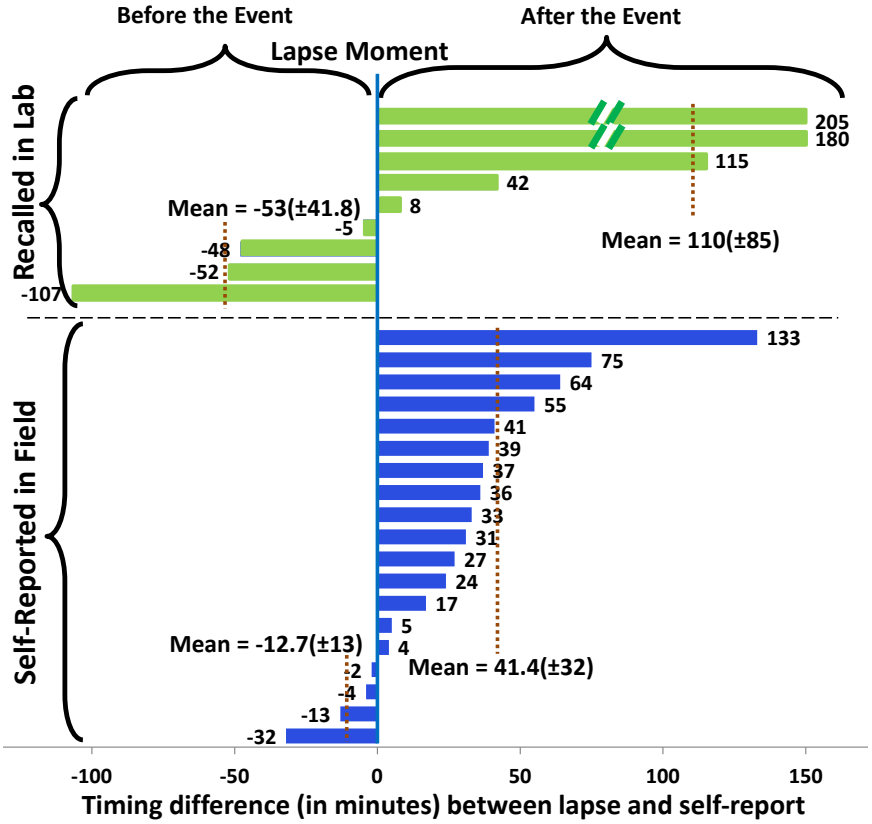


Fig. 2.16: Temporal inaccuracy of self-report

report their first lapse event promptly, partly due to being overwhelmed at the moment of this first failure in their cessation attempt. Now that we can pinpoint the timing of first lapse for 28 lapsers at the granularity of a respiration cycle (i.e., second-level accuracy), we can analyze the temporal imprecision in self-report or recall of the first lapse episodes (see Figure 2.16).

Of the 28, 9 did not self-report, 15 reported after the lapse, and 4 reported before they lapsed. The average delay for post-report was 41.4 minutes and that for a pre-report was 12.7 minutes. The temporal inaccuracy was even greater for those 9 who recalled the timing of lapse event next day in the lab. This is the first time that temporal inaccuracy in self-report of a smoking lapse has been reported.

2.6 Conclusion, Limitations, and Future Work

This is the first work to show that timing of first lapse in smoking cessation can indeed be detected using wearable sensors in real-life environment. It is also the first to show the temporal inaccuracy in self-report or recall of first smoking lapse. From a computational modeling perspective, it presents an explainable model for gesture recognition from 6-axis inertial sensors worn on wrist. It also presents a reusable approach for combining the inertial sensor data with respiration data for better detection accuracy by leveraging the diversity of these two sensor types.

But, this work has several limitations that present numerous opportunities for future works. First, the model itself can be improved in multiple ways. For example, we use a simple rule for episode construction. More sophisticated models can potentially improve the detection accuracy for smoking episodes. Personalized models that use pre-quit data of each participant to calibrate the model may provide an even better accuracy. Second, detection of other related behaviors (e.g., eating, drinking, brushing, driving) using our modeling approach from wrist-mounted inertial sensors and respiration sensor is an interesting opportunity for future work. Third, we use wrist sensors on both wrists as the smoking activity can be performed with either hand. The same is true for

other activities such as eating, typing, etc. But, wearing wrist sensors on both wrists may not be as prevalent outside of scientific studies. Obtaining similar accuracy of detection with only one sensor worn on the dominant (or non-dominant) hand is another interesting future work opportunity.

Fourth, our model was developed using data collected for cigarette smoking and hence may not directly work for cigars, e-cigarettes, hookah, etc. Fifth, since our model filters out isolated puffs and does not consider puffs to constitute a smoking episode unless there are 4 puffs in close vicinity. Hence, it may not work for detecting first lapses that consists of 3 or fewer puffs. Sixth, it also may not work when several people share a cigarette. In such a case, the time between successive puffs becomes longer than usual. Seventh, replication of our method in other populations can further improve its validity and utility. Finally, our work opens up a very rich area of research for discovering efficacious just-in-time interventions that can be triggered from predictors detected by sensors such as GPS, smart eyeglasses, electronic and social media, and physiological sensors.

Chapter 3

mSieve: Protecting Behavioral Privacy in Sharing of Time Series of Mobile Sensor Data

As discussed in the previous chapter, sharing mobile sensor data, especially physiological data, raises different privacy challenges, protecting private behaviors that can be revealed from time series of sensor data. Existing privacy mechanisms rely on noise addition and data perturbation. But the accuracy requirement on inferences drawn from physiological data, together with well-established limits within which these data values occur, render traditional privacy mechanisms inapplicable. In this chapter, we define a new behavioral privacy metric and propose a novel data substitution mechanism to protect behavioral privacy.

3.1 Introduction

Smart phones with their onboard sensors and their ability to interface with a wide variety of external body worn sensors, provide an appealing mobile health (mHealth) platform that can be leveraged for continuous and unobtrusive monitoring of an individual in their daily life. The collected data can be shared with health care providers who can use the data to better understand the influence of the environment on an individual and be proactive with their prognosis. On one hand, mHealth platforms have the potential to usher in affordable healthcare, but, on the other hand, their ability to continuously collect data about an individual raises serious privacy concerns and limits their adoption – concerns that are largely absent during traditional episodic treatments.

Motivating example: Consider a scientific study being conducted to assess the daily activities (e.g., sedentary versus active life styles) and behaviors of a user. To this end, the user participating in the study shares data from several body-worn sensors (e.g., respiration (RIP), electrocardiogram (ECG) and accelerometer sensors) with the study investigators. The data collected can be used to infer physical activities such as *walking, running, stationary* (from accelerometers), but also correlate these activities with

behaviors such as *conversation* episodes, *stress* episodes, detect when the user is *eating* or *drinking water/coffee*, and if the user *smokes* or takes cocaine. Note, activity can be detected from accelerometer data [43], conversation episodes [44] from respiration data, onset of stress [45, 46] can be inferred from ECG data, *eating* from wrist-worn sensors [47], smoking from respiration and wrist-worn sensors [2, 3, 48] and cocaine use from ECG data [49].

On one hand, some of the above inferences such as *walking*, *conversation*, *eating* are extremely useful in investigation of behavioral risk factors on health and wellness. But, on the other hand, inferences such as *smoking*, *cocaine use* and *stress* may be sensitive to the user and needs to be kept private. Thus, we have a conundrum, where the same time series data can be used for making both utility providing inferences (that are desirable) and also sensitive inferences (that need to be protected).

Challenges unique to physiological data: While there exists a large body of prior work on data privacy, there are several challenges that are unique to maintaining privacy of inferences drawn from time series of physiological data. First, the inferences themselves (e.g., detecting variation in heart rate, respiratory disorders) are extremely critical to proper diagnosis and incorrect inferences can severely affect and even threaten human life. Second, there are well-defined limits for various physiological signals (e.g., the interbeat interval in ECG is typically between 300ms and $2,000$ms [50], and so on) and non-conformance to those thresholds can render the data unusable. Third, there is high degree of correlation between an observed human behavior and the data recorded by these physiological sensors. For example, physical activities such as walking or running are associated with higher heart and respiration rates. Finally, physiological signals are high-dimensional, are extremely rich in information, and when continuously collected embed minute elements of an individual's lifestyle patterns. These patterns or inferences are often correlated making it difficult to protect the privacy of one inference in isolation of the others.

These above challenges place constraints on the mechanisms that can be used to protect privacy of physiological data. The constraints are in terms of the magnitude of noise that can be added while retaining the utility of the inferences and in handling of the correlation between the data streams from the various sensors. Anonymization techniques such as k-anonymity [51], l-diversity [52], and t-closeness [53] propose data obfuscation aimed towards protecting the identity of a user within a subpopulation. However, we consider a setting where a single-user shares data with (possibly) many recipients (primary/secondary researchers), and the identity of the user is already known to the data recipients.

A principled mechanism for preserving privacy during analysis is differential privacy [54]. While several variants of differential privacy have been proposed [55–57], the central idea there is to adequately obfuscate a query response computed on a multi-user statistical database (by adding noise typically drawn from a Laplace distribution) such that the presence or absence of any user in the database is protected. However, this notion of differential privacy cannot be directly applied to our single user setting to protect behavioral privacy. Recent model-based approaches for location privacy such as [58, 59] focus on effective data suppression to protect sensitive inferences, but these can't be applied directly to protect behavioral privacy from mobile sensor data either due to unique challenges listed above.

Our approach: In this paper, we propose *mSieve*, a model-based data substitution approach to address the privacy challenges arising from sharing of personal physiological data. We group the inferences that can be drawn from shared data into two sets – a *whitelist* and a *blacklist* [60, 61]. Inferences that are desirable for the user, such as tracking activity, conversation episodes, frequency of eating, are all utility providing to the user and are part of a whitelist. Other inferences such as smoking and onset of stress are sensitive to the user and need to be kept private. These inferences form part of the

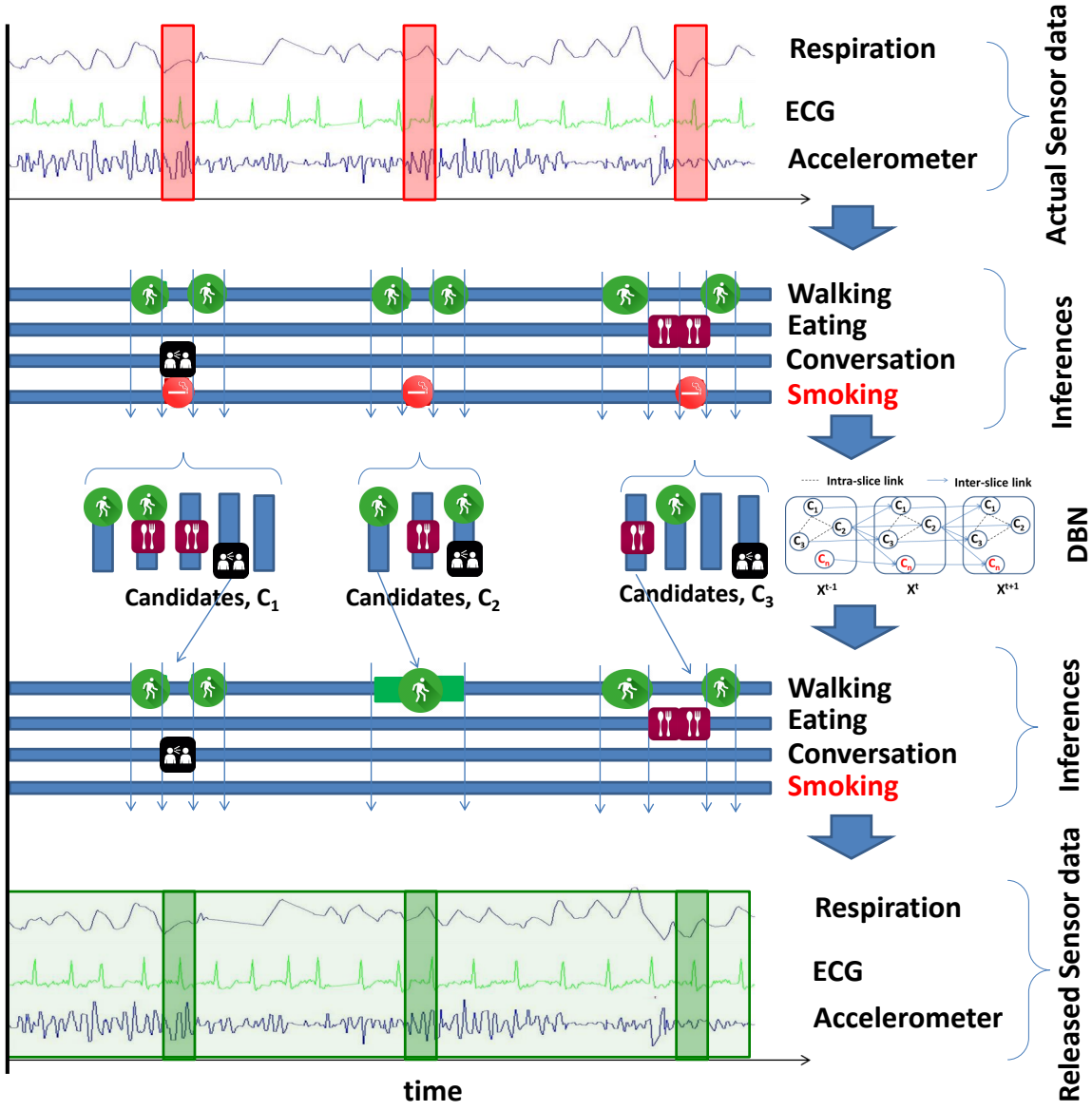


Fig. 3.1: Illustration of the *mSieve* process.

blacklist. Our goal is to prevent an adversary from making any of the inferences in the user-specified blacklist while being able to *accurately* compute the whitelisted inferences.

Figure 3.1 illustrates the flow of data and the various components of *mSieve*. In summary, given various streams of sensor data from a user, *mSieve* identifies sensitive data segments and substitutes them with the *most-plausible* non-sensitive data segments. To do so, it computes a Dynamic Bayesian Network (DBN) model over the user's data. The model maintains a distribution over the various behavioral states (e.g., smoking, running, conversation etc.) of the user. Note, these states are computed using the data collected from the body-worn sensors. To perform substitution, segments of data that reveal sensitive behavior are detected and removed. The DBN model is then used to identify candidate replacement segments from the same user's data that can be used in place of the deleted segments. We use several techniques (such as dynamic programming approach, and a greedy approach based best fit algorithm) to select the best segments that preserve privacy and simultaneously retain the overall statistics of the physiological signal (providing utility). To assess the privacy guarantees of our scheme, inspired by the privacy definition of differential privacy, we define the notion of *differential behavioral privacy* to protect sensitive inferences. The metric ensures that the information leaked about a sensitive inference from a substituted segment is always bounded.

We evaluate the efficacy of our substitution scheme using 660 hours of ECG, respiration, location and accelerometer data collected over multiple user studies with over 43 participants. We demonstrate that sensitive behavioral inferences, contributing to privacy loss, such as onset of stress, smoking, and cocaine use can be protected while still retaining meaningful utility ($\geq 85\%$ accuracy when privacy sensitivity is high and $\geq 90\%$ on average) of the shared physiological signals in terms of its use for tracking heart rate, breathing irregularities, and detecting conversation episodes.

Table 3.1: Summary of notations used.

User model, DBN	D_u
Raw sensor data	$\vec{r_i} = \{r_i(t) \| t_{start} \leq t \leq t_{end}\}$
Actual sensor data	$\vec{\mathbf{r}} = \{\vec{r_1}, \vec{r_2}, ..., \vec{r}_{n_s}\}$
Released sensor data	$\vec{\hat{\mathbf{r}}}(t) = \{\vec{\hat{r}}_1, \vec{\hat{r}}_2, ..., \vec{\hat{r}}_{n_s}\}$
Duration difference	$d_i = t_i - \hat{t}_i \; \forall 1 \leq i \leq n$
State	$x = (x_1, ..., x_n) \in \{0, 1\}^n$
State interval	$s = (x, t_s, t_e)$
Actual state sequence	$\vec{\mathbf{s}} =< s_1, s_2, ..., s_\tau >$
Obfuscated state sequence	$\vec{\hat{\mathbf{s}}} =< \hat{s}_1, \hat{s}_2, ..., \hat{s}_\tau >$
Set of sensitive states	$\mathbb{S}_B$
Set of safe states	$\mathbb{S}_W$
Set of all states	$\mathbb{S} = \mathbb{S}_W \cup \mathbb{S}_B$
Hole	h_k
Candidate states for k^{th} hole	$\mathbb{C}_k$

3.2 Definitions and Problem Statement

We first introduce notations and define terms we use throughout the paper and also formalize the problem statement.

Sensor Data: Let $r_i(t)$ denote the sensor data from the ith sensor at time t, where $i = 1, \ldots, n_s$. We define $\vec{r_i} = \{r_i(t) | t_s \leq t \leq t_e\}$ as the time-series of measurements from the ith sensor, from starting time t_s to ending time t_e. Finally, $\mathbf{r}(t)$ denotes the collection of time-series data from all the different sensors, i.e., $\mathbf{r}(t) = \{r_1(t), r_2(t), ..., r_{n_s}(t)\}$.

Inferences: An inference is a function computed (e.g., using a machine learning model) over a window of data values. Time-series data from different sensors (such as ECG, respiration, accelerometer) are used for computing inferences using data buffered over a chosen time interval. Let $x_i(t)$ be inference value of the ith inference at time t. We assume that all our inferences are binary classifiers, which output true when the inference occurs within the time interval and false otherwise, i.e.,[1] $x_i(t) \in \{0, 1\}$. We define $\vec{x_i} = \{x_i(t) | t_s \leq t \leq t_e\}$ to be the time-series for the ith inference within the time interval

[1] Any inference that produces categories can be easily converted to a set of binary inferences, one for each category of output.

(t_s, t_e). Again, $\mathbf{x}(t) = \{x_1(t), x_2(t), ..., x_n(t)\}$ represents the collection of all possible inference time-series.

Whitelist and Blacklist of inferences: As mentioned earlier, a key component of our privacy mechanism is the separation of the possible inferences into a Whitelist (denoted by W) and a Blacklist (denoted by B). In *mSieve*, a whitelist is a set of inferences that are essential for obtaining utility from the shared data, and the goal of the recipient is to accurately compute the distribution $p(x_i)$, where $x_i \in W$. Similarly, the blacklist B, is a list of inferences x_i, whose release the user would like to protect from the recipient.

State: A bit vector of length n is used to represent a user state $x = (x_1, x_2, ..., x_n) \in \{0, 1\}^n$. The ith element of the bit vector represents the value of the ith inference. Without loss of generality, we assign the first n_w bit values of state x to the whitelist, i.e., x_i for $1 \leq i \leq n_w$ and the remaining n_b bit values to the blacklist. We assume that whitelist and blacklist forms a disjoint partition of the inference set, i.e., $n = (n_w + n_b)$. A state is sensitive, if one or more bits corresponding to inferences in set B, are set to one. All sensitive states are included in set $\mathbb{S}_B$ and the non-sensitive states are in set $\mathbb{S}_W$.

State Interval: We define a state interval, $s = (x, t_s, t_e)$ as a state x at which the user dwells during an interval (t_s, t_e). Successive state intervals are indexed by s^j, where $j = 1, \ldots, \tau$. The value of each inference stays the same during an interval. Interval changes to the next one when any of the inference values change. Unless otherwise specified, we use the shorthand notation s_i^j for $s^j.x_i$.

State Sequence: We define a state sequence, $\mathbf{s} = (s^1, s^2, ..., s^\tau)$, where $s^j.x \neq s^{j+1}.x$ and $s^j.t_e == s^{j+1}.t_s$ for all $j = 1, \ldots, \tau - 1$.

3.2.1 User Model

Transition among different user states can be modeled using graphical models such as Markov Chain (MC), Hidden Markov Models (HMM), and Dynamic Bayesian Network (DBN). The Markov models, while suitable for modeling the temporal correlation among

the states across time intervals, do not capture their conditional independence within a particular time interval. Therefore, we model the transition between states as a DBN, D_u. In each time slice of the DBN, we maintain a uniform Bayesian Network described below:

- *Nodes:* Each node of the DBN is a random variable S^j representing a user state at time interval j. Denoting the ith inference within the state as S_i^j, we can write $S^j = \{S_1^j, S_2^j..., S_n^j\}$.

In addition to nodes, a DBN also contains two types of edges:

- *Intra-slice links:* For any time slice j, conditional independence between individual inferences X_1 to X_n is maintained as a Bayesian Network (BN). Denoting the parents of node S^j by $Pa(S^j)$ we have:

$$P(S_i^j \mid Pa(S_i^j)); \text{where } Pa(S_i^j) = \{S_k^j : P(S_i^j \mid S_k^j)\}$$

- *Inter-slice links:* A DBN not only models conditional independence among states within a time slice but also captures their temporal correlations across time slices. These transition probabilities among nodes in different BNs are represented by the inter slice links. We use a first order model, so these links are only between adjacent time slices.

$$P(S_i^j \mid S_i^1, ..., S_i^{j-1}) = P(S_i^j \mid S_i^{j-1});$$

These conditional probabilities are stored in a *Conditional Probability table (CPT)*, which is associated with each node S^j. An illustration of a DBN representing temporal behavior among states is presented in Figure 3.2.

3.2.2 Adversary Model

We use a DBN to capture adversarial knowledge. A DBN is a powerful graphical model that can effectively encode both the temporal and spatial correlation among

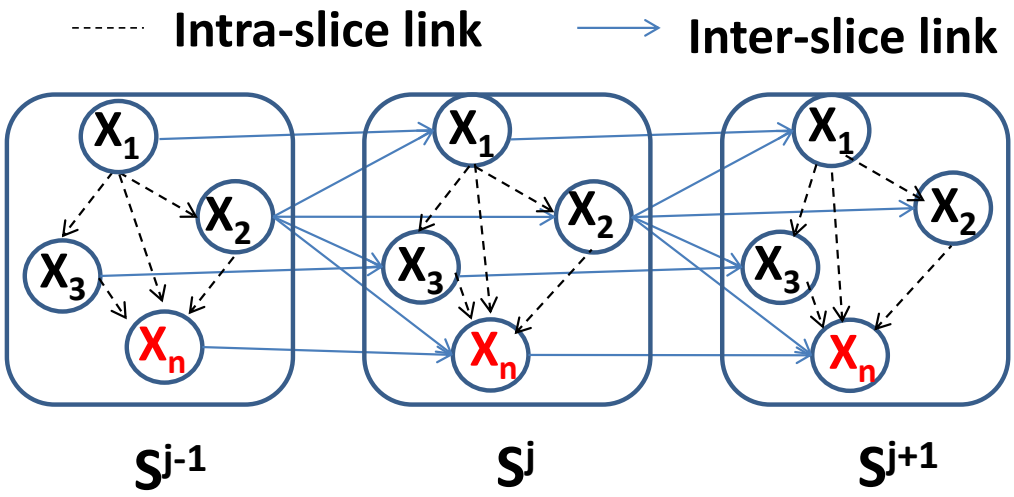


Fig. 3.2: DBN showing user states over different time slices.

inferences. It is also a generalization of Markov models (including the HMM) that are typically used to encode these information. We consider two types of adversarial attacks.

- *Data-based attack:* In this setting, the attacker has access to the raw sensor data.
- *Model-based attack:* The attacker has access to released inferences computed over raw data but not the raw data.

In addition, we assume that in both settings the attacker is aware of the blacklist inferences B, and the *mSieve* algorithm is publicly known.

The algorithms in *mSieve*, are designed under an assumption that an adversary uses a DBN or a less powerful model for capturing the correlation among the user states. However, if an adversary uses a model that is more expressive in terms of modeling state correlations, or has access to side channel information that is not contained in the user model then additional leakage may occur from the released data.

3.2.3 Privacy

The privacy guarantee we seek is such that an adversary with access to data released by *mSieve* should not be able to suspect a sensitive behavior in a released data

with significantly *higher* likelihood than when suspecting the same behavior in a corresponding reference data.

Corresponding Reference Data: For a given sensor time series $\vec{\mathbf{r}}$, a corresponding reference data $\vec{\bar{\mathbf{r}}}$ is such that it releases no more information about the blacklisted inferences than a null time-series, but is otherwise maximally close to $\vec{\mathbf{r}}$.

Differential Behavioral Privacy: A system Λ preserves ϵ-privacy, if for any input sensor time series $\vec{\mathbf{r}}$ with start time t_s and end time t_e, it produces an output $\vec{\hat{\mathbf{r}}}$ with same t_s and t_e such that for any query $q(.;.)$ on $\vec{\hat{\mathbf{r}}}$ about any sensitive state $b \in \mathbb{S}_B$ and for all $K \in Range(q)$, and the same query $q(.;.)$ on any $\vec{\bar{\mathbf{r}}}$ with the same t_s and t_e, the output is bounded by e^{ϵ}.

$$D(\vec{\hat{\mathbf{r}}}||\vec{\bar{\mathbf{r}}}) = \frac{P(q(\vec{\hat{\mathbf{r}}}; b) \in K)}{P(q(\vec{\bar{\mathbf{r}}}; b) \in K)} \leq e^{\epsilon} \tag{3.1}$$

The parameter ϵ denotes privacy sensitivity. A low value of ϵ implies a high privacy level and vice-versa.

3.2.4 Utility

We define utility over an entire released episode. Released data should preserve the same white list of inferences as the original data. Utility metric minimizes distribution difference between the original and released data.

Let p_i be the probability of inference x_i occurring in the actual signal. For simplicity, suppose Inference x_i occurs for $t_i = \sum_{j=1}^{\tau} I_{x_j^i=1} * (s^j.t_e - s^j.t_s)$ duration out of a total of $T = \sum_{j=1}^{\tau}(s^j.t_e - s^j.t_s)$ time units in the original data, where I is the identity function. Then $p_i = \frac{t_i}{T}$.

Utility Loss: Let $P = (p_1, p_2, ..., p_{n_w})$ be the probability vector of the white listed inferences in the actual data and $\hat{P} = (\hat{p_1}, \hat{p_2}, ..., \hat{p_{n_w}})$ be the probability vector of the white listed inferences in the released data, where $\hat{p_i} = \frac{\hat{t_i}}{T}$ and $\hat{t_i}$ is the duration of inference x_i in the released data. Then, we define our utility loss metric as,

$$U_{loss} = \|P - \hat{P}\| = \sum_{i=1}^{n_w} |p_i - \hat{p_i}| = \frac{1}{T}\sum_{i=1}^{n_w} |t_i - \hat{t_i}| \tag{3.2}$$

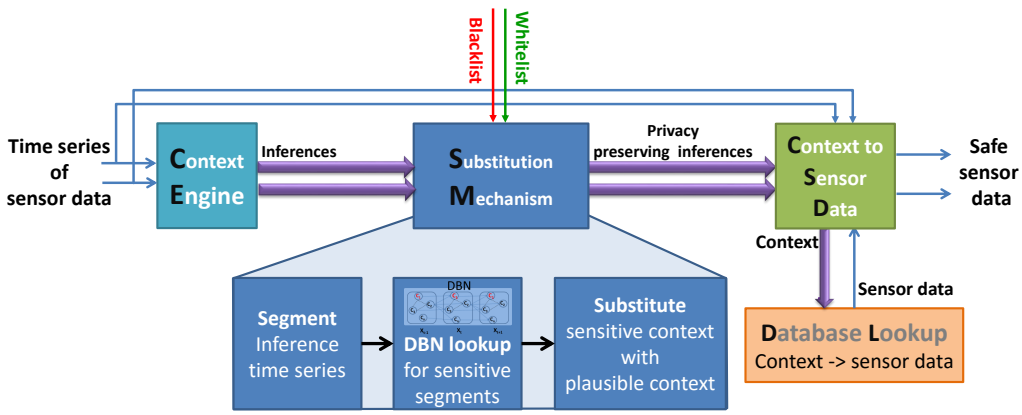


Fig. 3.3: An overview of the *mSieve* framework.

3.2.5 Problem Definition

The goal of *mSieve* is to obfuscate time series data $\vec{\mathbf{r}} \in DB$ and generate $\vec{\hat{\mathbf{r}}}$ with the same start and end time such that it satisfies the privacy constrains in Equation (3.1) and minimizes the utility loss in Equation (3.2).

Problem 1. *For any given tolerable privacy loss $\epsilon > 0$, the utility-plausibility tradeoff can be formulated as the following optimization problem:*

$$\begin{aligned} &\textit{minimize} && U_{loss} \\ &\textit{s.t.} && D(\vec{\hat{\boldsymbol{r}}}||\vec{\boldsymbol{r}}) \leq e^{\epsilon} \end{aligned}$$

3.3 System Overview

We now present an overview of the components, as shown in Figure 3.3, that are required to implement the end-to-end system from sensor data, to user states, to privacy preserving safe states, and finally to the release of sensor data.

3.3.1 Context Engine

The Context Engine (CE) generates inferences from raw sensor data. Inferred signals are more suitable for data modeling than unprocessed, raw sensor signals as it simplifies the modeling task. CE transforms the raw sensor signals to task-specific feature signals. For example, RR-interval is more suitable to infer heart rate than raw ECG signal.

CE takes raw signals as input and produces time series of inference values as output. All the white list and black list inferences are inferred by the CE.

3.3.2 Substitution Mechanism

We begin by *segmenting* the time series of inferences. Let τ be the total number of segments, where each segment is represented by state interval s^j where $1 \leq j \leq \tau$. As mentioned earlier, we assume that inferences are represented by a single bit and thus at a particular time interval the CE provides as output a bit vector of size n, which also constitutes the user state $s^j.x \in \{0, 1\}^n$. Finally, the segmentation generates a sequence of such state intervals, $\vec{s} =< s^1, s^2, ..., s^\tau >$.

Some of the states in $\vec{s}$ can be sensitive to the user. The first step in our substitution algorithm is to remove the sensitive states and also all other states that might lead to the sensitive states. This introduces discontinuity in that state sequence; we call these gaps as *holes*. The next step is to perform a *DBN lookup* to identify plausible candidates to fill all the holes. This lookup operation on user DBN, D_u, provides a set of candidate states for filling holes in any time interval j. We note that the plausible states are checked for continuation with the previous state. To select the best state from among the candidate states, we consider utility loss as our metric. We **substitute** a hole with a state or sequence of states that minimizes utility loss.

3.3.3 Context to Sensor Data

Another important consideration in the substitution process is to maintain signal continuity. The *Context-to-Sensor-Data* module accepts possible substitution candidates generated by the substitution mechanism and produces safe, privacy-preserving sensor data as output. This module communicates with the database to ensure that the released sensor data is safe, i.e., meets the privacy and utility criteria.

3.4 Solution Details

In this section, we discuss our proposed solution to mitigate privacy risks. Algorithm 1 takes raw sensor data $\vec{r}$, sensitive states $\mathbb{S}_B$, and user DBN D_u, and outputs raw data $\vec{\hat{r}}$ that does not contain any signature about sensitive inferences.

Algorithm 1 mSieve: Plausible Substitution Mechanism

Input: $\vec{r}, \mathbb{S}_B, D_u, \epsilon$
$\vec{s}$ = *getStateSequence*($\vec{r}$)
$k \leftarrow 1$
$\vec{\hat{s}} \leftarrow \vec{s}$
for each interval $j \in \{1, 2, ..., \tau\}$ **do**
    if isHole($s^j, D_u, \mathbb{S}_B, \epsilon$) is *true* **then**
        $\vec{\hat{s}}^j = h_k = (\emptyset, s^j.t_s, s^j.t_e)$
        $\mathbb{C}_k$ = *getPlausibleCandidateSet*($D_u, \vec{s}, h_k, \epsilon$)
        $k \leftarrow k + 1$
    end if
end for
Selected candidate $\{c_1, ..., c_k\}$ = *FillHole*($\{h_k\}, \{\mathbb{C}_k\}$)
$\vec{\hat{r}} \leftarrow \vec{r}$
for each hole $h_k = (., t_s, t_e)$, selected candidate c_k **do**
    $\hat{r}(t_s, t_e)$ = *getSensorDataFromDB*($c_k, t_e - t_s$)
end for
return $\vec{\hat{r}}$

3.4.1 Step 1: Sensor Data to State Sequence

We first convert the time-series of raw data samples from various sensors, $\vec{\mathbf{r}}$, to time-series of inferences using (classifier) models. The classifier models are specific to an inference and detects the time interval over which the inference occurs. Let $e_i = \{(t_s, t_e)\}$ denote the set of time intervals over which the ith inference is detected by a model where $1 \leq i \leq n = (n_w + n_b)$ (recall n_w and n_b are the number of whitelisted and blacklisted inferences respectively). We create the inference time series $x_i(t)$ for the ith inference as

$$x_i(t) = \begin{cases} 1 & \text{if } t_s \leq t \leq t_e \text{ for any } (t_s, t_e) \in e_i \\ 0 & \text{Otherwise.} \end{cases}$$

Collectively, we define, $x(t) = (x_1(t), x_2(t), \ldots, x_n(t))$ as the n-dimensional *state* at time t.

Let $\mathbf{t} = \cup_{i=1}^{n} \cup_{(t_s,t_e)\in e_i} \{t_s, t_e\}$ be the set of all time points (whether start or end) corresponding to the inference occurrences, arranged in ascending order and $\tau = |\mathbf{t}| - 1$. Note, a user stays in same state between time interval (t_i, t_{i+1}) for $i = 1, 2, ..., \tau$ and $t_i \in \mathbf{t}$. We denote by $s^i = (x(t), t_i, t_{i+1})$, where $t_i < t < t_{i+1}$, the ith state interval. It is clear that two consecutive states are different. It forms a state sequence $\vec{s} = \ < s^1, s^2, \ldots, s^\tau >$.

3.4.2 Step 2: Locate and Delete Sensitive and Unsafe States

We consider three types of states. a) Sensitive state, b) Unsafe state, and c) Safe state. We define a state s^j as **sensitive state** if any of the last n_b bits are set to one, i.e., if $\vee_{i=n_w+1}^{n} s_i^j = 1$, where $\vee$ is logical *or* operation (recall $\mathbb{S}_B$ is the set of sensitive states).

We define a state s^j as **unsafe state** if it is not directly sensitive but may contain some information about blacklist, e.g., act of walking outside of a building to smoke and returning back after smoking. We use the following local privacy check, which is similar as [59, 62], to mark a state as unsafe

$$\frac{P(S^{j+1} \in \mathbb{S}_B | S^j = s^j)}{P(S^{j+1} \in \mathbb{S}_B)} \leq \epsilon^\delta. \tag{3.3}$$

This condition provides us with a mechanism to stem privacy leakage from unsafe states. Here, δ is our local privacy sensitivity. Lemma 1 below describes how to select δ.

Lemma 1. *For any given $\epsilon > 0$ there exist a $\delta < \epsilon/2\tau$ such that if*

$$\frac{P(S^i \in \mathbb{S}_B | S^{i-1} = s^{i-1})}{P(S^i \in \mathbb{S}_B)} < e^\delta$$

then $D(\vec{\hat{r}} || \vec{\bar{r}}) \leq e^\epsilon$

A proof appears in the Appendix.

Deletion of sensitive and unsafe states in $\vec{s}$ results into $\vec{\bar{s}}$ that is punctuated with holes. Each hole consists of a starting time t_s, and an end time t_e. Thus, the kth hole is defined as $h_k = (t_s, t_e)$. We denote the state occurring immediately before a hole h_k as $pre(h_k)$, and the state after h_k as $next(h_k)$.

3.4.3 Step 3: Candidate Generation

A candidate (i.e., a state sequence) is a sequence of states that can be substituted in place of a hole. A candidate should be such that it does not contain any sensitive or unsafe state and it maintains continuity, i.e., transitions among the states in the filled up time series should be plausible using similar criteria as in (3.3). If there does not exist a candidate long enough to fill the hole by itself, multiple state segments can be composed together to obtain the desired length. We use a recursive function, described in Algorithm 2, to generate candidates for each hole h_k in time interval given by (t_s, t_e). For k^{th} hole, the `GenerateCandidate` function generates all the possible candidates $cand$ for the duration given by the interval length of the hole, i.e., $dur = h_k.t_e - h_k.t_s$, and stores the candidates in the set $\mathbb{C}_k$. Prior to invoking the `GenerateCandidate` function, we initialize set $\mathbb{C}_k = \emptyset$ and current candidate $cand =< . >$ to an empty vector. Note, `isConnect`(s^j, s^{j+1}) returns true iff Equation (3.3) returns true.

Algorithm 2 GenerateCandidate

Input: $cand =< c_1, \ldots, c_l >, \mathbb{C}_k, h_k, remDur$
if $dur == 0$ and $isConnect(c_l, next(h_k))$ **then**
    $\mathbb{C}_k = \mathbb{C}_k \cup cand$
end if
if $dur > 0$ **then**
    for each $s^j \in \mathbb{S}$ **do**
        if $isConnect(c_l, s^j)$ **then**
            $remDur' = remDur - (s^j.t_e - s^j.t_s)$
            $cand' =< c_1, \ldots, c_l, s^j >$
            $GenerateCandidate(cand', \mathbb{C}_k, h_k, remDur')$
        end if
    end for
end if

If `GenerateCandidate` does not generate any candidate state sequence, i.e., $\mathbb{C}_k = \emptyset$, then we delete either the previous state $pre(h_k)$ or the next state $next(h_k)$ (whichever provides a lower utility loss), to enlarge the size of the existing hole. We then invoke `GenerateCandidate` again for this newly created hole. This iterative process of increasing the size of the hole increases the chances of finding a candidate and in our

experiments we did not encounter any instance when the algorithm failed to find a suitable candidate. But, theoretically speaking, it is possible that the algorithm may not find any candidates due to the continuity constraint. Further improving this algorithm and proving its convergence is still an open question that we leave for future work.

Complexity analysis: Since length of the holes and the states are dynamic we will use expected values to analyze complexity of this step. Let l_h be the expected length of the hole, l_s be expected length of a state and y be the expected number of states reachable from any state. Here, y is the branching factor and $\ell = \lceil (l_h/l_s) \rceil$ is the expected depth of the search tree. Then, the expected time complexity is $O(\ell^y)$.

3.4.4 Step 4: Select Candidate and Fill Holes

After the hole creation and candidate generation steps, we obtain a series of holes $h_1, \ldots, h_{n_h}$ and a set of candidates $\mathbb{C}_i$ for the ith hole. Let $\vec{c} =< c_1, c_2, \ldots, c_m >$, where $1 \leq j \leq m$ is the index assigned to the candidate c_j, it is the jth candidate to be encountered as we enumerate through candidates in sets $\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_{n_h}$ in that order. We define an allocator matrix $A[i][k]$, where $A[i][k] = 1$ implies that candidate state sequence $c_k \in \mathbb{C}$ is a candidate for the ith hole, i.e. $c_k \in \mathbb{C}_i$. Let, after hole creation, $\bar{t}_j$ be the duration of the jth whitelist in $\vec{\bar{s}}$. Thus, the initial duration difference is $\bar{d}_j = \bar{t}_j - t_j$ and the initial utility loss, $U^0_{Loss} = \sum_{j=1}^{n_w} |\bar{d}_j|$. The next step is to select a candidate for each hole that minimizes the objective function specified in Equation 1. We formulate the above as a *Hole Filing Problem* stated below.

Problem 2. ***Hole filling Problem:*** *As stated earlier, for the jth whitelisted inference, t_j denotes its total duration in $\vec{r}$ and $\hat{t_j}$ its duration in $\vec{\hat{r}}$. Let $d_j = \hat{t_j} - t_j$. The goal is to fill all the holes with candidate state sequences such that*

$$\textit{Objective function: } min \sum_{i=j}^{n_w} |\hat{t_j} - t_j| = min \sum_{j=1}^{n_w} |d_j|$$

It can be shown that the above problem minimizes utility loss U_{loss} (in Equation 3.2). By reducing the bin packing problem to the unconstrained version of the hole filling problem, i.e., by setting $A[i][k] = 1$ for all i and k and the privacy sensitivity parameter ϵ, to a large number, it can be shown that the hole filling problem is *NP-hard*. Therefore, we first provide a dynamic programming based solution that gives optimal result but requires exponential memory. We then provide a greedy based solution that is not optimal but runs in polynomial time.

3.4.5 Dynamic Programming Solution

The main idea here is to compute the solutions to smaller sub-problems and store the solutions in a table, so that they can be reused repeatedly later to solve the overall problem. To do so, we need to decompose the hole filling problem in terms of smaller sub-problems and find a relation between the structure of the optimal solution for the original problem, and solution of the smaller sub-problems. We begin by defining the problem recursively as follows:

Recurrence Relation: Let L be similar to A, except that the entry $L[i][j]$ stores the minimum utility loss achieved if the kth candidate is used to fill the ith hole, where $1 \leq i \leq n_h$ and $1 \leq k \leq m$.

To solve the problem, we use a bottom-up approach. At first, we compute the optimal result for the first hole. Then, using this result we compute the optimal result for the second hole and so on. We begin with the following initialization

$$L[i][k] = \begin{cases} U^0_{Loss} & \text{if } i = 0 \\ \infty & \text{Otherwise.} \end{cases}$$

Let $\vec{d}_{i,k} = \ < d^{i,k}_1, \ldots, d^{i,k}_{n_w} >$ be the duration difference vector after assigning candidate c_k to i^{th} hole. Initialize $d^{0,k}_j = \bar{d}_j$ for all $1 \leq k \leq m$ and $1 \leq j \leq n_w$. To understand the working of the algorithm, suppose that holes h_1 through h_{i-1} have all been assigned, and

we are now ready to make an assignment for h_i, i.e., we are now in stage i. Let $dur_j(c_k)$ be the duration of the jth whitelist in candidate c_k. Then, we update $L[i][k]$ with

$$\min_{1 \le l \le m} \{L[i-1][l] + \sum_{j=1}^{n_w} -|d_j^{i-1,l}| + |d_j^{i-1,l} + dur_j(c_k)|\} \tag{3.4}$$
$$\text{for } 1 \le i \le n_h \text{ and } 1 \le k \le m$$

For the l, that results into a minimum, we update $d_j^{i,k} = d_j^{i-1,l} + dur_j(c_k)$ for $1 \le j \le n_w$. We also maintain an additional data structure $pre(i,k)$ that stores index of previous candidate from where we update $L[i][k]$, i.e. $pre(i,k) = l$. We continue this process till $i = n_h$ and $k = m$. Finally, we select c_k for last hole h_{n_h} such that $L[n_h][k]$ is minimum. Using the pre data structure, we determine complete assignment of the holes by invoking `AssignCandidate`(n_h, k) (Algorithm 3).

Algorithm 3 AssignCandidate

1: **Input:** $pre(.,.), i, k$
2: **if** $i > 0$ **then**
3: Assign c_k in i^{th} hole
4: `call` $AssignCandidate(pre, i-1, pre(i,k))$
5: **end if**

Complexity analysis: Maximum number of candidates for each hole is $O(\ell^y)$ (since any combination of whitelisted inferences can be a candidate sequence) and the maximum number of holes is $O(\tau)$. Thus, the upper bound on space required for a dynamic program based solution is $O(\tau\ell^y)$ and the upper bound on time is $O(\tau\ell^y\ell^y)$.

3.4.6 Greedy Solution

We now provide a greedy strategy for the hole filling problem. For each hole, we choose an item that minimizes the utility loss, U_{loss}, among all the candidates. We repeat this process until all the holes are filled. This process is described in Algorithm 4.

Since, in every iteration, we select an item that locally minimizes U_{loss} for that hole, this method does not always provide an optimal result. However, the space

Algorithm 4 GreedySolution

1: **Input:** $\{\mathbb{C}_i\}_{i=1}^{n_h}, \{h_i\}_{i=1}^{n_h}, \vec{\bar{d}}$
2: $\vec{d} = \vec{\bar{d}}$
3: **for** `each` $i = 1, 2, \ldots, n_h$ **do**
4: select $\min_{c_k \in \mathbb{C}_i} \sum_{j=1}^{n_w} -|d_j| + |d_j + dur_j(c_k)|$
5: $d_j = d_j + dur_j(c_k)$; for all $1 \leq j \leq n_w$
6: **end for**

complexity reduces to $O(\tau)$ and time complexity reduces to $O(\tau \ell^y)$ which is ℓ^y times smaller than the previous solution using dynamic programming.

3.4.7 Step 5: Sensor Data Substitution

Now, for each hole h_k, we have to select a segment of sensor data that corresponds to the selected candidate state $c_k \in \mathbb{C}_k$. For this, we maintain a mapping database, M, that stores sensor segment of different length for each possible state. However, for substituting the sensor data for a state within an interval, we have to maintain consistency at both boundaries of the hole. We use the case of ECG signals to illustrate feature value consistency. One can also consider morphological consistency or others relevant characteristics.

Feature value consistency

Let rr_i be the ith RR interval in an ECG data, $\vec{\hat{\mathbf{r}}}$. The RR interval is used to calculate other features of the signal defined as below.

1. Point of time error: Limit check on the current value

$$e(rr_i) = \begin{cases} 0 & \text{if } 300 < rr_i < 2000 \\ 1 & \text{Otherwise.} \end{cases}$$

2. Continuity error: Limit check on the first order difference

$$e(rr_i|rr_{i-1}) = \begin{cases} 0 & \text{if } |rr_i - rr_{i-1}| < 200 \\ 1 & \text{Otherwise.} \end{cases}$$

Let $\{rr_1, \ldots, rr_k\}$ be sequence of RR intervals calculated in an ECG data, $\vec{\hat{\mathbf{r}}}$. We define feature value error by,

$$e_f(\vec{\hat{\mathbf{r}}}) = \frac{e(rr_1) + \sum_{i=2}^{k} e(rr_i) \vee e(rr_i|rr_{i-1})}{n}$$

mSieve maintains a database of sensor data corresponding to each candidate state, and while substituting, it selects sensor data corresponding to a released state that minimizes the boundary errors on both boundaries.

3.4.8 Limits of the *mSieve* Algorithm:

We discuss three limits of *mSieve*. First, if all the inferences are part of the blacklist, then the released data will correspond to the null state $x = \emptyset$, i.e. $x_i = 0$ for $i = 1, ..., n$. No data release is possible in this case because every inference is sensitive.

Second, if either the number of data sources or the number of inferences are increased, then the size of the DBN will grow. This will increase the complexity of learning the adversary model. It will also increase the amount of space and time required to obtain a solution (see the complexity analysis of the dynamic program approach).

Finally, since there are imperfections in any computational model that can be used by *mSieve* to detect data segments corresponding to a blacklisted inference, there are some lower bounds on the privacy level that can be achieved with *mSieve*. We formalize it with the following lemma.

Lemma 2. *Suppose false negative rate of the computational model used in mSieve for detecting black list $b \in B$ is F_b. Define $\eta = \max_{b \in \mathbb{S}_B}\{F_b\}$. Then lower bound of ϵ is $\ln(\eta)$.*

A proof appears in the Appendix. We note that the above limit can be improved by using better inference models.

Table 3.2: Data Statistics of study one (D-1)

Item	Avg. # sample per participant (One day)	Total # sample (37 Participants one day each)
Respiration	778,480	28,803,773
ECG	2,155,825	79,765,527
Accelerometer	682,991	75,812,073
Gyroscope	690,142	76,605,825

Table 3.3: Data Statistics of study two (D-2)

Item	Avg. # sample per participant (Three day)	Total # sample (6 Participants 3 days each)
Respiration	1,664,383	9,986,300
ECG	5,957,359	35,744,158
Accelerometer	1,362,898	8,177,390
Gyroscope	1,364,200	8,185,203

3.5 Evaluation

3.5.1 Study Design and Data Collection

We use data collected in two different studies to evaluate *mSieve*. We first summarize the data collection process and provide statistics of data from both studies which were approved by the IRB. In the first study (D_1), physiological data was collected from 37 participants. The goal of this study was to evaluate the value of wearable sensors in monitoring and reflecting upon daily stress, activity, and conversation. Each participant wore mobile-sensors for one full day. In total, 37 days of data was collected (one participant per day).

In the second study (D_2), data was collect from 6 daily smokers. Each participant wore mobile-sensors for three days, for a total of 18 days of data. The goal of this study was to develop and validate a computational model to detect smoking. In both studies, each participant wore a physiological chest band, inertial wristband, and GPS-enabled smartphone for a day. Each sensor transmitted data continuously to a mobile phone during the study period. At the end of each day, the data collected on the phone was transferred to a server.

In both studies, participants wore the sensors for 12.04 ± 2.16 hours per day during their awake period. Using the accelerometer sensor, we found that participants were physically active for 22.19% of the time, on average. Physiological data in the natural environment can be of poor quality for several reasons, such as physical activity, loose attachment, wireless disconnection, etc. [39]. We note that stress assessment model is applied to the data to obtain stress at each minute only when data collected was of good quality and not affected by these confounders [46]. Table 3.2 and Table 3.3 summarizes the number of data-points collected from participants in both studies.

Inferences from Sensor Data: We implemented the cStress model [46] to infer *stress inference*. It uses a set of features from both ECG and respiratory waveforms. If the stress probability of a particular minute is above 0.33, it is labeled as a *stressed* minute. We detected physical activity from wearable accelerometers using the model in [63]. We used the puffMarker model [2] to detect *smoking* episodes from wrist sensor data and respiration data. Finally, we used the mConverse model [44] to infer *conversation* episodes.

3.5.2 Model Learning

A key element of our scheme is the DBN model that we use for identifying the substitution segments. For model learning, we divided the day into state intervals. We then used the data from all the users (not a specific user) to study the convergence of the model learning, i.e., find the time interval over which the transition probabilities converged to a stable value. Let D_{con} be the converged transition matrix, and D_d be the conditional dependency probability matrix on day d. We define our normalized distance as $\frac{\sum(D_{con}-D_d)}{\sum D_{con}}$ where the sum is over all elements of the matrix. Figure 3.4 shows the convergence of DBN for multiple users. For both cases, more than 80% convergence occurs within first 9 days and 90% convergence occurs within 14 days.

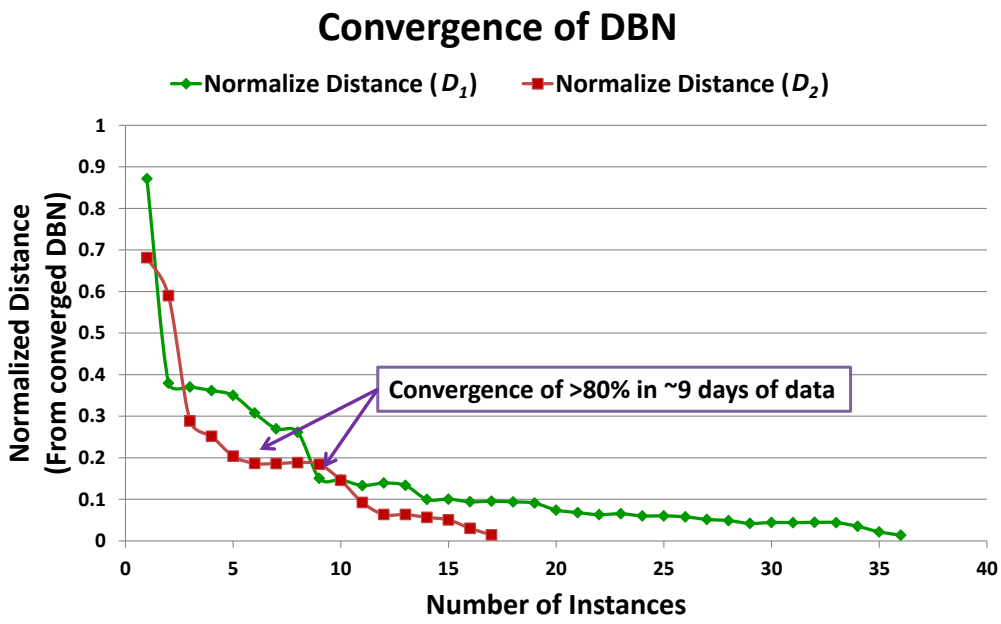
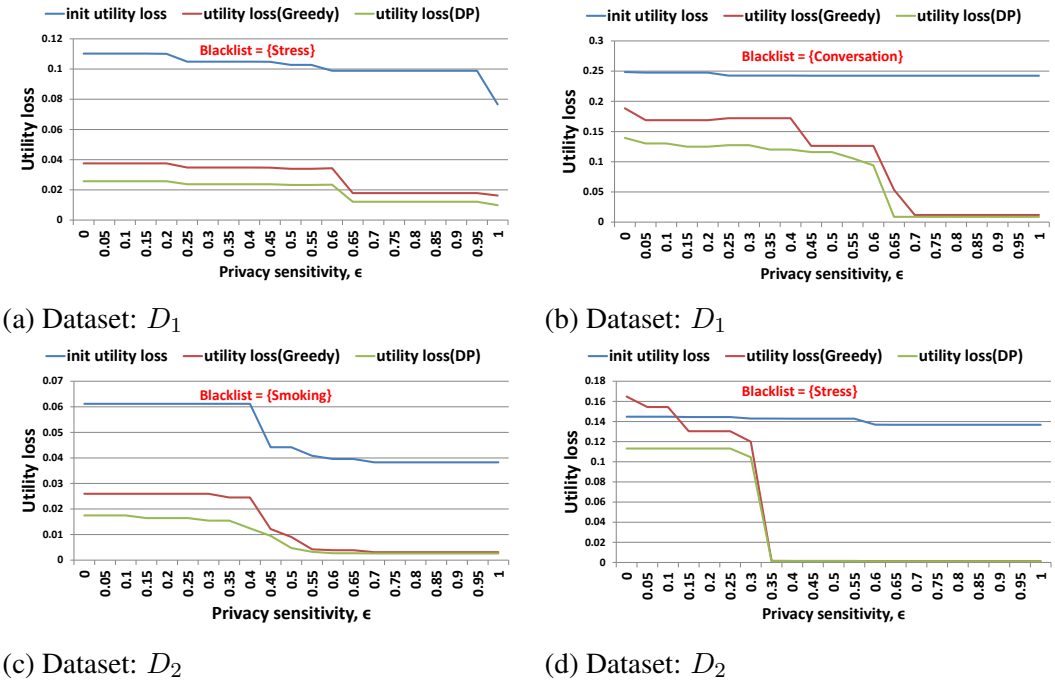


Fig. 3.4: Convergence rate for DBN training. The model is trained using aggregate data from all the users.

3.5.3 Privacy-Utility Tradeoff

To understand the privacy-utility tradeoff of both the dynamic program and greedy approaches, we vary the privacy parameter ϵ and observe the utility loss U_{loss} in each case. We conducted four experiments, two on each dataset, by changing the configuration of the blacklist. In the first experiment, we set the $Blacklist = \{Stress\}$, and in the second experiment we set the $Blacklist = \{Conversation\}$, and performed our evaluation on dataset D_1. In third and fourth experiments, we set the $Blacklist = \{Smoking\}$ and $Blacklist = \{Stress\}$ respectively and conducted the evaluation on dataset D_2. In all the experiments, we vary ϵ from zero to one in steps of size 0.05. Figure 3.5 shows the results obtained for each of the four experiments. Note, in each plot we show the utility loss for both the dynamic program and greedy approaches. Note also *init utility loss* is the utility loss after creating holes. When $\epsilon = 0$ we get $e^{\epsilon} = 1$, which means that the posterior belief about blacklist should not be more than the prior expectation. At that point, U_{loss} is maximum and we get an average of 11% utility loss for the greedy algorithm and 7%



(a) Dataset: D_1 (b) Dataset: D_1

(c) Dataset: D_2 (d) Dataset: D_2

Fig. 3.5: Privacy-Utility tradeoff for different blacklist configurations and varying privacy sensitivity ϵ. Results are shown for both the datasets.

utility loss for the DP algorithm. As we increase the value of ϵ, U_{loss} reduces. On average, we get less than 10% utility loss for both algorithm.

3.5.4 Dynamic Program vs. Greedy Algorithm

Utility Loss: We computed the U_{loss} value for both the dynamic programming algorithm and the greedy algorithm. In Figure 3.8, we also show U_{loss} after substitution. DP always produces better result than the greedy algorithm and both produce better results than the initial U_{loss}. For some users, initial U_{loss} is less than our solution. This is because, using our approach, we always fill each hole resulting in occasional overfilling of the whitelist inferences.

Distribution of Safe, Unsafe, and Sensitive States: We investigate the distribution of the three states in the data released by each algorithm. To do so, we vary the value of the privacy sensitivity ϵ and observe the percentage of nodes that are safe, unsafe, and sensitive. Figure 3.6 shows the results. Recall that because of plausible substitution, in addition to safe and sensitive states, we also have unsafe states that are not-sensitive, yet

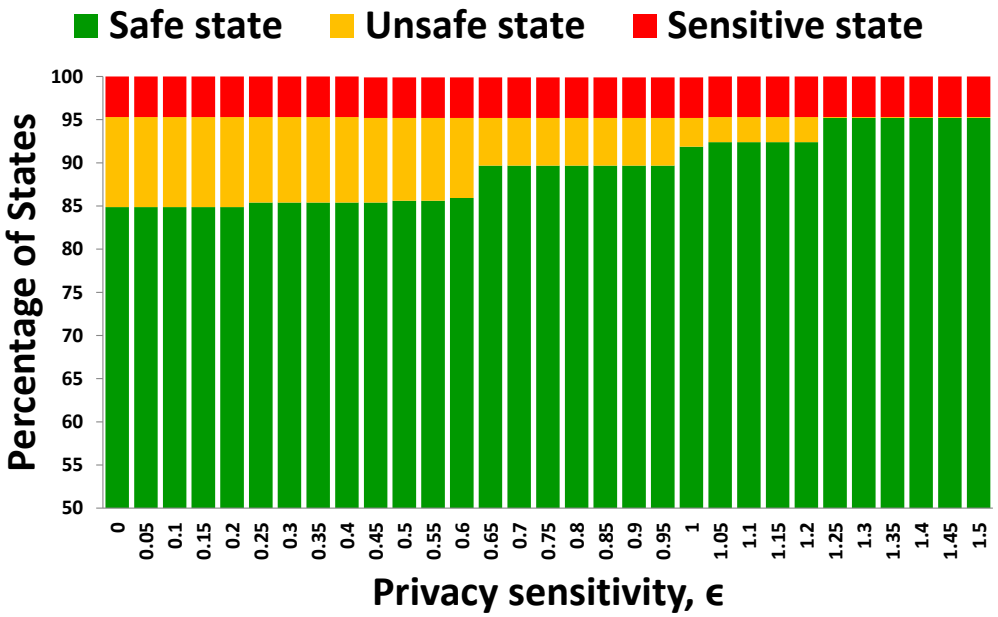


Fig. 3.6: Variation in the percentage of node types with privacy sensitivity ϵ. Recall, lower value of ϵ means higher privacy.

unsuitable for release as they are highly correlated with sensitive states. As we increase the value of ϵ, the number of safe states also increases. For a given privacy sensitivity to $\epsilon = 0.5$, the average distribution of the various state types in a user trajectory is shown in Figure 3.7.

3.6 Related Work

Various transformation techniques have been proposed to protect data privacy. Below, we summarize some of the techniques relevant to our problem setting.

Anonymization metrics: A vast majority of the literature on privacy preserving data publishing consider anonymization metrics, such as k-anonymity [51], l-diversity [52], and t-closeness [53]. These approaches operate under the threat model in which an adversary is aware of quasi-identifiers in a multi-user dataset and wants to perform linkage attack using other auxiliary data sources to infer the sensitive attributes of individuals within the same dataset. We consider a different setting where data from a single user (no multi-user database is present) is being protected against sensitive inferences. We further assume that the identity of the user is known and hence the anonymization mechanisms

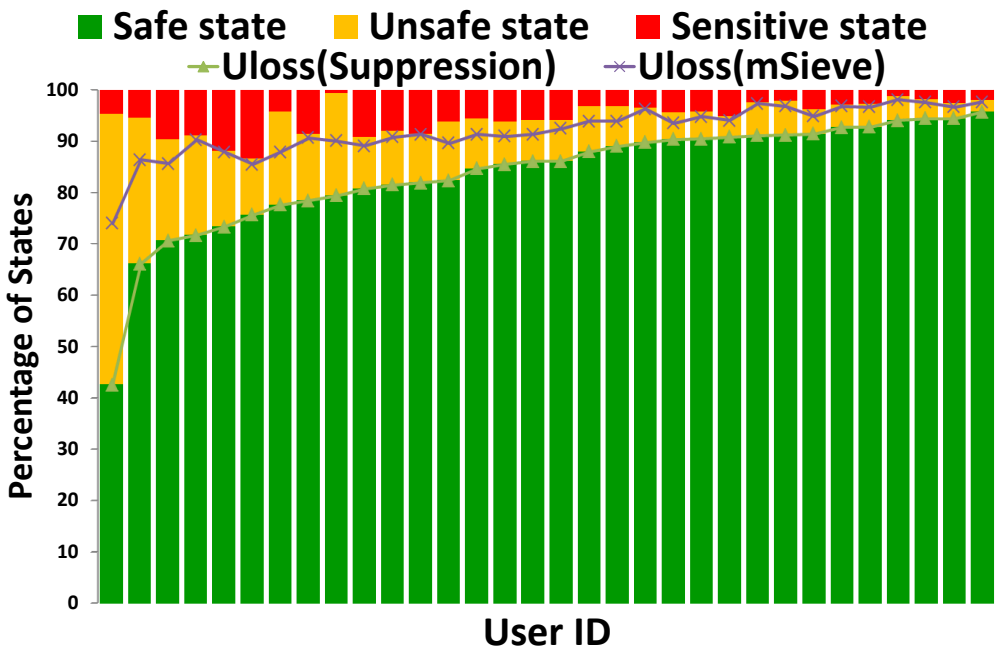


Fig. 3.7: Percentage of node types for the different users. Users IDs are sorted in ascending order of safe node percentage. Privacy sensitivity $\epsilon = 0.5$.

are not useful. In addition, instead of static (time-independent) relational databases, we consider time-series of physiological sensor data.

Data Randomization: Randomization techniques add noise to the data in order to protect the sensitive attributes of records [64, 65]. Evfimievski et al. [66] proposed a series of randomization operators to limit the confidence of inferring an item's presence in a dataset using association rule mining. Differential privacy offers a formal foundation for privacy-preserving data publishing [67, 68]. It can be classified as a data distortion mechanism that uses random noise (typically from a Laplace distribution) to ensure that an attacker, with access to the noisy query response, fails to guess the presence or absence of a particular record in the dataset. Compared to the general-purpose data publication offered by k-anonymity, differential privacy is a strictly verifiable method [69]. A survey of results on differential privacy can be found in [54]. While most of the research on differential privacy has focussed on interactive settings [55–57], non-interactive settings as an alternate to partition-based privacy models have also been considered in recent works [70–72]. Since we focus on physiological data, the randomization approaches are

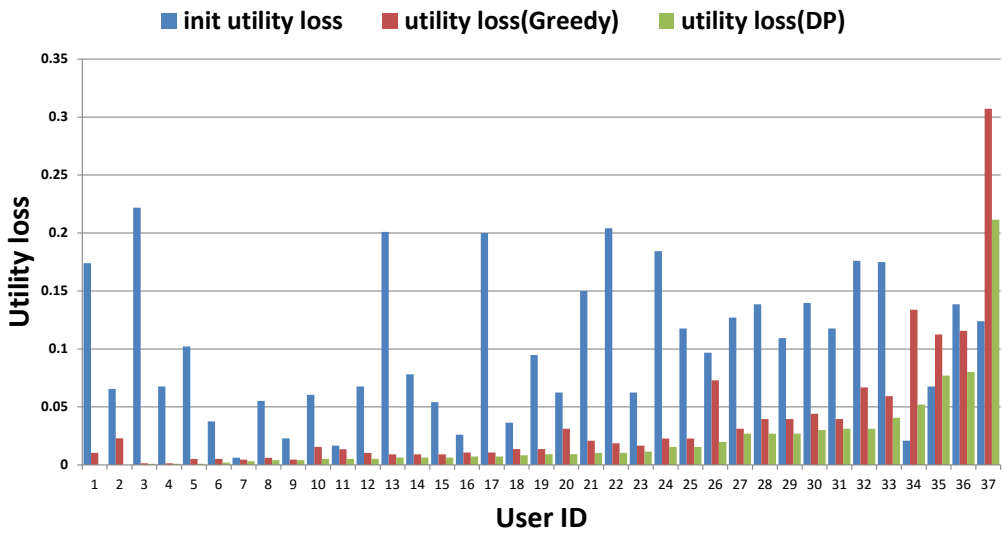


Fig. 3.8: Comparison of utility loss for the DP and Greedy algorithms. The user IDs are sorted according to the utility loss using the DP algorithm.

often unsuitable as they distort the data making it unusable for critical inferences, especially for physiological data.

Distributed privacy preservation: Results are aggregated from datasets that are partitioned across entries [73]. Partitioning could be horizontal [74–77], i.e., records distributed across multiple entries, or vertical [74, 75, 78], i.e., attributes distributed across multiple entries. It is possible that individual entities may not allow sharing their entire dataset. However, they consent to limited information sharing with the use of a variety of protocols. The overall goal of such techniques is to maintain privacy for each individual entity, while obtaining aggregated results. We consider a single user setting for which the above techniques do not work.

Data Synthesis: Synthetic data generators exists for location data [79] that are used to protect the privacy of sensitive places. To obtain these synthetic traces, data from different users are pooled together into a population-scale model, which is then used to generate the data. Physiological data of each person is unique and used for obtaining bio-metrics [80, 81]. Thus, population-scale models often results in significant degradation in utility of the data. The well-defined temporal correlation between data segments in a

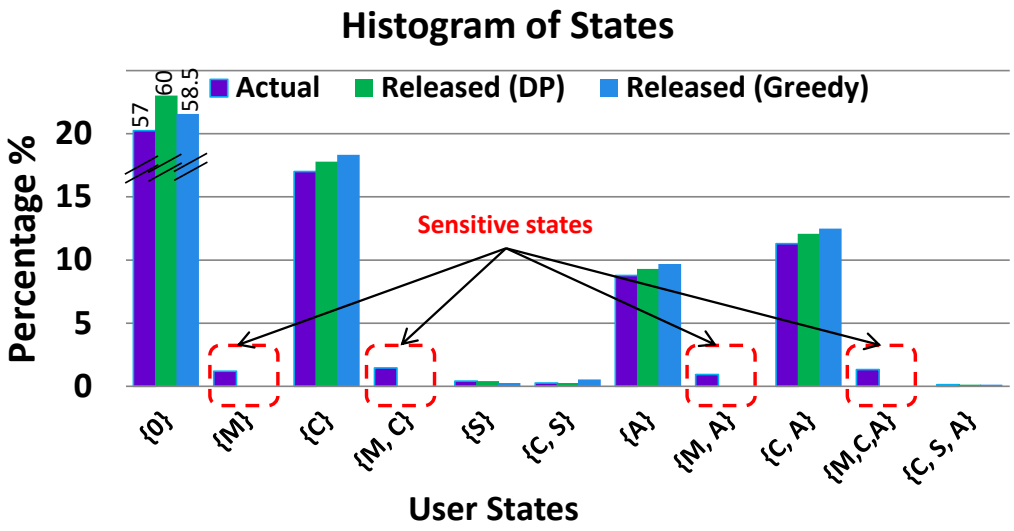


Fig. 3.9: Comparison of histogram of different states between actual data and released data.

physiological time series (i.e., continuity between presiding and succeeding contexts) makes it even harder to generate synthetic data.

In summary, protecting behavioral privacy when sharing time series of mobile sensor data (especially physiological data), pose new challenges and unique research opportunities that have usually not been considered in prior works.

3.7 Limitations and Discussion

We presented *mSieve*, as a first step towards building systems that can use substitution as an effective mechanism to protect privacy of behavior while sharing personal physiological data. Instead of random substitution of sensitive segments, which can degrade the utility of the overall dataset, *mSieve* performs model-based substitution. Our approach opens up new directions in privacy and differential privacy research, namely to define suitable metrics and mechanisms that can be used to protect private behaviors in a time series of mobile sensor data. Specific challenges are as follows:

- **Algorithmic Scalability:** There are several aspects that constitute the scalability of the system. Although, we have applied our model to several data sources, its applicability to other newer sources of mobile sensor data is yet to be established. Furthermore, an increase in the number of sensors, would imply a significant

increase in the number of possible inferences (obtained using different combinations of the sensors), and a corresponding increase in the whitelist and blacklist set sizes. Improving the efficiency of our algorithms in computing candidate segments in such scenarios, which effectively would depend on the size of the DBN model, is an interesting open question.

- **Offline vs. Online:** Our model is an offline model that assumes availability of all data. In practice, data may need to be shared in real-time as they are produced. Significant adaptations may be needed to develop an online version of *mSieve* that can run in real-time on mobile phones.

- **Adversarial Setting:** Stronger adversarial models, where the adversary may possess more information regarding the user than the behavioral model captures, may lead to new challenges in ensuring privacy guarantees and represents interesting privacy research. In fact, this also represents a significant bottleneck for model-based privacy approaches, where a model is essential for maintaining the utility of the shared data, but the very use of a specific model leads to assumptions on adversarial capabilities. While in principle such approaches can be modified to protect against a worst-case adversary, it is difficult to provide meaningful utility in such cases.

- **Plausibility:** The current formulation of *mSieve* only provided local plausibility by adjusting the boundaries of substituted segments. Incorporating plausibility as part of the privacy formulation itself will be an interesting extension of the current work.

- **Privacy leakage due to *Graylist*:** We consider inferences that are associated with whitelist or blacklist, but other behaviors could be inferred from raw sensor data. We call those additional inferences as *graylist*. Information leakage due to *graylist* need to be investigated further.

- **New and emerging inferences:** Finally, rapid advances are being made in being able to infer human behaviors from seemingly innocuous sensor data, which may challenge the notion of white list and black list, especially when raw sensor data is being shared.

3.8 Conclusion

We presented *mSieve*, a system that uses substitution as a mechanism to protect privacy of behaviors to facilitate sharing of personal physiological data collected continuously in the natural environment. Instead of random substitution of sensitive segments, which can degrade the utility of the overall dataset, we perform model-based substitution. We employ a Dynamic Bayesian Network model that allows us to search for plausible user-specific candidate segments that satisfy the statistics of the sensitive segment and thus preserve the overall consistency of the shared data. Through experimentation on real-life physiological datasets, we demonstrated that our substitution strategies can indeed be used for preserving the utility of inferences while achieving differential behavioral privacy. This work opens the doors for follow up research and real-life deployment as adoption of physiological sensors in daily wearables such as smartwatches grow.

Chapter 4

User Re-identification Risk and Mitigation Approach from Wrist-worn Accelerometry Data

4.1 Introduction

Increasing number of sensory datasets consisting of motion sensor data from wrist-worn devices are becoming publicly available for research (shown in Figure 4.1). It is indicative of a larger trend: growing adoption of wrist-worn devices (e.g., smartwatches, activity trackers) and a growing research body that uses motion sensor data from wearables to make new inferences of daily behaviors. These novel inferences range from detection of mundane daily behaviors such as eating [4], drinking [82], brushing and flossing [5], to more sensitive ones such as smoking [48], tremors [83], pain [84], and drug use [7].

Each released database usually strips the data of any identifiers and is typically anonymized using recommended practices (e.g., using k-anonymity [11], l-diversity [12], and t-closeness [13]) before release. But, they consist of raw sensor data streams assuming low risk of re-identification. Although there exist works showing that attributes such as *age, gender, race*, or *job type* can be inferred from accelerometer data alone [85–88], but they need additional information, such as restaurant check-ins, to re-identify the user [89, 90], which is not available in publicly released datasets.

In this paper, we investigate the problem of re-identifying a user only from her accelerometer data from wrist-worn devices that is usually released in public datasets. If so, a user contributing to different datasets can be linked, resulting in revelation of attributes, health states, and behaviors present in any of these datasets. For example, an insurance company or an employer collecting motion sensor data of its subscriber or employee (to reward healthy lifestyle) can learn of the users' prior history with smoking, pain, drug use, tremor, etc., from public datasets that this user may have contributed to previously to help advance science.

The device fingerprinting problem is closely related to our problem. In recent years, several approaches have been proposed for device fingerprinting including techniques that model the noise distribution in inertial sensors [91], use factory calibration data [92], or a vibration motor to stimulate accelerometer data [93]. But, these approaches cannot be extended or easily adopted for user re-identification.

Similarly, several accelerometer-only approaches using gait features [94, 95], or hand movements [96] have been proposed for user authentication on smartphone and smartwatches. But, most of these studies were conducted in controlled environments, with participants performing activities as per a set script. They provide a key insight that for a given user, gait represents a unique pattern during a specific activity (e.g., walking, running etc.). We hypothesize that these unique discriminative patterns can be extracted using micro-event modeling and used to re-identify an individual.

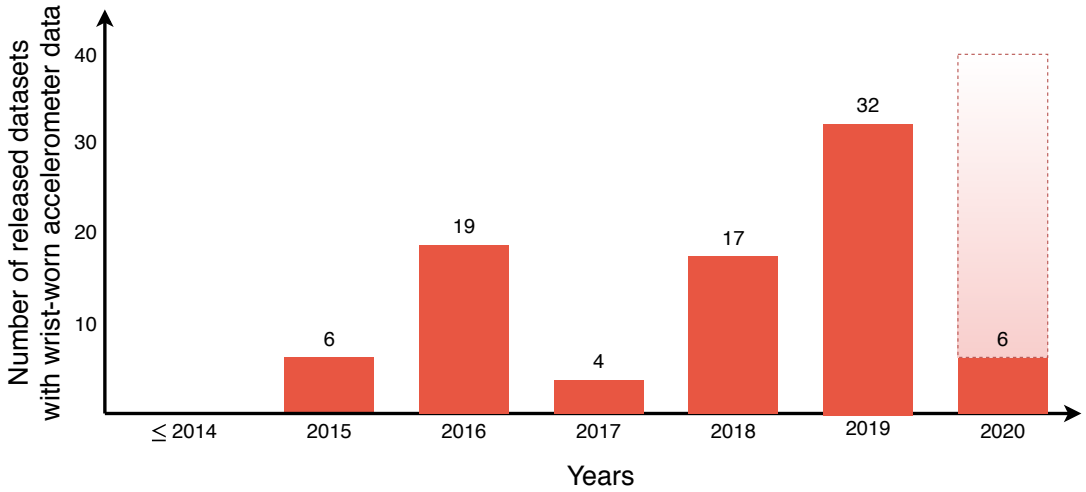


Fig. 4.1: Based on the search results of 'wrist acelerometer' from Google's dataset search. Number of open datasets increases over time.

In our problem formulation, we assume an adversary with access to an anonymized sensor database consisting of labeled wrist-worn accelerometry data from n users. The labels may refer to a health condition or diagnosis of the user, or unhealthy daily behaviors for which the researchers were seeking to develop a treatment or intervention for. Furthermore, the adversary also has access to wrist-worn accelerometer

data from a user whose identity is known to the attacker (Figure 4.2). The goal of the adversary is to determine, with high confidence, if the users' data are also contained in the anonymized database and, if so, re-identify the anonymized user in the database.

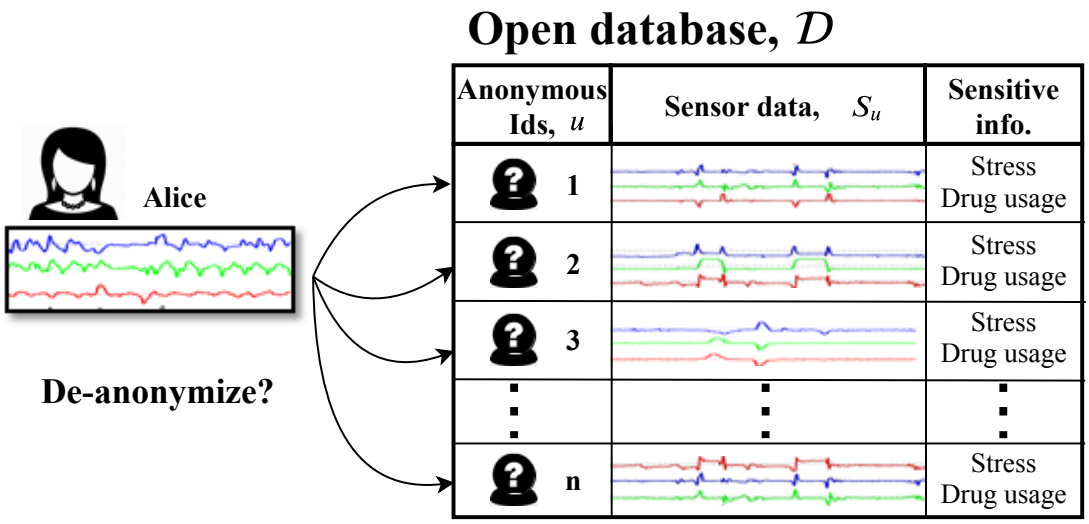


Fig. 4.2: Example scenario of user re-identification.

Successful re-identification of a user from wrist-worn accelerometer data involves addressing several technical challenges. First, sensory data are unstructured representing multiple interleaving activities, making feature extraction difficult. Second, accelerometer data are usually low-frequency and noisy, which complicates the task of finding unique patterns from the data. Third, the amount of sensor data in the training set, i.e., released database, may vary from the amount of testing data available. Moreover, collecting sensor data in research studies is expensive. Therefore, in practice, an adversary might have access to only a small segment of user sensory data. Fourth, re-identifying users is an open set classification problem. In other words, the database will not contain the data of a large number of users and for all these users the model should correctly predict their absence from the database. Fifth, scalability or uniqueness of the solution is highly desirable but can be difficult to achieve as even the unique signature of a user might change over time. Finally, despite a growing number of public databases, there is still a

lack of wrist-worn accelerometry dataset that consists of data collected from the unscripted free-living environment for long periods of time.

We present a deep learning model (called *WristPrint*) to solve the user re-identification problem from wrist-worn acceletometer data collected from the natural environment. We first develop a filter to identify segments of data from whole day data that consists of identifiable patterns. Second, we find optimal quantization of the filtered data that allows for unique patterns identifying an individual, while still allowing repeated assessments for boosting re-identification performance. Third, we develop a base deep learning model. We use a convolutional neural network layer to extract latent representation of micro movements and a recurrent neural network layer to identify temporal pattern in a sequence of micro movements.

Fourth, we need a loss function that can guide the learning of base model to minimize the intra-class variation (for consistency in identifying a user) and maximize the inter-class distance in feature space (to amplify distinction among different users). Taken together, such a loss function should not help achieve high re-identification rate for known users, but also leave the feature space largely unencumbered so as to recognize the absence of an unknown user when presented with their test data. To solve this open set learning problem, we propose a novel consistency-distinction loss function.

Fifth, we present two boosting models that uses the output of the base model on each unit of test data to further improve the re-identification performance.

Sixth, we use a new dataset consisting of 353 users (knowledge-workers holding full-time jobs in diverse industries with a wide variety of job functions) who wore a wrist-worn device daily for ten weeks, contributing a large amount of motion sensor data. We split this data into train and test sets and experiment with different values of data quantization, training data length, test data length, and decision threshold (for deciding when the likelihood of a class is large enough to declare a re-identification). We find that two hours of physically-active data is sufficient to train our model for re-identifying a

known user with 99.7% true matching rate (when presented with 60 minutes of test data), while keeping the false acceptance rate to 0.1%. Finally, we experiment with different noise models to determine the kind and level of noise addition can mitigate the re-identification risk. We find that introducing even a low level of Laplace noise can be used to successfully mitigate the re-identification risk identified here.

4.2 Problem Setup

We first introduce notations, define the terms we use throughout the paper, and formalize the re-identification problem as an open-set machine learning problem.

Table 4.1: Symbols and Notations

Notation	Meaning
s_u	Sensor trace of user with Id u
$\mathcal{D}$	Sensor database, $\mathcal{D} = \bigcup\{(s_u, u)\}$
$\mathcal{U}$	Known users: Set of User Id's $u : s_u \neq \phi \in \mathcal{D}$
$\mathcal{U}'$	Unknown users: complement of $\mathcal{U}$
Δ	Common unit length of quantization for all s_u
s_u^i	i^{th} segment (of length Δ) from s_u , for $i \in \{1, 2, ..., m_u\}$, where $m_u = \lfloor \lvert s_w \rvert / \Delta \rfloor$
$\mathbb{D}$	Quantized version of dataset $\mathcal{D}$, $\mathbb{D} = \bigcup\{(s_u^i, u)\}$
$\mathbb{D}_{train}, \mathbb{D}_{val}, \mathbb{D}_{test}$	Train, validation, and test set partitions of $\mathbb{D}$
$\mathcal{F}$	Feature space of sensor data in $\mathbb{D}$
$\phi : \mathbb{D} \rightarrow \mathcal{F}$	Feature generator
$\varphi : \mathcal{F} \rightarrow [0, 1]^n$	Likelihood from classifier, where $n = \lvert\mathcal{U}\rvert$
$0 \leq \mathcal{T} \leq 1$	Decision threshold for likelihood for a positive re-identification
$M_\Delta = (\phi, \varphi, \mathcal{T})$	Base classification model
$\mathcal{M}$	Boosting model with N boost
ρ_Δ	Accuracy of base model
$\rho(\rho_\Delta, m)$	Accuracy of boosting model with m boost

4.2.1 Notation and Terminology

Throughout this paper, we use the following definitions to describe our wearer re-identification solution.

Sensor Trace Let $s_u(t) = (v_1(t), v_2(t), ..., v_d(t)) \in \mathbb{R}^d$ denote the sensor data point from a user u at time t. A *sensor segment* $s_u(t_s, t_e)$ is a contiguous time-series of sensor data

point from time t_s to t_e, i.e., $s_u(t_s, t_e) = \{s(t) : t_s \leq t \leq t_e\}$. We define *sensor trace* s_u as a collection of all sensor segments from a user u, i.e., $s_u = \bigcup_{(t_s,t_e)} \{s_u(t_s, t_e)\}$.

For application, we use accelerometry data collected from wrist-worn devices, such as activity trackers and smartwatches.

Sensor database Let $\mathcal{U} = \{1, 2, ..., n\}$ be the set of user identifiers with non-empty sensor trace, s_u. Then, sensor database $\mathcal{D} = \bigcup_{u \in \mathcal{U}} \{(s_u, u)\}$. For any user $u \in \mathcal{U}$, with non-empty entry $(s_u, u) \in \mathcal{D}$ we say that $u \in \mathcal{U}$, otherwise $u \notin \mathcal{U}$.

4.2.2 Data Selection

The ability to re-identify users from accelerometry data depends on discovering unique patterns that are specific to each wearer. Accelerometry data collected from the natural environment consists of several segments that have low variability such as when the sensor is not worn or when the sensor is worn, but the user is sedentary. In such cases, there is insufficient variability in the data to reveal user-specific patterns. Therefore, we need criteria to identify segments of data (for both training and testing) that have sufficient variability.

Prior work on user authentication [94–97] have shown that users can be authenticated from gait analysis during walking or running by using data from accelerometers placed on strategic locations of the body (e.g., extremities and torso). Even though these works used a controlled and scripted environment, limiting their applicability in the free-living environment, they show that accelerometry signals during walking or running consist of a unique signature. But, people usually walk only 10% of the time, follow no scripted pattern, and may exhibit variability in their gait in different situations (e.g., brisk walking, leisurely stroll, rushing, etc.). Therefore, to re-identify users, we seek to identify a common pattern when a person is physically active, despite natural variabilities.

To find physically-active segments, we use a variance-based approach described in [39]. At first, we compute the minute-wise standard deviation of the magnitude of accelerometry data. We then filter out those one-minute segments as stationary if the standard deviation is more than a threshold. The selected value of the threshold is 0.21.

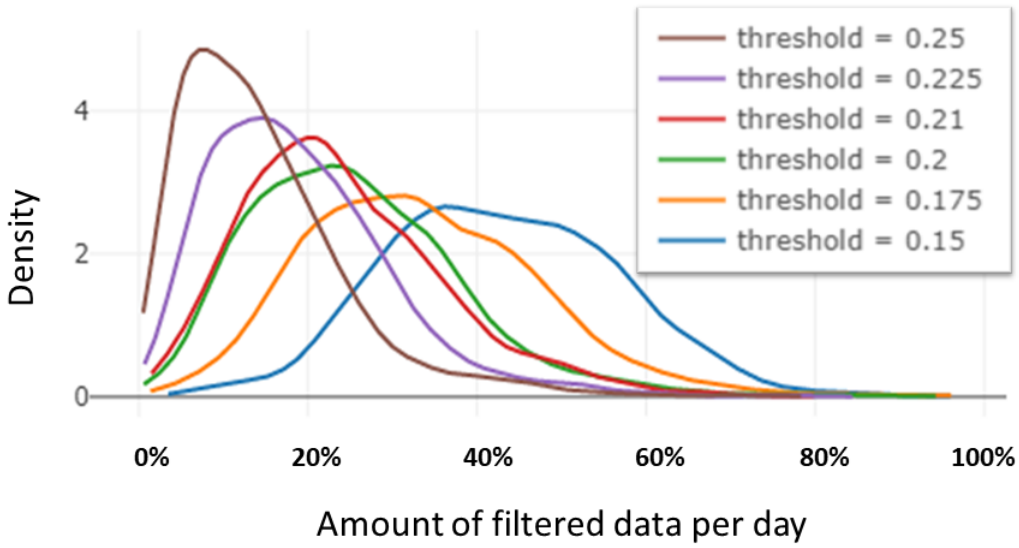


Fig. 4.3: Amount of movement over a day.

To identify a unique signature that suffices for user re-identification, the selected physically-active situations should be sufficiently frequent in the free-living environment. Figure 4.3 shows that applying our criteria on our free-living dataset results in approximately two hours of active data per day.

Without loss of generality, we now assume that the database $\mathcal{D}$ as well as any externally-provided test data consist of only those accelerometry segments that have sufficient variability.

4.2.3 Attack Model

We assume that the adversary has access to a database $\mathcal{D}$ and an accelerometry trace s_x from a user. The goal of the attacker is to determine whether the user of s_x is in $\mathcal{D}$, and if so, to determine the identity u of the user.

Similar to other re-identification problems [98, 99], our problem can be formulated

as a similarity search problem. In a similarity search problem, one is given a database of items and a similarity function. The similarity is large if two items are similar and small if they are not similar. Given a new item, one wants to efficiently find the item closest to this new item in the database. Usually, in a similarity search problem the similarity metric is defined using a suitable mapping $\phi : D \rightarrow F$ of the items to some metric space F, and then $sim(s_i, s_j) =_F (\phi(s_i), \phi(s_j))$ where $_F(\cdot, \cdot)$ denotes the metric in F. For example, in the fingerprint matching problem, the mapping might map a fingerprint image to fixed set minutiae [98] or a compact fixed length FingerCode [99], and the distance metric is the Euclidean distance.

To solve our user re-identification problem, one can find, using a suitable similarity search query, the most similar data to the given input s_x. Here, our database $\mathcal{D}$ consists of time-series segments. But, instead of being confined to a fixed set of activities such as walking or running, sensor data collected from daily life consist of a whole range of activities a user may perform in the free-living environment. Therefore, our challenge is to identify suitable structural patterns that can be considered as features or latent state, where the mapping ϕ, as well as the metric space F are not obvious. We develop machine learning algorithms to discover distinctive features or latent state from these unstructured data.

We assume the adversary employs a classification model $\mathcal{M}$ for the re-identification task. Here, we say that the attack succeeds if the attacker can correctly re-identify the user, i.e., if for any given input s_x from a known wearer $u \in \mathcal{U}$, mapping $\mathcal{M}$ outputs wearer u and similarly, for any given input s_x from an unknown wearer $\bar{u} \notin \mathcal{U}$, model outputs 0 for "not present".

4.2.4 Privacy Risks

Let $\mathcal{D}$ be a released dataset. We compute the risk score r_l as expected difference between the posterior probability and the prior probability of detecting user identity u

from any test data s_x of length l,

$$r_l = E_{s_x,|s_x|=l,x=u}\left[Pr(u|\mathcal{D}, s_x) - Pr(u|s_x)\right] \tag{4.1}$$

We consider a simple prior, i.e., random guess, so that the prior probability for any user $u \in \mathcal{D}$ is $\frac{1}{n}$ for all s_x. We assume that the attacker learns a machine learning model $\mathcal{M}$ using $\mathcal{D}$ such that for any test data s_x, $\mathcal{M}(s_x)$ outputs the closest matching user identifier from the database, if the matching score is acceptable, and 0 otherwise. We can now reformulate the risk score as follows:

$$r_l = E_{s_x,|s_x|=l,x=u}\left[Pr[\mathcal{M}(s_x) = u]\right] - \frac{1}{n}. \tag{4.2}$$

We say that a released open database $\mathcal{D}$ has (ϵ, l)-re-identification risk against an adversary, if for any test data s_x of length l, the risk score for identifying the true user is greater than ϵ:

$$r_l > \epsilon \tag{4.3}$$

Minimum length of test data: In addition to the risk of re-identification, we also want to know the extent of the risk, i.e., for a given ϵ, what is the the minimum amount of data required for the attack model to re-identify any user $u \in \mathcal{D}$ with at least ϵ risk score?

$$l_\epsilon =_l r_l > \epsilon \tag{4.4}$$

4.3 Proposed Approach: *WristPrint* Model

We now present the overall architecture of the attack model and a high-level overview of the re-identification attack.

Overview of the *WristPrint* Model

An overview of the components in the the end-to-end *WristPrint* model appears in Figure 4.4. At first, it segments sensor trace into fixed unit size segments. Then feature

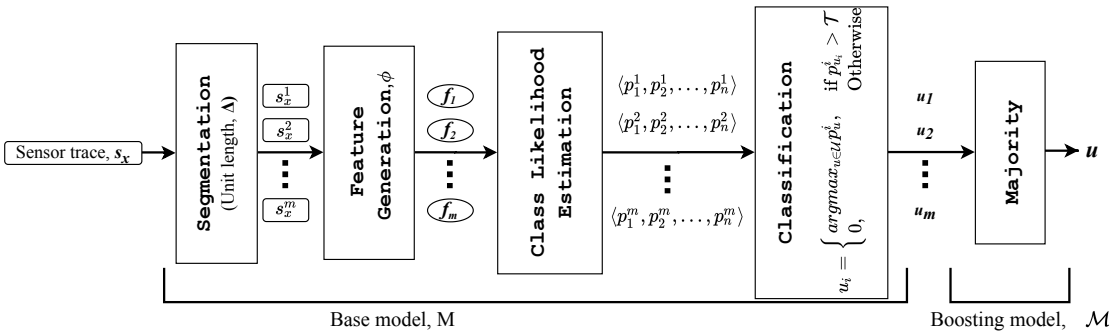


Fig. 4.4: Overview of the re-identification algorithm.

generator maps these segments into a feature space. The goal of this feature transformation and separation is that all the points in the feature space from the same individual fall into one region. That is, all the points should create a cluster, and each cluster should be separated. Then classifier assigns each feature vector to the nearest user id. Finally, it combines the outputs of the classifier applied to each segment to determine the final user id label. We call this base-boosting pair model architecture since the base model takes each unit length segment as input and detects user identifiers as output and boosting model groups the detected user identifiers to produce a single detection.

Re-identification Attack

In this section, we present the overview of the re-identification attack as illustrated in Algorithm 5.

At first, the model segments each sensor trace $s_u \in \mathcal{D}$ into unit segments of length Δ. Let $\mathcal{S}_u = \{s_u^1, s_u^2, ..., s_u^{m_u}\}$ be the set of all Δ-long segments of sensor data from user u. These segments from all users generate a new database $\mathbb{D} = \{(s_u^i, u) | s_u^i \in \mathcal{S}_u, \forall u \in \mathcal{U}\}$.

As in other similarity search problems, we consider a base model M_Δ which is a randomized algorithm. Given an unseen sensor data segment s_x^i, M_Δ outputs a user $u \in \mathcal{U}$ from $\mathcal{D}$ or a special identifier for "not present". Such a randomized algorithm might be a machine learned classifier.

We assume M_Δ is a function composition of ϕ and φ, i.e., $M_\Delta = \phi \circ \varphi$. Here, the function ϕ, trained possibly by a neural network model, which maps $s_u^i \in \mathbb{D}$ into an

Algorithm 5 `WristPrint` , $\mathcal{M}(\mathcal{D}, s_x)$

Input: $\mathcal{D}$: Sensor dataset of n users
s_x: Sensor trace from an unknown user
Output: User of s_x
function TRAINBASEMODEL($\mathcal{D}$)
  $\Delta = \texttt{FindOptimalUnitLen}(\mathcal{D})$
  $\mathbb{D} = \texttt{Segment}(\mathcal{D}, \Delta)$
  $\phi, \varphi, \mathcal{T} = \texttt{train}(\mathbb{D})$
  ▷ ϕ: Feature generator, φ: classifier, $\mathcal{T}$: decision threshold
  $M_\Delta \equiv (\phi, \varphi, \mathcal{T})$
  return M_Δ
end function
$M_\Delta = \texttt{TrainBaseModel}(\mathcal{D})$
$\langle s_x^1, ..., s_x^m \rangle = \texttt{Segment}(s_x, \Delta)$ ▷ $m = \lfloor \frac{s_x}{\Delta} \rfloor$
Initialize $P \leftarrow \phi$
for $i = 1$ **to** m **do**
  $x = M_\Delta(s_x^i)$
  P.`add(x)`
end for
return `findMajority`(P)

appropriate feature space $\mathcal{F}$. The function ϕ needs to be trained such that feature space $\mathcal{F}$ preserves consistency within the same user and distinction among different users. Therefore, the goal is to maintain intra-class similarity and inter-class differences.

We now need to solve the similarity search problem mentioned above. However, our situation is slightly different since for a given user there is no single point in $\mathcal{F}$ but a set of points in the feature space. The adversary trains a classifier, $\varphi : \mathcal{F} \rightarrow [0, 1]^n$, that creates partitions in feature space for each class. For a feature vector of any given sensor data segment s_u^i, it outputs the probability of each class, i.e., $\varphi(\phi(s_u^i)) = \langle p_1, .., p_n \rangle$ where $p_i = Pr[u_i|s_x]$. Finally, an threshold $\mathcal{T}$ is learned, such that if all the probabilities are less than the threshold, then it outputs "not present".

The re-identification attack can now be defined as:

Sensor to wearer mapping M_Δ A mapping M_Δ is a tuple (ϕ, φ, $\mathcal{T}$), where:

- Δ: Optimal length of the input sensor data segment

- $\phi : \mathcal{D} \rightarrow \mathcal{F}$ is a mapping function
- $\varphi : \mathcal{F} \rightarrow [0,1]^n$ is a classifier that generates the probability of each class
- Decision threshold $\mathcal{T}$

For any given accelerometry data segment s_x^i of length Δ, generating output is a two step process: a) compute the maximum probability over known classes, and b) label the data as *"not present"* if all probabilities are below the decision threshold $\mathcal{T}$:

$$u* = M_\Delta(s_x^i) = \begin{cases} {}_{u_i \in \mathcal{U}} Pr[u_i|s_x^i], & \text{if } Pr[\bar{u}|s_x^i] > \mathcal{T} \\ 0 \text{ or 'unknown'}, & \text{Otherwise} \end{cases}$$

where, $\varphi(\phi(s_x^i)) = \langle p_1, p_2, ..., p_n \rangle$ and $p_i = Pr[u_i|s_x^i]$

We call the model M_Δ as **Base Model** since it works on unit length data.

We say that the attack succeeds if the attacker can correctly re-identify a user, i.e., if for any given input s_x^i from a known wearer $u \in \mathcal{U}$, mapping M_Δ outputs wearer u. Similarly, for any given input $s_{\bar{u}}^j$ from an unknown wearer $\bar{u} \notin \mathcal{U}$, the model outputs 0 for "not present". We define the accuracy of the base model as follows.

Accuracy of the *base model*, ρ_Δ The accuracy ρ_Δ of the mapping M_Δ is equal to the probability of correctly detecting the user from an unknown user's unit sensor segment in one run of the mapping algorithm. That is for any sensor segment s_u^i from a user $u \in \mathcal{U}$ and for any sensor segment $s_{\bar{u}}^j$ from a user $\bar{u} \in \mathcal{U}'$ the value if ρ_Δ is defined as,

$$\rho_\Delta = E\left[Pr[M(s_u^i) == u \text{ and } M(s_{\bar{u}}^i) == 0]\right]$$

Clearly, to be better than a random assignment we want $\rho_\Delta > 1/(n+1)$ where n is the number of users in $\mathcal{D}$.

For a given database $\mathcal{D}$ and a sensor traces s_x of length more than Δ, the attacker creates a **Boosting Method** $\mathcal{M}$ by running base model $L \equiv \frac{|s_x|}{\Delta}$ times. Boosting model

combines the results of the base model to generate the final output. Boosting capability depends on the value of l, i.e., if value is greater then boosting capability is high and vice versa. We can now formulate the re-identification task with quantified parameters guaranteeing the quality, as opposed to the basic mapping outlined above.

Re-identification Task: *Given a database $\mathcal{D}$ of sensor data from user $u \in \mathcal{U}$, we want to find/ train a model $\mathcal{M}$ that minimizes the error bound ϵ. That is, for any sensor trace s_x from unknown user $x \in \{\mathcal{U} \cup \mathcal{U}'\}$, we want to*

$$\text{minimize } \epsilon$$

Such that,

$$Pr[\mathcal{M}(s_x) = u] \text{ is } \begin{cases} \geq 1 - \epsilon, & \text{if } x = u \in \mathcal{U} \\ < \epsilon, & \text{otherwise} \end{cases}$$

As we argue in the next section, the accuracy $\rho*$ of the mapping algorithm $\mathcal{M}$ depends on the length of training data in $\mathcal{D}$ and the length of the test data s_x.

4.4 Base Model

In this section, we present our proposed architecture of the base model. Since the accelerometry trace is unstructured, and we seek distinctive features for each person, we employ a deep-learning architecture as a base model. The accelerometry trace has been segmented in such a way that each segment's length is Δ. In Section 4.5.2, we describe how to find the optimal value of Δ, given the base model's structure. For model development, we consider an arbitrary length Δ of the sensor segment.

Our model consists of two key blocks of computation: feature computation and classification. Since we seek to identify unique pattern over the accelerometry time-series when users are physically active, we consider the signal characteristics along both the time and amplitude axes to create a unique fingerprint of the user. As shown in Figure 4.5a, base model's overall architecture consists of two convolutional layers, two

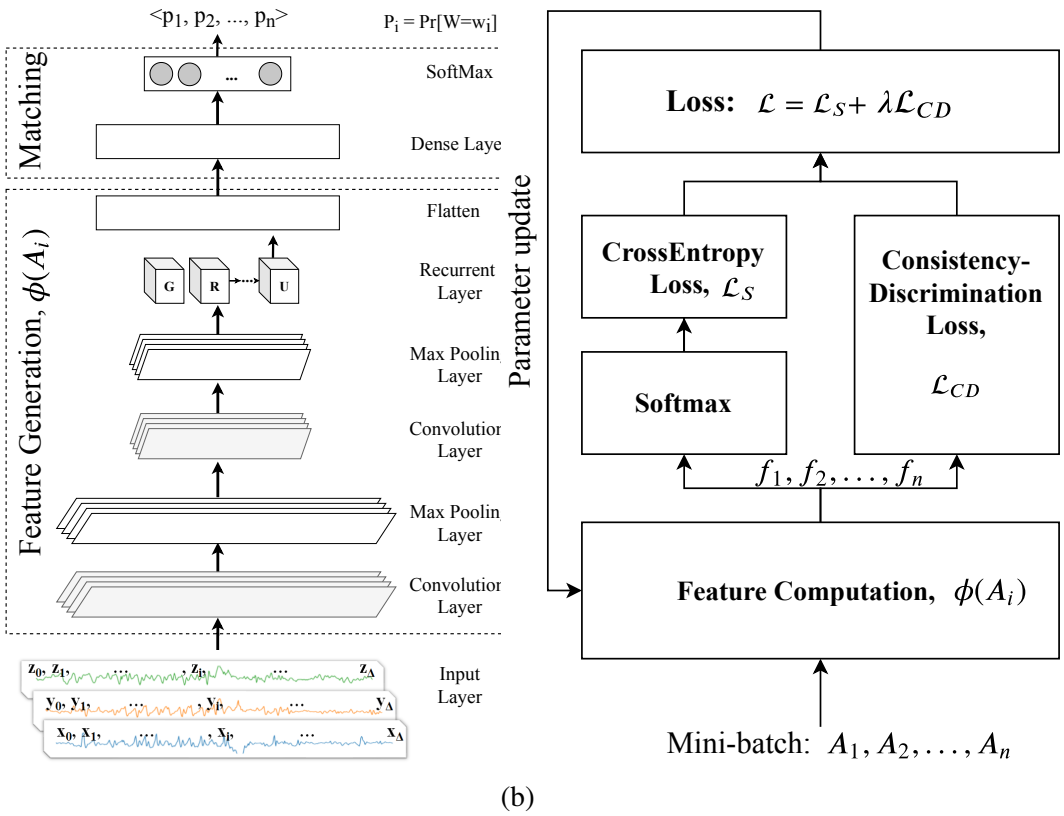


Fig. 4.5: (a) The convolutional-recurrent-deep network architecture for user re-identification experiments using accelerometry segment. At first, model generates deep feature from raw accelerometry data segment and then by fully connected layer it classifies wearer, (b) The proposed framework for loss computation.

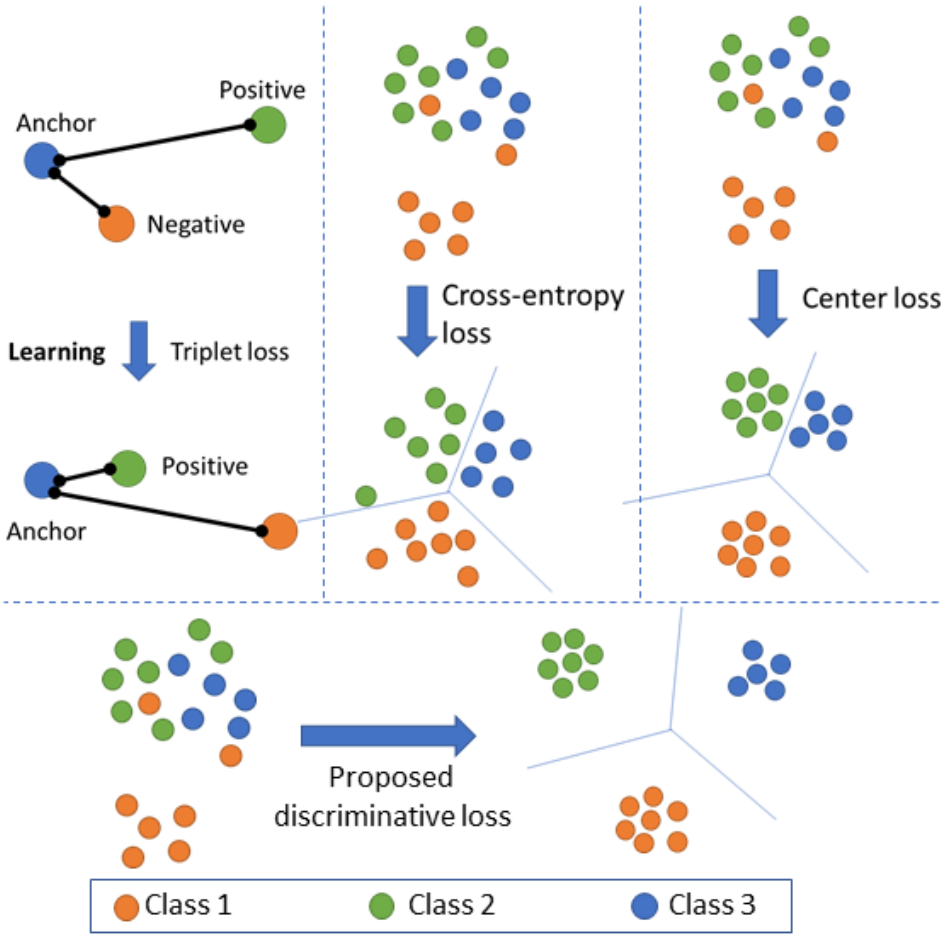


Fig. 4.6: The Triplet Loss minimizes the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity. Cross-entropy loss minimizes minimizes the false detection by the model. Center loss minimizes distance between any two points of same class. Proposed discriminative loss minimizes intra-class differences and maximizes inter-class differences.

max-pooling layers (followed by each convolution layer), one recurrent layer, one fully connected layer, and one softmax layer. Accelerometry segments from wearable sensors are first processed by two convolutional layers to learn micro features from the raw sensor data such as wrist movement or rotation. Next block in the pipeline is a Gated Recurrent Unit (GRU) to capture temporal patterns of the micro-feature sequence. Third block in the pipeline is a fully connected Deep Neural Network (DNN) layer to generate a classification score. Finally, the last layer's output is passed through a softmax function to generate the likelihood of each class. We describe details of these blocks in the following.

4.4.1 Model Architecture

Convolution layer

This is the first layer to operate on the input data segment, s_x^i. We use 1-D convolution in each convolution layer. The convolution layer defines a filter (or also called feature detector) with a kernel size. Only defining one filter would allow the neural network to learn one single feature in this convolution layer. We define n_k filters that allow us to train n_k different features on this convolution layer of the network. Since the input shape of this layer is Δx3, therefore, the output of the first neural network layer is a Δxn_k neuron matrix. Each column of the output matrix holds the weights of one single filter. With the defined kernel size and considering the length of the input matrix, each filter will contain Δ weights.

It extracts useful features by convolving the input feature map with different filters. let o_k be the output vector of the feature/kernel $k; 1 \leq k \leq n_k$. Since input channel is 3 and length of each kernel is Δ_k, therefore, shape of weight of the kth is Δ_kx3. Let $w_k(c)$ be the weight vector of cth channel of kth kernel. Let $a(c) =< a_1^c, a_2^c, ..., a_n^c >$ be the input vector of channel c. Then,

$$o_k = \sigma\left(b_k + \sum_{c=1}^{3} a(c) * w_k(c)\right)$$

Where σ is the activation function, we use Rectified linear unit (ReLU) for activation function, b_j is the bias term for the jth feature map and $a * b$ is the convolution operator.

1D Convolution Let a be our input vector and w be our kernel, and n is the length of a and n_k is the length of w. The convolution $a * w$ of a and w is defined as:

$$(a * w)(i) = \sum_{j=1}^{n_k} w(j).a(i - j + n_k/2)$$

Pooling layer

A pooling layer is a form of nonlinear subsampling which is used to reduce the complexity of the output and prevent overfitting of the data. In our proposed architecture, we chose a size of two. That means the size of the output matrix of this layer is half of the input matrix.

Gated Recurrent Unit (GRU) [1]

GRU units are modified version of the standard RNN units. GRU (Gated Recurrent Unit) aims to solve the vanishing gradient problem which comes with a standard recurrent neural network. To solve the vanishing gradient problem of a standard RNN, GRU uses, so called, update gate and reset gate. Basically, these are two vectors which decide what information should be passed to the output. The special thing about them is that they can be trained to keep information from long ago, without washing it through time or remove information which is irrelevant to the prediction. To solve the gradient exploding problem, we use gradient clipping. Gradient clipping involves forcing the gradient values (element-wise) to a specific minimum or maximum value if the gradient exceeded an expected range.

The *update gate* helps the model to determine how much of the past information (from previous time steps) needs to be passed along to the future. We start with calculating *the update gate* z_t for time step t using the formula:

$$z_t = \sigma(W_{hz}h_{t-1} + W_{xz}x_t + b_z)$$

When x_t is plugged into the network unit, it is multiplied by its own weight W_{xz}. The same goes for $h_(t-1)$ which holds the information for the previous $t-1$ units and is multiplied by its own weight W_{hz}. Both results are added together and a sigmoid activation function, $\sigma(.)$ is applied to squash the result between 0 and 1.

The *reset gate* is used from the model to decide how much of the past information

to forget. To calculate it, we use:

$$r_t = \sigma(W_{hr}h_{t-1} + W_{xr}x_t + b_r)$$

This formula is the same as the one for the update gate. The difference comes in the weights and the gate's usage.

Let's see how exactly the gates will affect the final output. First, we start with the usage of the reset gate. We introduce a new memory content which will use the reset gate to store the relevant information from the past. It is calculated as follows:

$$\tilde{h_t} = g(W_h x_t + U_h(r_t \odot h_{t-1} + b_h))$$

Where, $\odot$ represents Hadamard (element-wise) product.

As a last step, the network needs to calculate h_t vector which holds information for the current unit and passes it down to the network. In order to do that the update gate is needed. That is done as follows:

$$h_t = (1 - z_t)\dot{h}_{t-1} + z_t \odot \tilde{h_t}$$

Dropout layer

For regularization in the GRU layer we user a dropout layer that randomly selects a set of neurons and assigns 0 weights to the neurons in the network to make the network less sensitive to small variations in the data. Therefore it should further increase our accuracy on unseen data. We chose a rate of 0.5, which means that randomly 50% of the neurons will receive a zero weight. The output of this layer is the same as the input of this layer.

Fully connected layer with Softmax activation

The final layer will reduce the vector of height $(\frac{\Delta}{4} \times n_k)$ to a vector of n since we have n users thus n classes that we want to predict. This reduction is done by another matrix multiplication. *Softmax* is used as the activation function. It forces all n outputs of the neural network to sum up to one. The output value will, therefore, represent the probability for the input data segment s_x^i to belong to each of the n classes.

4.4.2 Proposed Loss Function

Key to training a deep learning model is the choice of an appropriate loss function. We propose a new loss function that can guide the deep learning model to discover a representation of the input data and an accompanying classifier that can extract commonality among the data segments belonging to the same user and maximize distinction from the data segments from all other users (including the unseen ones). We now describe how we construct our loss function, which we call the *consistency-distinction loss*.

Consistency is preserving the commonality of the signal from the same participant, and the *distinction* is amplifying the differences among different participants. Both are essential for open set classification task. We want to project raw accelerometry data to a feature space representation that the deep learning model can use to identify class boundaries satisfying both consistency and distinction. With the standard cross-entropy loss, the proposed architecture (in Section 4.4.1) ensures separation among different users/classes, but it does not guarantee consistency and distinction.

Our Consistency-Distinction loss (CD-loss) function builds upon commonly used Triplet Loss [100] and Center-Loss [101] functions. The Triplet Loss function seeks to maximize the separation among the classes (to amplify distinction), while the Center Loss function seeks to minimize the footprint of each class (to sharpen consistency). We first introduce the triplet loss and center loss, and then describe our proposed CD-loss loss function.

The Triplet Loss

The triplet loss [100, 102] is usually trained on a series of triplets (s_u^i, s_u^j, s_v^k), where s_u^i and s_u^j are data from the same user u, and s_v^k is from a different user v. The triplet loss is designed to keep s_u^i closer to s_u^j than s_v^k, and widely used in many areas, such as face recognition and person re-identification [100]. It is formulated as follows:

$$\mathcal{L}_{trp} = \sum_{(s_u^i, s_u^j, s_v^k)} \{\|\phi(s_u^i) - \phi(s_u^j)\| - \|\phi(s_u^i) - \phi(s_v^k)\| + \alpha\}$$

where, $\phi(s_u^i)$ is features of input s_u^i and the threshold α is a margin that is enforced between positive and negative pairs, ensuring that the minimum separation among different classes is at least α. In the above formulation, the triplet loss adopts the Euclidean distance to measure the similarity of extracted features from two sensor segments.

The Center Loss

For each iteration of training a deep learning model, the Center Loss [101] to be used in the current iteration is trained on a mini-batch consisting of m data segments (s_x^j) from $\mathbb{D}$, i.e. $\mathbb{D}_{MB} \subset \mathbb{D}$. The collection of s_x^j are randomly selected from $\mathbb{D}$ so $\mathbb{D}_{MB}$ can consist of any data segment from any user. The Center Loss functions seeks to minimize the intra-class variations. Using $\phi(\mathcal{S}_x)$ to denote deep features of all data segments from a user x, the Center Loss function is defined as follows

$$\mathcal{L}_C = \frac{1}{2} \sum_{s_x^j \in \mathbb{D}_{MB}} \|\phi(s_x^j) - \overline{(\phi(\mathcal{S}_x))}\|_2^2,$$

where $\overline{(\phi(\mathcal{S}_x))}$ is the centroid of deep features from Class x.

The Consistency-Distinction Loss Function

As described above, the Triplet Loss function can be used to maximize inter-class separation and the Center Loss function can be used to maximize the intra-class

consistency. But, our goal is to guide the deep learning model to achieve both distinction and consistency concurrently. Therefore, we need a new loss function that can simultaneously optimize both criteria. There are several challenges in developing such a composite loss function.

First, the input for both loss functions are different. The Triplet Loss function expects a triplet consisting of two data segments from the same user and the other data segment from another user in each training iteration. The Center Loss, on the other hand, expects a mini-batch randomly selected from all training data, without any preference for selecting data segments belonging to a common user. The second challenge is how to adapt the consistency metric so that the footprint of the classes are not disproportionately enlarged due to the presence of some outliers, as it may adversely impact the goal of maximizing the inter-class separation (including future classes, for new users). The final challenge is how to compose a new combined goal that prioritizes both consistency and distinction from the goals of the two loss functions, each of which has a diverse goal.

We first address the challenge of input mismatch of the two loss functions. Triplet loss selects triplets as input, but selecting tuples for triplets is difficult, and the performance and stability of the network depend on the correct order of the training set, which results in a weaker generalization capability. Instead of training the model as triplets, we train our model as mini-batch $\mathbb{D}_{MB} \subset \mathbb{D}$ in each iteration. We use the formulation of Triplet Loss when composing the overall loss function.

We now define the specific distance metric we use in our loss function. As described earlier, Neural network $\phi : \mathbb{D} \rightarrow \mathcal{F}$ computes deep features for each sensor segment, where $\mathcal{F}$ is the feature space and $f_u^i = \phi(s_u^i)$ is the computed deep feature vector of sensor segment s_u^i. Let feature space be a metric space with L^2-norm.

Recall that $\mathcal{S}_u$ contains all the sensor segment of user u. The distance between sensor segment s_x^i and a class of sensor segments $\mathcal{S}_u$ is defined by the average distance between s_x^i and all other elements of $\mathcal{S}_u$ in the feature space,

$$d(s_x^i, \mathcal{S}_u) = \frac{1}{|\mathcal{S}_u|} \sum_{s_u^j \in \mathcal{S}_u} \|\phi(s_x^i), \phi(s_u^j)\|_2^2$$

We use this definition of distance metric instead of the distance from Centroid used in the Center Loss function in order to reduce the number of model parameters. We now describe our definition of the consistency and distinction metric, before presenting our overall loss function.

Consistency (for Intra-class variation) of $\phi(s_u^i)$ is the average distance of point $\phi(s_u^i)$ from all other points $\phi(s_u^j)$ of same class/user in feature space F. More formally, consistency of point s_u^i is,

$$C(s_u^i) = d(s_u^i, \mathcal{S}_u)$$

Now consistency of the Class u is defined as an aggregated function of all the point consistencies in the class.

$$C_u = \psi\big(\{C(s_u^i)\}_{s_u^i \in \mathbb{D}_{MB}}\big)$$

We want this aggregated function to measure the sparsity of the class and not be susceptible to outliers (see the second challenge above). For this purpose, we can use a percentile measure for ψ. For our experiments, we use the 95th percentile of the point consistency values of a class. Finally, consistency is defined by the mean consistency of all the classes.

$$C = \frac{\sum_{u \in \mathcal{U}} C_u}{n}$$

Distinction (for Inter-class variation) of $\phi(s_u^i)$ is the distance of point ϕs_u^i from the closest class points in the feature space:

$$D(s_u^i) = \min_{v \in \mathcal{U}, v \neq u} d(s_u^i, \mathcal{S}_v)$$

Now the distinction is defined as the mean distinction of all the points.

$$D = \frac{\sum_{s_u^i \in \mathbb{D}_{MB}} D(s_u^i)}{|\mathbb{D}_{MB}|}$$

To address the third challenge of composing an overall loss function that can concurrently optimize both consistency and distinction, we formulate our loss function using a similar formulation of triplet loss (by replacing positive and negative distances with consistency and distinction, respectively). More specifically, we propose the *Consistency-Distinction Loss (CD-loss)* as follows

$$\mathcal{L}_{CD} = C - D + \alpha * C$$

We note that the deep learning model minimizes the value of loss function, resulting in minimizing consistency C and maximizing distinction D, until the value of D is at least α times the consistency. Here, α is a threshold on the ratio that is enforced between intra-class distance and inter-class distance. We note that our formulation differs from the Triplet Loss that uses α as a constant threshold on the difference in pairwise distances. We instead use α for ratio between the intra-class distance and inter-class distance because our loss function is not measuring distance between two points, but distance within and between two clusters.

For our proposed loss function to be acceptable in training of a deep learning model, we need to show that it is differentiable. Since our distance function is a sum of several distances and each distance is differentiable; therefore, the distance function is also differentiable. The gradient of $d(s_x^i, \mathcal{S}_u)$ with respect to point in feature space f_x^i is,

$$\frac{\partial}{\partial f_x^i} d(s_x^i, \mathcal{S}_u) = \frac{1}{|\mathcal{S}_u|} \sum_{s_u^j \in \mathcal{S}_u} (\phi(s_x^i) - \phi(s_u^j))$$

Since the proposed loss function $\mathcal{L}_{CD}$ is a linear combination of multiple

Algorithm 6 `MiniBatch-Loss,` $\mathcal{L}_{\mathcal{CD}}(\{(x_i, y_i)\}_{1 \leq i \leq n})$

Input: f_i: deep features
Output: Discriminative loss for this mini-batch
Initialize $\mathcal{L}_{\mathcal{D}} \leftarrow 0$
for $i = 1$ **to** n **do**
  $X_{same} = \bigcup_{x_j, y_j = y_i} \{x_j\}$
  $X_{diff} = \bigcup_{x_k, y_k \neq y_i} \{x_k\}$
  $\mathcal{L}_{\mathcal{CD}} = \mathcal{L}_{\mathcal{CD}} + |d(x_i, X_{same}) - d(x_i, X_{diff}) + \alpha|$
end for
return $\frac{\mathcal{L}_{\mathcal{CD}}}{n}$

differentiable functions thus the loss function is also differentiable. The gradients of $\mathcal{L}_{CD}$ with respect to f_x^i is computed as:

$$\frac{\partial \mathcal{L}_{CD}}{\partial f_x^i} = \frac{1}{m} \sum_{s_u^j \in \mathcal{S}_u} \left(\frac{\partial}{\partial f_u^j} d(s_u^j, \mathcal{S}_u) - \frac{\partial}{\partial f_u^j} d(s_u^j, \mathcal{S}_v) \right)$$

The Loss Function

We adopt the joint supervision of softmax loss and CD-loss to train our proposed neural network for discriminative feature learning. More specifically,

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_{CD},$$

where $\mathcal{L}_S$ is cross-entropy soft-max loss [103] and a scalar λ is used for balancing the two loss functions. Algorithm 6 summarizes learning steps in the base model with joint supervision.

4.5 Boosting Model and Optimal Data Quantization

We now describe the boosting model and optimization of the data segment length Δ that is used in the base model.

4.5.1 Boosting Method

The boosting methods use the user id's produced by the base model on each data segment of a given sensor trace to improve the re-identification performance. We present

Algorithm 7 `Majority Boosting,` $\mathcal{M}(M_\Delta, s_x)$

Input: s_x: sensor trace of l length
$M_\Delta \equiv (\phi, \varphi, \mathcal{T})$: base model that takes Δ length sensor segment
Output: user of s_x
$\langle s_x^1, ..., s_x^m \rangle = \texttt{Segment}(s_x, \Delta)$ $\triangleright\ m = \lfloor \frac{s_x}{\Delta} \rfloor$
Initialize $P \leftarrow \phi$
for $i = 1$ **to** m **do**
  $p_1^i, p_2^i, ..., p_n^i = \varphi(\phi(s_x^i))$
  $x = 0$ if max$(\{p_u^i\}_{u \in \mathcal{U}}) < \mathcal{T}$, otherwise $_{u \in \mathcal{U}} p_u^i$
  P.`add(x)`
end for
return `findMajority`(P)

two boosting methods: a) Majority boosting and b) MaxMean boosting, and compare their performance in experiments.

Majority boosting: Algorithm 7 shows the pseudocode of *Majority boosting* process. For a given test data sample s_x of length l, we first partition s_x into $m = \lfloor \frac{s_x}{\Delta} \rfloor$ segments where each segment is of length Δ. Second, each segment is fed as input to the base model M_Δ, resulting in a sequence of n likelihoods, one for each user $u \in \mathcal{U}$. From m segments, we obtain a $m \times n$ matrix of likelihoods P. Third, we replace each row with the user with the highest likelihood in that row, reducing the matrix of likelihoods in a vector with m most likely user id's. Finally, we report the majority prediction from these m user id's. In both steps of majority assignment, we break any ties randomly.

Lemma 3. *If M_Δ is a multi-class classifier with accuracy $\rho_\Delta = \left(\frac{1}{n} + \delta\right)$, where the number of classes is n and $\delta > 0$, then the re-identification accuracy $\rho(\rho_\Delta, m = \lfloor \frac{l}{\Delta} \rfloor)$ of $\mathcal{M}$ (Algorithm 7) on a test sensor data of length l is greater than $(1 - ne^{-m\delta^2/4})$. That is,*

$$\rho\left(\rho_\Delta, m = \frac{l}{\Delta}\right) > 1 - ne^{-m\delta^2/4} \tag{4.5}$$

Proof. Assume, that the correct user is v. Let X_u^i be random variables where for

$i = 1, 2, \ldots, m$ and $u = 0, 1, 2, \ldots, n$,

$$X_u^i = \begin{cases} 1, & \text{if the } i^{th} \text{ segment is classified as } u \\ 0, & \text{Otherwise} \end{cases}$$

Let random variable $X_u = \sum_{i=1}^{n} X_u^i$ be the random variable that represents the number of times the classification was u.

Now let Y_i^u be a random variable where for $u \neq v$,

$$Y_u^i = \begin{cases} 1, & \text{if the } i^{th} \text{ segment is classified as } v \\ -1, & \text{if the } i^{th} \text{ segment is classified as } u \\ 0, & \text{Otherwise} \end{cases}$$

Now, $Pr[Y_u^i = 1] = \rho_\Delta$ and $Pr[Y_u^i = -1] = \frac{1-\rho_\Delta}{n}$. Therefore, mean of Y_u^i = $1 \times \rho_\Delta + (-1) \times \frac{1-\rho_\Delta}{n}$ = $\frac{1}{n}(n\rho_\Delta - 1 + \rho_\Delta)$.

Now,

$$\begin{aligned} Pr[X_n \leq max(X_0, X_1, ..., X_{n-1})] &\leq \sum_{u=0}^{n-1} Pr[X_n \leq X_u] \\ &= \sum_{u=0}^{n-1} Pr[\sum_{i=1}^{m} Y_u^i \leq 0] \end{aligned}$$

Since mean of each Y_u^i is $\frac{1}{n}(n\rho_\Delta - 1 + \rho_\Delta)$, the mean μ of $\sum_{i=1}^{m} Y_u^i$ is $\frac{m}{n}(n\rho_\Delta - 1 + \rho_\Delta)$.

Let k be such that $(1-k)\mu = 0$, thus, $k = 1$. Using Chernoff bound we get,

$$
\begin{aligned}
Pr[\sum_{i=1}^{m} Y_u^i \leq 0] = Pr[\sum_{i=1}^{m} Y_u^i \leq (1-k)\mu] &\leq e^{-\frac{\mu^2 k^2}{m(1-(-1))^2}} \\
&= e^{-\frac{m^2}{4mn^2}(n\rho_\Delta - 1 + \rho_\Delta)^2} \\
&< e^{-\frac{m}{4n^2}(n\rho_\Delta - 1)^2} \\
&= e^{-m(\rho_\Delta - \frac{1}{n})^2/4} \\
&= e^{-m\delta^2/4}
\end{aligned}
$$

Therefore,

$$Pr[X_n \leq max(X_0, X_1, ..., X_{n-1})] < ne^{-m\delta^2/4}$$

Thus,

$$Pr[X_n > max(X_0, X_1, ..., X_{n-1})] \geq 1 - ne^{-m\delta^2/4}$$

□

Algorithm 8 `MaxMean Boosting,` $\mathcal{M}(M_\Delta, s_x)$

Input: s_x: sensor trace of l length
$M_\Delta \equiv (\phi, \varphi, \mathcal{T})$: base model that takes Δ length sensor segment
Output: user of s_x
$\langle s_x^1, ..., s_x^m \rangle = \texttt{Segment}(s_x, \Delta)$ ▷ $m = \lfloor \frac{s_x}{\Delta} \rfloor$
Initialize $P \leftarrow \phi$
for $i = 1$ **to** m **do**
  $\langle p_1^i, p_2^i, ..., p_n^i \rangle = \varphi(\phi(s_x^i))$
  $P.\texttt{add}(\langle p_1^i, p_2^i, ..., p_n^i \rangle)$
end for
$\langle \bar{p}_1, \bar{p}_2, ..., \bar{p}_n \rangle = \texttt{Mean}(P)$ ▷ $\bar{p_u} = \frac{1}{m}\sum_{i=1}^{m} p_u^i$
return 0 if max($\{\bar{p}_u\}_{u \in \mathcal{U}}$) $< \mathcal{T}$, otherwise $_{u \in \mathcal{U}} \bar{p}_u$

MaxMean boosting: Algorithm 8 shows the pseudocode of *MaxMean boosting* process. Instead of taking the majority from m detections by the base model, the MaxMean boosting method creates a likelihood vector of size n from the likelihood matrix P by computing the mean likelihood of each user. Finally, it outputs user id with

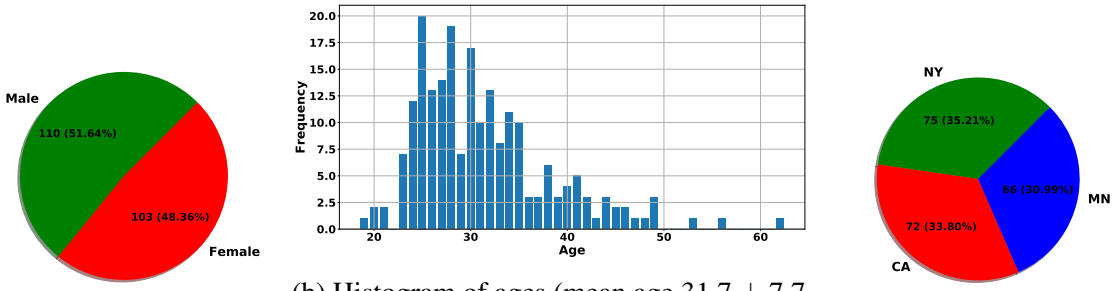
the maximum likelihood if it is greater than the decision threshold $\mathcal{T}$, and outputs 0, otherwise.

4.5.2 Selection of unit Length Δ

Recall that the unit length of the accelerometry trace (Δ) is the minimum amount of accelerometry data that is sufficient to identify both distinction of activity pattern from other users and consistency with other segments from the same user. The choice of $Delta$, therefore, is critical. The goal of data quantization problem is to find the optimal value of Δ. We note that the choice of Δ can have substantial impact on the re-identification performance.

First, the performance of the base model is expected to increase monotonically as we grow the value of . This is because if a smaller value of Δ_1 has a better performance than a higher value of $\Delta_2 (> \Delta_1)$, then the base model can locate the segment of length Δ_1 within the large segment of Δ_2 to achieve at least the same performance as that when provided a sub segment of length Δ_1. Intuitively, when the value of Δ is large (e.g., full day), it can capture different aspects of the user, identify uniqueness and consistency in daily pattern such as routines. Therefore, a larger size of Δ increases the accuracy of the base model.

However, for a fixed length l of test sample, as the value of Δ increases, the number of units that can be assessed by the base model decreases, reducing the opportunity to boost the re-identification performance by a boosting model. Hence, there is a trade-off between the value of Δ and the number of units of data assessed by the base model that can be used to boost the overall performance. As we show in experiments, the performance of the boosting model exhibits a convex function behavior, allowing us to select an optimal value of Δ for a given test length l.



(a) Gender profile

(b) Histogram of ages (mean age 31.7 ± 7.7 years)

(c) City profile

Fig. 4.7: Demographics of different participants

4.6 Evaluation

4.6.1 Dataset

Our goal is to analyze re-identification risks in motion sensor data collected in the users' natural environment. Even though the number of publicly available datasets of motion sensor data increases, they are still quite limited in how representative they are of the natural environment of the users. Most of the datasets contain data collected in a scripted setting (e.g., when performing a range of prescribed or selected activities such as exercise, walking, smoking, etc.). Hence, majority of research on person re-identification is limited to lab setup or a scripted dataset.

Our approach is to consider the full range of activities people perform in their natural living environment. Therefore, we used a newly collected dataset from a scientific study. This study was conducted to predict the work performance of employees using mobile sensors. The study was approved by the Institutional Review Board (IRB), and all participants provided written informed consent. The study consisted of 400 participants, each of whom wore wrist-worn sensors (consisting of 3-axis accelerometers sampled at 25 Hz for each axis and 3-axis gyroscopes sampled at 25 Hz for each axis) and carried a smartphone with data collection app installed. They collected data for at least 8 hours each day for 10 weeks (i.e., 70 days). Data collected included location, motion, app usage, privacy-preserving audio features, etc. The participants were knowledge-workers from diverse professions including management, information technology, education,

engineering, production, sales, transportation, etc., covering various posts from the CEO of a tech company to production personnel of a dairy product company.

Among the 400 participants, we excluded data from any participant who had less than 200 total minutes of usable quality physical activity data (due to data loss, data corruption, or metadata mismatch). As a result, we were able to use data collected from 353 participants (174 males, 123 females; mean age 31.7 $\pm$ 7.5 years).

For data processing, we first segment the accelerometry data into one-minute windows. We retain a minute if it contains at least 85% data of the original sample rate. To protect against data corruption or metadata mismatch from affecting our experiments, we also exclude those minutes, where the absolute value of any of the three accelerometer axes is greater than 4 (which is the maximum range of the sensor). To identify data segments that correspond to physically-active state of the user, we retain those minutes that have a standard deviation (from the magnitude of all three axes) of at least 0.21.

To have a uniform and sufficient amount of data from each participant in our experiments, we use 200 minutes of physically-active data from each participant. For greater ecological validity, we proportionately (with random selection) select minutes from each day so that the total number of minutes selected per participant is 200 minutes, and it represents the proportion of the day that the participant was physically active (while wearing the sensors) on each day. For example, if a participant had 1,000 minutes of usable quality physical activity data over 70 days of sensor wearing, and a specific day had 50 minutes of physically-active data, we randomly select 10 minutes from this day. In total, we use 70,600 minutes of accelerometry data for our experiments.

All the data storing and processing was done in an Apache PySpark based open-source platform for computational modeling of high-frequency data.

4.6.2 Experiment Setup

The performance evaluation of the open-set user re-identification task involves sensor traces from two groups of users: a) Known users ($\mathcal{U}$): users whose data are in the

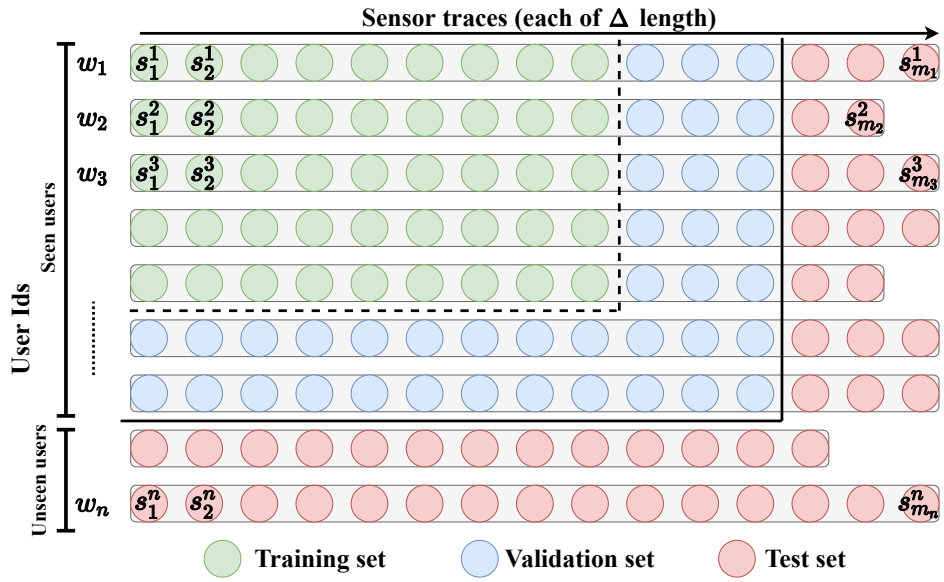


Fig. 4.8: Splitting of training, validation, and testing set. The dataset is first divided into training and testing sets, and then the training set is further divided into a fitting set and a validation set containing a closed set and an open set.

database $\mathcal{D}$ and b) Unknown users ($\mathcal{U}'$): whose data is not present in the database $\mathcal{D}$. Training set is constructed by sensor traces from known users and these data are used to train the base model. Testing set contains data from both the known users $\mathcal{U}$ but with sensor traces from different time and data from unknown users $\mathcal{U}'$.

First, we randomly select 80% of the users as $\mathcal{U}$ (282 out of 353 users) and the remaining 20% users as $\mathcal{U}'$. Then 2/3 of the sensor segments from each known user are selected as training data D_{train}. The rest of the 1/3 segments from each known user and all the sensor segments from unknown users form the testing set. The splitting process is shown in Figure 4.8. Since training deep learning is compute-intensive; therefore, for parameter learning such as experiment on finding optimal unit length, we use a smaller version of the validation dataset, namely $D_{100,50} \subset D_{train}$, with 100 known users and 50 unknown users.

4.6.3 Performance Metrics

Now we introduce two most commonly used evaluation metrics for open set recognition (OSR) used in our experiments. For evaluating classifiers in the OSR

scenario, a critical measure is the false recognition of Unknown users. Performance for open-set identification is characterized by two performance statistics — the detection and identification rate (DIR) and the false alarm rate (FAR).

Detection and Identification Rate (DIR):

A sensor trace from a known user $u \in \mathcal{U}$ is detected and identified if the model correctly identifies the user. This metric is also referred to as true matching rate (TMR).

$$TMR \text{ or } DIR = \frac{|\{(s_x, x) : M(s_x) = x \text{ where } x \in \mathcal{U}\}|}{|(s_x, x) : x \in \mathcal{U}|}$$

False Accept Rate (FAR)

The false accept rate occurs when a model detects one of the users in the database ($u \in \mathcal{U}$) for a sensor trace from an unknown user $x \notin \mathcal{U}$. Recall that $\mathcal{M}$ outputs 0 if it detects that the user of the test sample is not present in the database.

$$FAR = \frac{|(s_x, x) : \mathcal{M}(s_x) \neq 0 \text{ where } x \notin \mathcal{U}|}{|(s_x, x) : x \notin \mathcal{U}|}$$

There is a trade-off between TMR and FAR that is usually shown on a receiver operator characteristic (ROC).

Besides these two metrics, we use another metric that provides combined performance for known users and unknown users. Accuracy of the model $\mathcal{M}$ is defined as the percentage of correct classification by the model. Recall that M outputs 0 if it predicts user of the test sample is not present in the database.

$$\begin{aligned} Accuracy =& \frac{1}{2}\frac{|(s_x, x) : M(s_x) = x \text{ where } x \in \mathcal{U}|}{|(s_x, x) : x \in \mathcal{U}|} \\ &+ \frac{1}{2}\frac{|(s_x, x) : M(s_x) = 0 \text{ where } x \notin \mathcal{U}|}{|(s_x, x) : x \notin \mathcal{U}|} \\ =& \frac{1}{2}TMR + \frac{1}{2}(1 - FAR) \end{aligned}$$

4.6.4 Optimizing Unit Length (Δ)

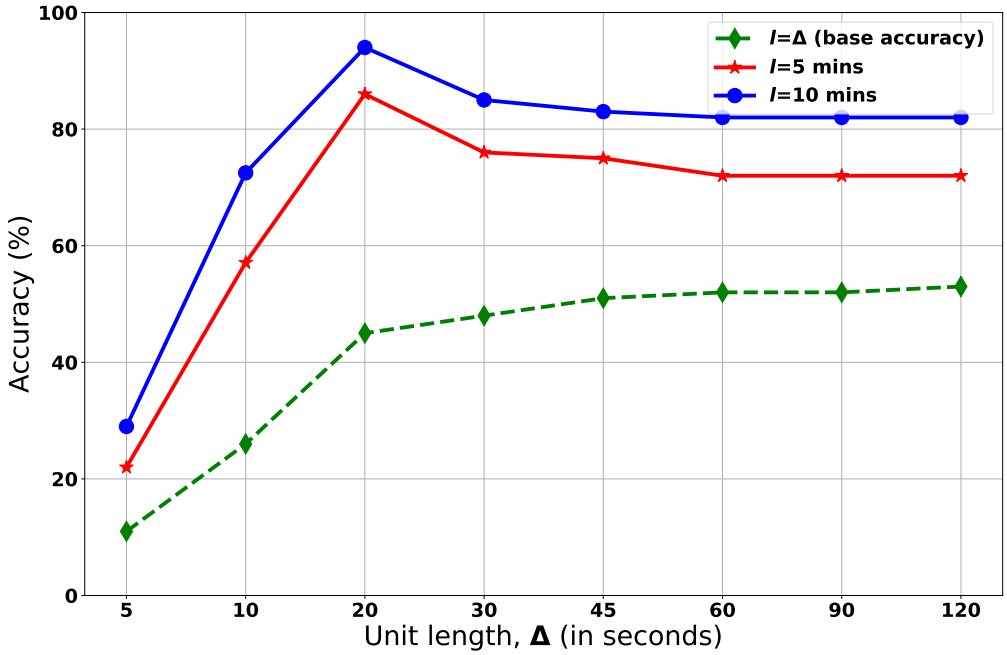


Fig. 4.9: Re-identification accuracy for different Δ. For $\Delta = 20$ seconds accuracy is maximum, therefore, we select 20 seconds as unit of sensor data. For training and validation we use $D_{100,50}$ dataset.

We start our experiments with finding the optimal unit length Δ. For this experiment, we use different values of Δ from a set $\{5, 10, 20, 30, 45, 60, 90, 120\}$ and observe the performance of the boosting model for test samples of length (l) 5 minutes and 10 minutes.

From Figure 4.9, we observe that the performance of the boosting model approximates a convex function as the value of Δ increases from 5 seconds to 120 seonds, peaking at $\Delta = 20$ seconds. Therefore, we select $\Delta = 20$ seconds as unit length for subsequent experiments.

4.6.5 Modeling Choices

Recall that our proposed base model consists of both convolutional and recurrent layers. In this experiment, we evaluate if a simpler model with only a) convolutional, or only b) recurrent layer can provide a comparable performance. The convolutional only

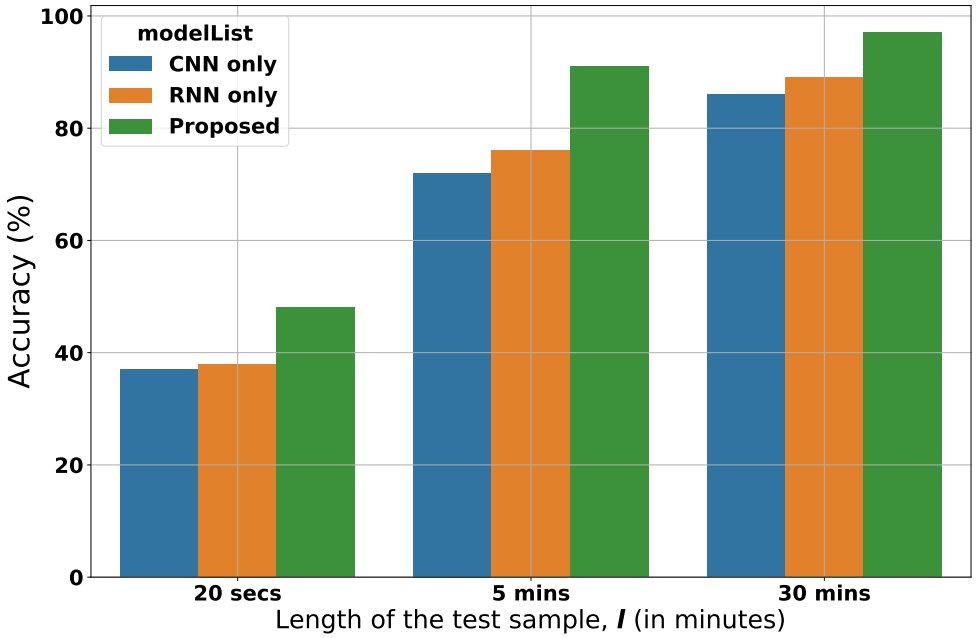


Fig. 4.10: Performance of our proposed model compared to only CNN layers and RNN layers

model consists of two convolutional layers, two max-pooling layers, and ReLu as an activation function. For the recurrent model, we use GRU units with a tanh activation function and 50% dropout.

Figure 4.10 shows that the accuracy of the proposed model is 10% greater than the convolutional or recurrent only models. Our model achieves 90% accuracy with only 5 minutes of test data. But, for CNN and RNN only models, 30 minutes of test sensor trace is required to achieve a similar accuracy.

Regarding the choice of boosting approach, both the Majority and MaxMean result in similar performance. Hence, for brevity we omit their detailed comparison.

4.6.6 Choice of The Decision Threshold

Both true matching rate (TMR) and false accepted rate (FAR) depend on the choice of threshold $\mathcal{T}$. Increasing $\mathcal{T}$ reduces FAR, but it also reduces TMR. On the other hand, decreasing $\mathcal{T}$ makes the system more tolerant to input variance improving TMR, but it also worsens FAR. Hence, there is a trade-off in selecting $\mathcal{T}$.

True matching of the true class is called genuine distribution and from other class

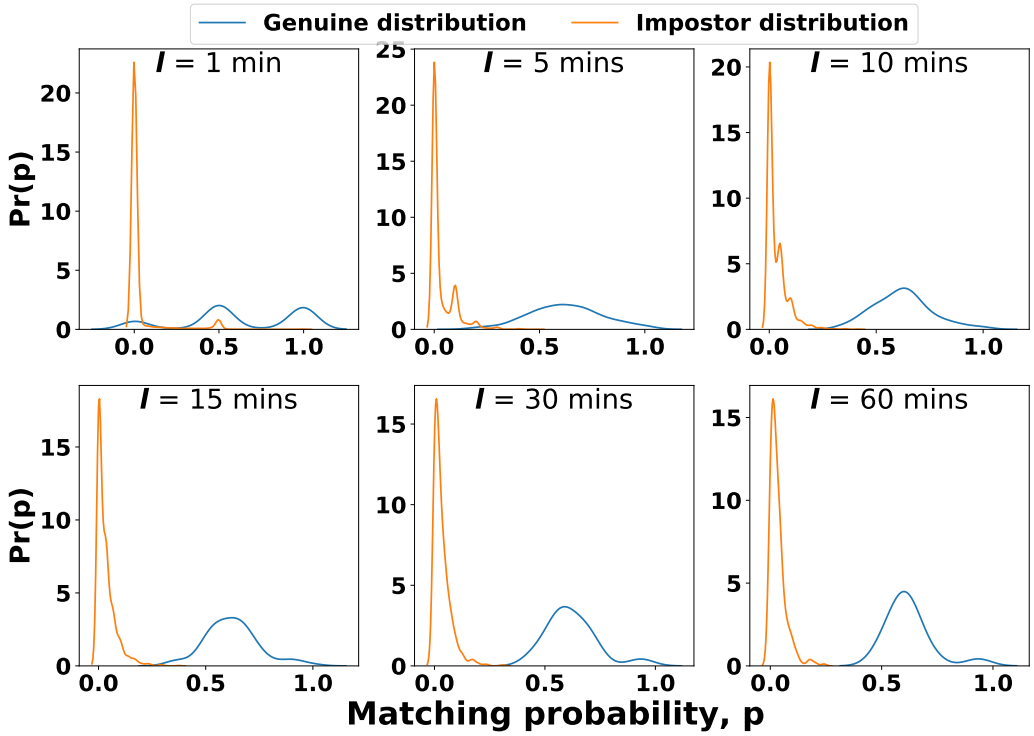


Fig. 4.11: Genuine and impostor distribution for different values of test length l. Decision threshold $\mathcal{T} = 0.35$ maximally separates both distributions in all the plots.

is called impostor distribution (see Figure 4.11). Decision threshold $\mathcal{T} = 0.35$ maximally separates both distributions in all the plots.

4.6.7 Effect of the Test Data Length

Figure 4.12 shows an ROC curve of boosting model for different values of the length of test data (l). For a test length of only 5 minutes, the model achieves a 90% TMR and 11% FAR. As we increase the test length to 30 minutes, the model achieves more than 95% TMR for an FAR of 0.1%. If the model is provided with test data of 60 minutes, the model achieves a TMR of 99.95% while keeping the FAR to 0.1%.

We also analyze the model performance for FAR of 100% that corresponds with a closed-set formulation (i.e., the model will always assign a known user id to any test data). Under this setting, the model achieves 92% TMR for test data length of 5 minutes and 99.7% for test data length of 30 minutes.

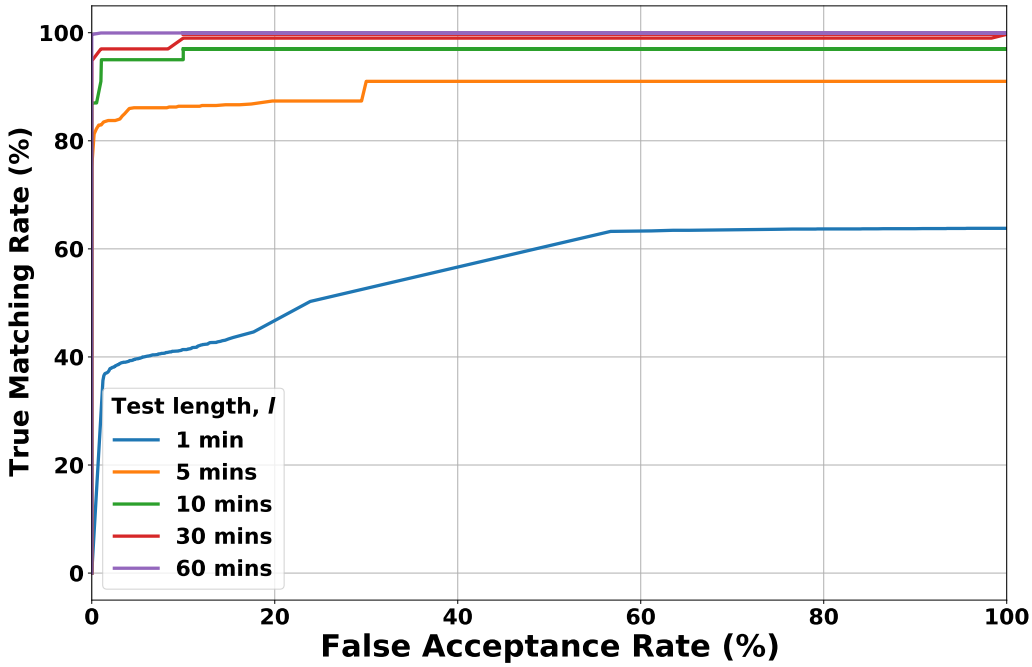


Fig. 4.12: ROC for Different Choices of Test Data Length

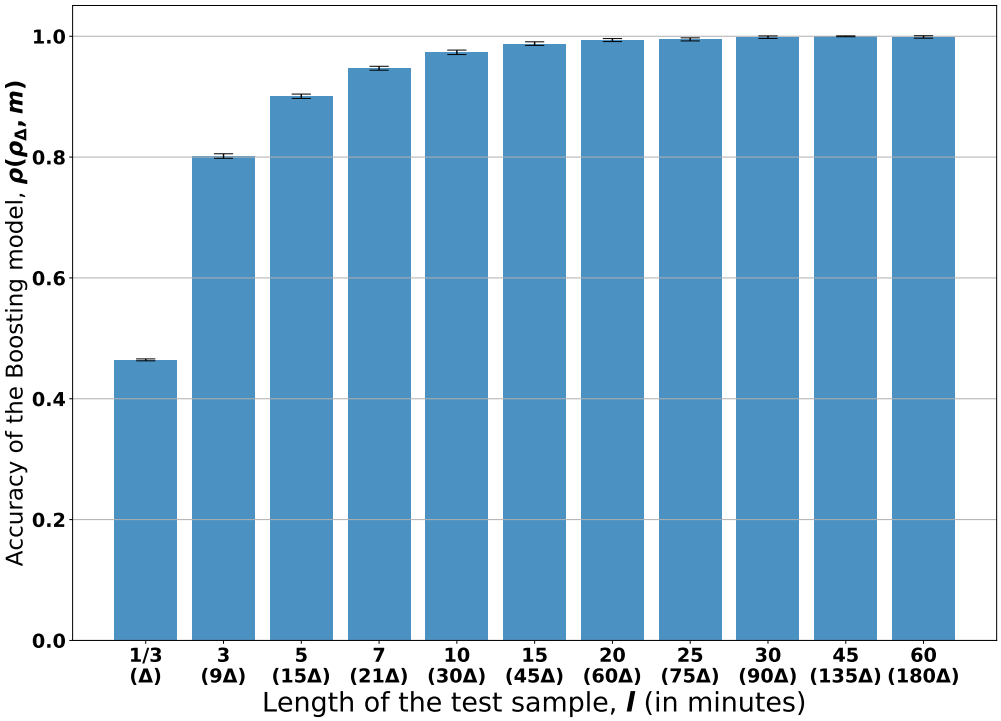


Fig. 4.13: Person re-identification results

Table 4.2: True Matching Rates (%) of the Boosting model for Different Test Data Lengths

Test length, l	FAR=0.1%	FAR=1%	FAR=10%	FAR=100%
1 min	0	0	42.56	62.04
5 mins	73.18	82.08	88.25	91.86
10 mins	87.26	91.53	95.58	97.04
30 mins	95.76	97.34	99.33	99.70
60 mins	99.95	99.95	99.95	99.97

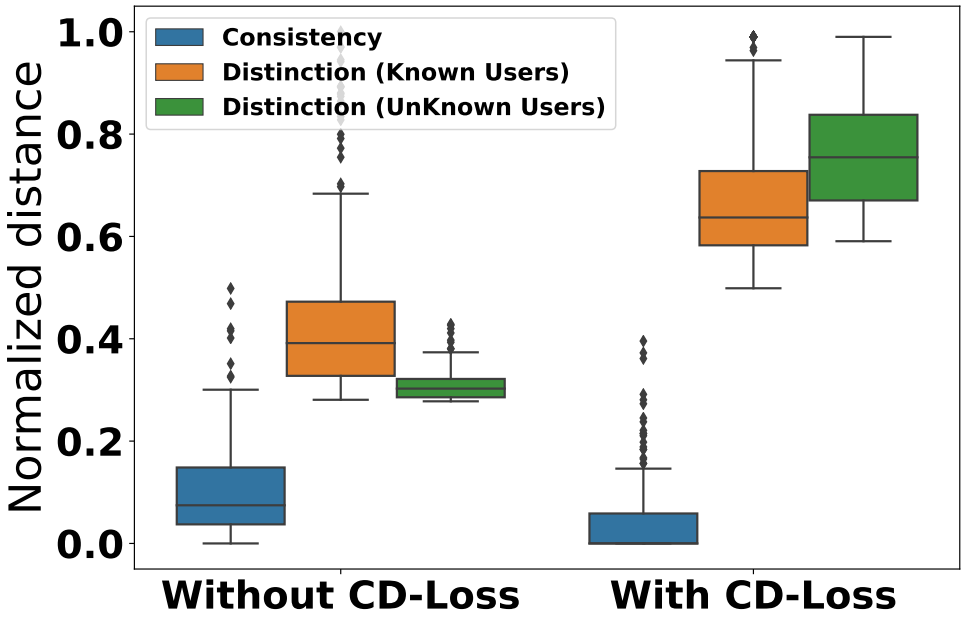


Fig. 4.14: The distribution of consistency and distinction in terms of normalized distance from model trained on both with CD-Loss and without CD-Loss.

4.6.8 Choice of the Loss Function

To evaluate the impact of our Consistency-Distinction (CD) loss function on model performance, we determine the intra-class spread and inter-class distance (from the closest class), both in the feature space. We normalize the set of both the distances to be in $[0, 1]$ for ease of visual comparison. The distribution of intra-class distances and inter-class distances from models trained with and without CD-loss (i.e., using only the cross-entropy loss) are shown in Figure 4.14. We observe that using our CD Loss function significantly reduces the intra-class distance (improving consistency) and widens the

inter-class distances, improving distinction. Additionally, we observe that the use of CD Loss results in better open set performance, i.e., reducing the footprint of known classes, reflected in greater distinction for unknown users.

4.6.9 Effect of Training Data

Table 4.3: True Matching Rate (%) for Different Training Data Lengths.

		Test length, l				
		20 secs	1 min	5 mins	10 mins	30 mins
Training Data (in minutes)	5	9.48	10.62	21.90	28.07	34.31
	15	12.29	13.74	35.42	48.10	71.56
	30	38.56	46.75	73.05	79.62	86.06
	60	47.46	55.24	87.96	92.22	97.56
	120	53.25	62.10	91.86	97.04	99.70

All results thus far used 130 minutes (after taking out 1/3 of the 200 minutes for testing) or more of training data. Table 4.3 presents the true matching rates if smaller training data is used. Further, we preceding results show, we need at least 5 minutes of test data for acceptable performance. We, therefore, present TMR for test data length from 5 minutes to 30 minutes and training data lengths from 30 minutes to 120 minutes. We observe that the model can achieve over 90% TMR using a training data of 60 minutes if the test length is 10 minutes. We also observe that using 120 minutes of training data provides 99.7% TMR, implying that two hours of training data is sufficient to get a high re-identification performance.

4.6.10 Performance of Demographic Information

We also evaluate the performance of detecting demographic information from the accelerometry traces. For gender detection, we consider that as a two-class classification (either male or female). The model achieves more than 90% accuracy with a test sample of length 30 minutes. For the age detection, we start with two age group (either greater than 30 or less than 30 since age 30 in the median age in our population). Since the base

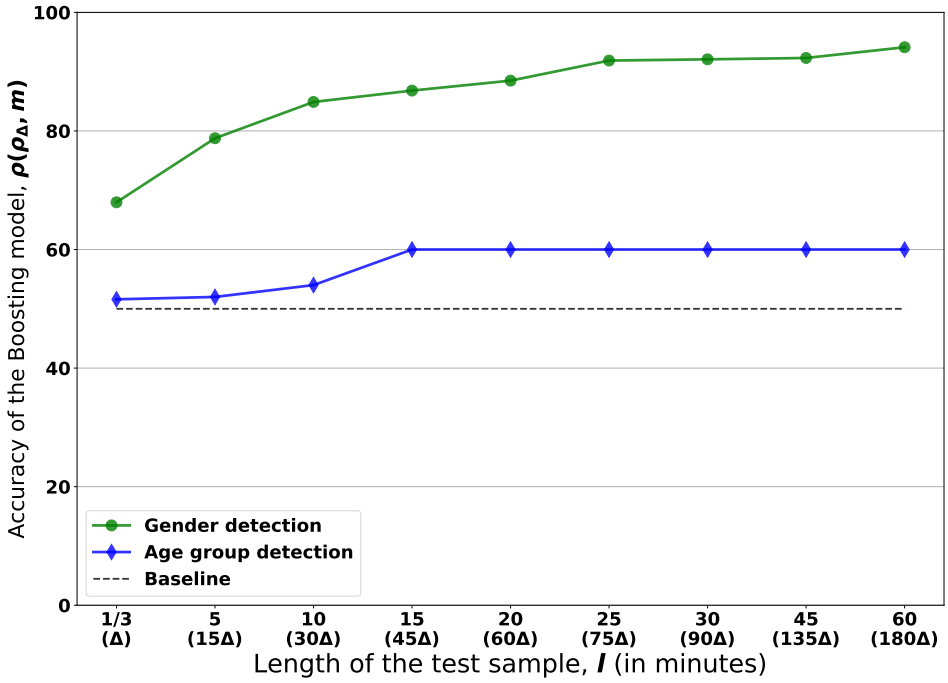


Fig. 4.15: Performance of demographic information extraction

model's performance is slightly better than a random guess, we observe a minimal improvement in boosting. Figure shows the robustness of the model to noise.

4.7 Mitigation of Re-identification Risk

Now we investigate if noise is introduced in the sensor data stream to reduce the risk of re-identification, what kind and what level of noise has how much impact in reducing the re-identification performance. We experiment with the natural choice Gaussian distribution and the Laplace distribution which is used to achieve differential privacy [104], i.e., $s_u(t) = s_u(t) + e_u(t)$ where $e_u(t) \sim N(0, \sigma)$ or $e_u(t) \sim Laplace(0, \lambda)$. Figure 4.16 shows that Laplacian noise is indeed more effective in reducing the re-identification performance.

Now we investigate the specificity of the model against random input. We generate a time series of accelerometry trace where each sample is taken from a Gaussian distribution, i.e., N(0, σ^2). We also observe Laplace noise's effect by taking a random

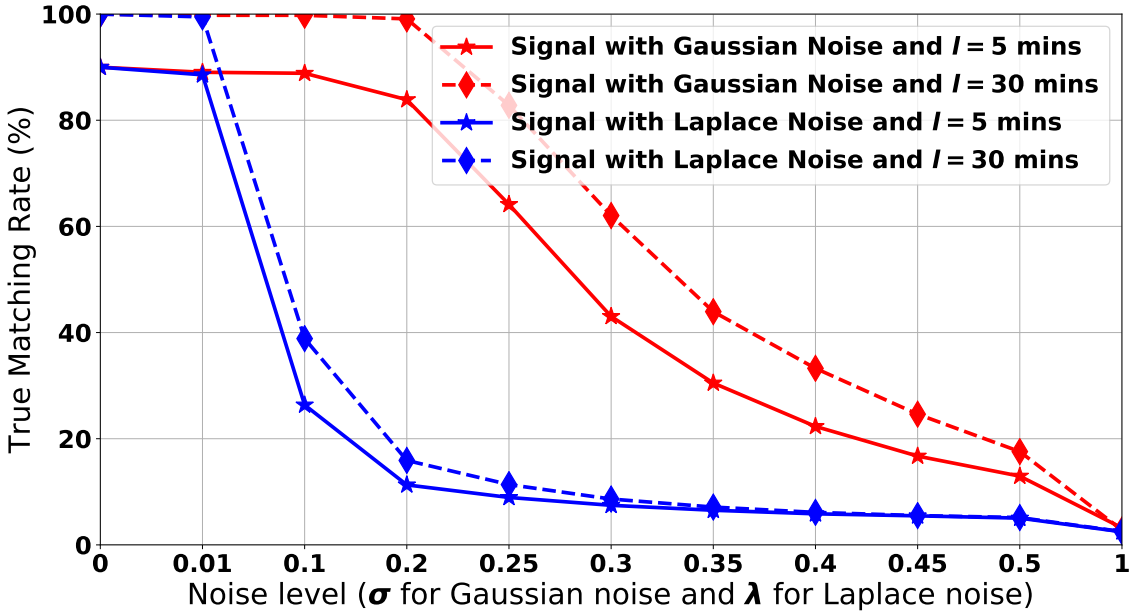
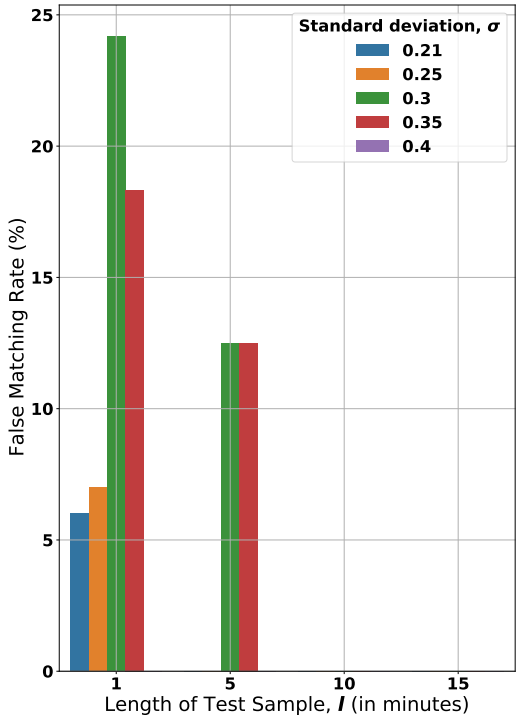
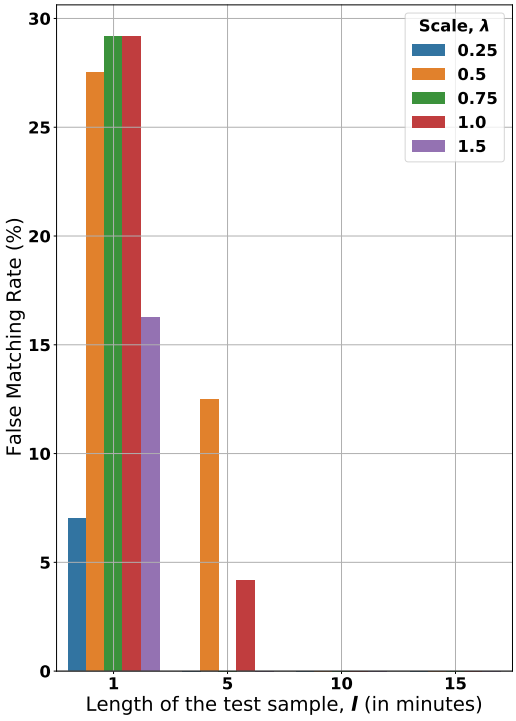


Fig. 4.16: Sensitivity of the model against noisy test signal



(a) Taking random samples from $N(0, \sigma^2)$.



(b) Taking random samples from $Laplace(0, \lambda)$.

Fig. 4.17: Effect of the random data.

sample from $Laplace(0, \lambda)$. We vary the value of σ and lambda from 0.20 to 0.40 with 0.05 increment and report the false accept rate in Figure 4.17.

We also vary the length of the test sample (l) and observe $l = 1$ 1 minute produces the highest false acceptance rate. For test length of 10 minutes or higher, the false acceptance rate is negligible. We, therefore, conclude that our model can identify and reject fictitious data of length 10 minutes or more.

4.8 Related Work

Person re-identification methods can be classified into two major categories, audio/visual and non-audio/visual. There has been extensive research in audio/visual methods for person, specifically vision-based methods. The main purpose of these methods is to authenticate a person via fingerprint, facial and iris/retina recognition to unlock a smartphone, authorize payment, etc. Other applications apart from authentication include criminal identification from CCTV footage, auto checkout in shops, automatic class/office attendance, etc.

Visual based methods learn appearance and texture related features. Zheng et. al. [105] developed an adaptive query system to classify good and bad features for image searching and person re-identification. Person re-identification problem can be called a special case of image searching where different types of distance functions are used to learn similarity learning. Chen et al. [106] imposed spatial constraints to solve the person variation problem using Mahalanobis distance function.

Person re-identification using visual features is still challenging due to intra-class variations with different input devices (cameras) [107]. But, advancements in Convolutional Neural Network (CNN) and generative deep learning models have boosted the accuracy of person re-identification.

A significant portion of person re-identification research focuses on metric learning loss, such as a combination of identification loss with verification loss [108, 109], and triplet loss [110–112] for models based on a variation of CNN.

Using a Generative adversarial network (GAN) for re-identification has been another active research direction. GAN has been used in [113–115] for person identification, where the generative and discriminative networks are separate from each other. Zheng et. al. [107] presented a joint learning method for both discriminative and generative methods for person re-identification. Despite all these improvements, visual-based person re-identification faces challenges due to frequent changes in appearance, camera calibration, occlusion, subject orientation, illumination, etc. Apart from these challenges, recent research [116, 117] shows that facial recognition algorithms are racially biased, i.e., the miss-identification rate of these algorithms are much higher for people from one or more specific races compared to others.

Nevertheless, vision-based methods have laid a strong foundation for person re-identification methods. Specifically, they have shown that deep learning architecture (to learn the latent state) coupled with an appropriate loss function (to measure the re-identification performance and guide the learning) can lead to high accuracy of re-identification. We use both of these lessons, due to significant difference in the nature of data, i.e., motion sensor data instead of image pixels and time series instead of a collection of pixels, requires innovations in the modeling architecture and development of a new loss function.

Speaker identification (from time-series of audio data) is another popular modality for person re-identification and authentication. Extensive research over a long time has resulted in several features from audio data that have personal identifying characteristics in vocal patterns, variations, and content. Kunz et. al. [118] use Hidden Markov Model (HMM) to verify users from real-time phone calls continuously. An et. al. [119] use a variation of CNN and Residual network with an attention mechanism for the same task. Due to significant existing domain knowledge, a high sampling frequency (thousands of hertz instead of tens of hertz for motion sensors), and occasional use of the speech content, small snippets of voice samples are usually sufficient for speaker identification.

But, due to low sampling frequency, lack of obvious identifying features, and free-living natural environment conditions, require the use of significantly larger data segments and subsequent novelty in processing them to achieve a good re-identification performance with wrist-worn motion sensors.

Recent studies show that audio/visual based person re-identification methods can be faked [120, 121], which is not as obvious yet for models based on sensor data from wearable devices or smartphones. Sensor data-based approaches can be grouped into two categories again, behavioral biometric approaches and device fingerprinting.

Behavioral biometric approaches have been used for user authentication in several research works. Examples include [122] and [123] that use hand waving detected from two different sources (accelerometer and ambient light sensor) to authenticate a user in smartphones. Others [124–128] feed keystroke biometrics and touchscreen interaction pattern (key pressed location, duration of keypress, size, drift, etc.) in different machine learning models to authenticate phone users. These methods are not directly applicable to person re-identification from wrist-worn accelerometer data because they rely on scripted settings (e.g., waving hand in a certain way or holding the phone in hand).

Another popular approach to behavioral biometric is gait-based authentication [94–97, 129]. These approaches extract the gait-based unique fingerprint from physical activities such as walking or running, using motion data from accelerometers placed on different body locations, sometimes supplemented with a video. Some recent research on gait based person identification use a variation of Deep Neural Networks with high accuracy [130]. These works establish the feasibility of extracting unique characteristics of the user from their motion pattern. But, the applicability of these methods is limited due to their restricted data collection methods, i.e., multiple sensors placed in different parts of the human body. Additionally, their methods are mostly trained to learn a similarity function that measures matching scores of two templates given the condition that the user performed a specific activity. Therefore, none of the existing

behavioral biometric solutions show the feasibility of person re-identification from wrist-worn accelerometery data collected from the natural environment.

Another complementary body of work seeks to re-identify a device (and subsequently a user, if the device is not shared among multiple users and until the user changes the device, e.g., upgrades their phone). These works, referred to as device fingerprinting, aim to generate a unique signature, or fingerprint, that uniquely recognizes a specific device. Several works find the fingerprint by extracting statistical features and using supervised machine learning approaches when the phone vibrates (for example, during an incoming call or message) [93] or when stationary [131, 132]. These methods were found to have an F1 score of 60% in field setting when devices are held in hand.

Bojinov et. al. [91] models the imprecision in accelerometer calibration via a device-specific scaling and translation of the measured values. For analysis, they collected data when the device was stationary, achieving a re-identification rate of 53% for devices in their dataset. More recently, [92] estimated the calibration matrix more accurately by considering all three errors: scaling factor, bias, and non-orthogonality misalignment errors. All of these methods model the error of the sensor due to the hardware imperfections during the sensor manufacturing process. Our work is complementary to these works as we seek to extract distinctive and unique features from the micro-movements patterns of a user when they are physically active.

Database linkage attack is a common method used by adversaries to de-anonymize publicly available anonymous data. Previous work in the area of de-anonymization was mostly focusing on correlating information of auxiliary data (sometimes referred to as side-channel information) from several independent data sets (datasets from different sources) or re-identifying data records [133–137]. Narayanan and Shmatikov used side-channel information (e.g., zip code, age, and gender of users) to de-anonymize the Netflix dataset [133]. Several approaches, such as $k-$anonymity [11, 138], l-diversity [12], and $t-$closeness [13], have been developed to make the database

anonymous and to prevent linkage attacks. These methods perturbed the database such that each row of the database cannot be uniquely identifiable that is indistinguishable from at least $k-1$ other data rows. In our scenario, we consider the released open dataset to be anonymized, and further linkage attack is not possible using attributes of the users.

4.9 Limitations and Discussion

Although our *WristPrint* method achieves 99.70% re-identification rate, there are several limitations to the presented work that open up numerous opportunities for future research.

First, in our dataset, each user's data came from the same device. Different wrist-worn devices differ in sampling rates, sensitivity range, mounting orientations, etc. This work did not experiment with these variations and hence their impact on re-identification performance can be investigated in future works. More specifically, a higher sampling rate and lower noise of the signal may allow the model to capture finer-grained micro-movements, potentially improving re-identification performance and reducing the amount of data needed for training and testing for a specified level of performance. Future work can also investigate the case when the model is trained on data from one device but tested on another device.

Second, for this analysis, we only looked at the wrist-worn device. Motion sensors are included in wearable devices such as earbuds and smart eyeglasses that are worn on different body locations. Future work can investigate the suitability of the presented modeling approach for re-identification using motion data collected from such devices.

Third, our experiments show that the distinctive features of the user's wrist movement remain consistent for ten weeks. Future work can investigate the deterioration in re-identification performance over time as user's movement patterns evolve, especially after major events such as accidents, pregnancy, and job changes, among several others.

Fourth, our experiments show that the impact of segmentation length choice on re-identification performance exhibits a convex shape, displaying unique optimal value for

a given test data length (see Figure 4.9). Future work can develop theoretical frameworks to prove such a property and derive optimal values analytically. Fifth, we experimented on 353 users to test the suitability of our open-set approach to modeling. Future works can investigate how well our approach can generalize to a significantly larger number of users.

Finally, we show that adding noise reduces re-identification risk. Future work can investigate what level of noise can still retain the intended utility of the dataset.

4.10 Conclusions

Several modalities of data are routinely used for user re-identification and sometimes even for authentication. They include video, voice, and fingerprints. But, new modalities of data are emerging that capture users' movement patterns at very fine granularity. Wrist-worn devices have emerged as one such increasingly popular device. To support research for new inferences of daily behaviors from these devices, data collected from user studies are publicly shared assuming lack of any identifying information embedded in them. Our work shows that data collected from such devices, even at 25 HZ can support user re-identification with 99.7% accuracy. This creates both new opportunities and raises new research, privacy, and ethical challenges for the community.

REFERENCES

[1] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[2] N. Saleheen, A. A. Ali, S. M. Hossain, H. Sarker, S. Chatterjee, B. Marlin, E. Ertin, M. al'Absi, and S. Kumar, "puffmarker: a multi-sensor approach for pinpointing the timing of first lapse in smoking cessation," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2015, pp. 999–1010.

[3] A. A. Ali, S. M. Hossain, K. Hovsepian, M. M. Rahman, K. Plarre, and S. Kumar, "mpuff: automated detection of cigarette smoking puffs from respiration measurements," in *Proceedings of the 11th international conference on Information Processing in Sensor Networks*. ACM, 2012, pp. 269–280.

[4] E. Thomaz, I. Essa, and G. D. Abowd, "A practical approach for recognizing eating moments with wrist-mounted inertial sensing," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2015, pp. 1029–1040.

[5] S. Akther, N. Saleheen, S. A. Samiei, V. Shetty, E. Ertin, and S. Kumar, "moral: An mhealth model for inferring oral hygiene behaviors in-the-wild using wrist-worn inertial sensors," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 1, p. 1, 2019.

[6] K. Hovsepian, M. al'Absi, E. Ertin, T. Kamarck, M. Nakajima, and S. Kumar, "cstress: towards a gold standard for continuous stress assessment in the mobile environment," in *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*. ACM, 2015, pp. 493–504.

[7] S. M. Hossain, A. A. Ali, M. M. Rahman, E. Ertin, D. Epstein, A. Kennedy, K. Preston, A. Umbricht, Y. Chen, and S. Kumar, "Identifying drug (cocaine) intake events from acute physiological response in the presence of free-living physical activity," in *Proceedings of the 13th international symposium on Information processing in sensor networks*. IEEE Press, 2014, pp. 71–82.

[8] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[9] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE Pervasive Computing*, vol. 16, no. 4, pp. 62–74, 2017.

[10] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Protecting sensory data against sensitive inferences," in *Proceedings of the 1st Workshop on Privacy by Design in Distributed Systems*. ACM, 2018, p. 2.

[11] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[12] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, pp. 3–es, 2007.

[13] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 2007, pp. 106–115.

[14] "Smoking-attributable mortality, years of potential life lost, and productivity losses - United States, 2000 - 2004," *Morbidity and Mortality Weekly Report*, vol. 57, no. 45, pp. 1226–1228, 2008.

[15] A. Mokdad, J. Marks, D. Stroup, and J. Gerberding, "Actual causes of death in the united states, 2000," *The Journal of the Americal Medical Association*, vol. 291, no. 10, pp. 1238–1245, 2004.

[16] S. A. Spohr, R. Nandy, D. Gandhiraj, A. Vemulapalli, S. Anne, and S. T. Walters, "Efficacy of sms text message interventions for smoking cessation: A meta-analysis," *Journal of Substance Abuse Treatment*, 2015.

[17] H. Brendryen, P. Kraft, and H. Schaalma, "Looking inside the black box: Using intervention mapping to describe the development of the automated smoking cessation intervention'happy ending'," *The Journal of Smoking Cessation*, vol. 5, no. 1, pp. 29–56, 2010.

[18] N. Hymowitz, M. Sexton, J. Ockene, and G. Grandits, "Baseline factors associated with smoking cessation and relapse," *Preventive Medicine*, vol. 20, no. 5, pp. 590–601, 1991.

[19] K. Doherty, T. Kinnunen, F. Militello, and A. Garvey, "Urges to smoke during the first month of abstinence: relationship to relapse and predictors," *Psychopharmacology*, vol. 119, no. 2, pp. 171–178, 1995.

[20] J. Hughes and D. Hatsukami, "Signs and symptoms of tobacco withdrawal," *Archives of General Psychiatry*, vol. 43, no. 3, pp. 289–294, 1986.

[21] J. Killen and S. Fortmann, "Craving is associated with smoking relapse: Findings from three prospective studies," *Experimental and Clinical Psychopharmacology*, vol. 5, no. 2, pp. 137–142, 1997.

[22] K. Matheny and K. Weatherman, "Predictors of smoking cessation and maintenance," *Journal of Clinical Psychology*, vol. 54, no. 2, pp. 223–235, 1998.

[23] S. Shiffman, "Reflections on smoking relapse research," *Drug and Alcohol Review*, vol. 25, no. 1, pp. 15–20, 2006.

[24] S. Shiffman, J. Paty, M. Gnys, J. Kassel, and M. Hickcox, "First lapses to smoking: Within-subjects analysis of real-time reports," *Journal of Consulting and Clinical Psychology*, vol. 64, no. 2, pp. 366–379, 1996.

[25] M. Stitzer and J. Gross, "Smoking relapse: the role of pharmacological and behavioral factors," *Progress in Clinical and Biological Research*, vol. 261, pp. 163–184, 1988.

[26] G. Swan, M. Ward, and L. Jack, "Abstinence effects as predictors of 28-day relapse in smokers," *Addictive Behaviors*, vol. 21, no. 4, pp. 481–490, 1996.

[27] S. Shiffman and A. Waters, "Negative affect and smoking lapses: A prospective analysis," *Journal of Consulting and Clinical Psychology*, vol. 72, no. 2, pp. 192–201, 2004.

[28] H. Ashton, D. Watson, R. Marsh, and J. Sadler, "Puffing frequency and nicotine intake in cigarette smokers," *The British Medical Journal*, pp. 679–681, 1970.

[29] D. Kalman, "The subjective effects of nicotine: methodological issues, a review of experimental studies, and recommendations for future research," *Nicotine & Tobacco Research*, vol. 4, no. 1, pp. 25–70, 2002.

[30] S. Shiffman, D. Scharf, W. Shadel, C. Gwaltney, Q. Dang, S. Paton, and D. Clark, "Analyzing milestones in smoking cessation: illustration in a nicotine patch trial in adult smokers," *Journal of Consulting and Clinical Psychology*, vol. 74, no. 2, pp. 276–285, 2006.

[31] A. Parate, M.-C. Chiu, C. Chadowitz, D. Ganesan, and E. Kalogerakis, "RisQ: Recognizing smoking gestures with inertial sensors on a wristband," in *Proc. ACM MobiSys*, 2014.

[32] Q. Tang, D. J. Vidrine, E. Crowder, and S. S. Intille, "Automated detection of puffing and smoking with wrist accelerometers," in *Proc. Pervasive Health)*, 2014.

[33] A. Ali, S. Hossain, K. Hovsepian, M. Rahman, K. Plarre, and S. Kumar, "mPuff: Automated detection of cigarette smoking puffs from respiration measurements," in *Proc. ACM IPSN*, 2012.

[34] P. Lopez-Meyer, S. Tiffany, and E. Sazonov, "Identification of cigarette smoke inhalations from wearable sensor data using a support vector machine classifier," in *Proc. IEEE EMBC*, 2012.

[35] J. Palmer and K. Hiiemae, "Eating and breathing: interactions between respiration and feeding on solid food," *Dysphagia*, vol. 18, no. 3, pp. 169–178, 2003.

[36] P. M. Scholl, N. Kücükyildiz, and K. V. Laerhoven, "When do you light a fire?: Capturing tobacco use with situated, wearable sensors," in *Proc. ACM UbiComp Workshop on Human Factors and Activity Recognition in Healthcare, Wellness, and Assistend Living*, 2013.

[37] P. Wu, J. Hsieh, J. Cheng, S. Cheng, and S. Tseng, "Human smoking event detection using visual interaction clues," in *Proc. IEEE Int'l Conf. Pattern Recognition*, 2010.

[38] E. Ertin, N. Stohs, S. Kumar, A. Raij, M. al'Absi, and S. Shah, "Autosense: Unobtrusively wearable sensor suite for inferring the onset, causality, and consequences of stress in the field," in *Proc. of ACM SenSys*, 2011, pp. 274–287.

[39] M. Rahman, R. Bari, A. Ali, M. Sharmin, A. Raij, K. Hovsepian, and et. al. "Are we there yet?: Feasibility of continuous stress assessment via wireless physiological sensors," in *Proc. of ACM BCB*, 2014, pp. 479–488.

[40] J. Murphy, *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. New York Institute of Finance, 1999.

[41] C. Marian, R. J. O'Connor, M. V. Djordjevic, V. W. Rees, D. K. Hatsukami, and P. G. Shields, "Reconciling human smoking behavior and machine smoking patterns: implications for understanding smoking behavior and the impact on laboratory studies," *Cancer Epidemiology Biomarkers & Prevention*, vol. 18, no. 12, pp. 3305–3320, 2009.

[42] M. Pedley. (2014) Tilt sensing using a three-axis accelerometer. [Online]. Available: http://www.freescale.com/files/sensors/doc/app_note/AN3461.pdf

[43] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive computing*. Springer, 2004, pp. 1–17.

[44] M. Rahman, A. A. Ali, K. Plarre, M. Absi, E. Ertin, and S. Kumar, "mconverse : Inferring conversation episodes from respiratory measurements collected in the field," *Wireless Health*, 2011.

[45] K. Plarre, A. Raij, S. M. Hossain, A. A. Ali, M. Nakajima, M. al'Absi, E. Ertin, T. Kamarck, S. Kumar, M. Scott, *et al.*, "Continuous inference of psychological stress from sensory measurements collected in the natural environment," in *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*. IEEE, 2011, pp. 97–108.

[46] K. Hovsepian, M. al'Absi, E. Ertin, T. Kamarck, M. Nakajima, and S. Kumar, "cstress: towards a gold standard for continuous stress assessment in the mobile environment," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2015, pp. 493–504.

[47] E. Thomaz, I. Essa, and G. D. Abowd, "A practical approach for recognizing eating moments with wrist-mounted inertial sensing," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2015, pp. 1029–1040.

[48] A. Parate, M.-C. Chiu, C. Chadowitz, D. Ganesan, and E. 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*. ACM, 2014, pp. 149–161.

[49] S. M. Hossain, A. A. Ali, M. M. Rahman, E. Ertin, D. Epstein, A. Kennedy, K. Preston, A. Umbricht, Y. Chen, and S. Kumar, "Identifying drug (cocaine) intake events from acute physiological response in the presence of free-living physical activity," in *Proceedings of the 13th international symposium on Information processing in sensor networks*. IEEE Press, 2014, pp. 71–82.

[50] G. D. Clifford, F. Azuaje, and P. McSharry, *Advanced methods and tools for ECG data analysis*. Artech House, Inc., 2006.

[51] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[52] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 3, 2007.

[53] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*. IEEE, 2007, pp. 106–115.

[54] C. Dwork, "Differential privacy: A survey of results," in *Theory and applications of models of computation*. Springer, 2008, pp. 1–19.

[55] M. Hay, V. Rastogi, G. Miklau, and D. Suciu, "Boosting the accuracy of differentially private histograms through consistency," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1021–1032, 2010.

[56] A. Roth and T. Roughgarden, "Interactive privacy via the median mechanism," in *Proceedings of the forty-second ACM symposium on Theory of computing*. ACM, 2010, pp. 765–774.

[57] A. Friedman and A. Schuster, "Data mining with differential privacy," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 493–502.

[58] Y. He, S. Barman, D. Wang, and J. F. Naughton, "On the complexity of privacy-preserving complex event processing," in *Proceedings of the Thirtieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ser. PODS '11, 2011, pp. 165–174.

[59] M. Götz, S. Nath, and J. Gehrke, "Maskit: Privately releasing user context streams for personalized mobile applications," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '12, 2012, pp. 289–300.

[60] S. Chakraborty, C. Shen, K. R. Raghavan, Y. Shoukry, M. Millar, and M. Srivastava, "ipShield: A Framework For Enforcing Context-Aware Privacy," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, 2014, pp. 143–156.

[61] S. Chakraborty, K. R. Raghavan, M. P. Johnson, and M. B. Srivastava, "A framework for context-aware privacy of sensor data on mobile systems," in *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*. ACM, 2013, p. 11.

[62] S. Chakraborty, "Balancing behavioral privacy and information utility in sensory data flows," Ph.D. dissertation, University of California, Los Angeles, 2014.

[63] L. Atallah, B. Lo, R. King, and G.-Z. Yang, "Sensor placement for activity detection using wearable accelerometers," in *2010 International Conference on Body Sensor Networks*. IEEE, 2010, pp. 24–29.

[64] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 439–450.

[65] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2001, pp. 247–255.

[66] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," *Information Systems*, vol. 29, no. 4, pp. 343–364, 2004.

[67] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2003, pp. 202–210.

[68] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography*. Springer, 2006, pp. 265–284.

[69] N. Li, W. H. Qardaji, and D. Su, "Provably private data anonymization: Or, k-anonymity meets differential privacy," *Arxiv preprint*, 2011.

[70] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta, "Discovering frequent patterns in sensitive data," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 503–512.

[71] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. Vadhan, "On the complexity of differentially private data release: efficient algorithms and hardness results," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM, 2009, pp. 381–390.

[72] Y. Xiao, L. Xiong, and C. Yuan, "Differentially private data release through multidimensional partitioning," in *Secure Data Management*. Springer, 2010, pp. 150–168.

[73] B. Pinkas, "Cryptographic techniques for privacy-preserving data mining," *ACM SIGKDD Explorations Newsletter*, vol. 4, no. 2, pp. 12–19, 2002.

[74] R. Chen, N. Mohammed, B. C. Fung, B. C. Desai, and L. Xiong, "Publishing set-valued data via differential privacy," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 1087–1098, 2011.

[75] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Tools for privacy preserving distributed data mining," *ACM Sigkdd Explorations Newsletter*, vol. 4, no. 2, pp. 28–34, 2002.

[76] W. Du and M. J. Atallah, "Secure multi-party computation problems and their applications: a review and open problems," in *Proceedings of the 2001 workshop on New security paradigms*. ACM, 2001, pp. 13–22.

[77] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *Advances in Cryptologyâ"CRYPTO 2000*. Springer, 2000, pp. 36–54.

[78] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 639–644.

[79] V. Bindschaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," in *2016 IEEE Symposium on Security and Privacy*. IEEE, 2016.

[80] L. Biel, O. Pettersson, L. Philipson, and P. Wide, "Ecg analysis: a new approach in human identification," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 3, pp. 808–812, 2001.

[81] M. Pagani, N. Montano, A. Porta, A. Malliani, F. M. Abboud, C. Birkett, and V. K. Somers, "Relationship between spectral components of cardiovascular variabilities and direct measures of muscle sympathetic nerve activity in humans," *Circulation*, vol. 95, no. 6, pp. 1441–1448, 1997.

[82] O. Amft, H. Junker, and G. Troster, "Detection of eating and drinking arm gestures using inertial body-worn sensors," in *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*. IEEE, 2005, pp. 160–163.

[83] G. Rigas, A. T. Tzallas, M. G. Tsipouras, P. Bougia, E. E. Tripoliti, D. Baga, D. I. Fotiadis, S. G. Tsouli, and S. Konitsiotis, "Assessment of tremor activity in the parkinson's disease

using a set of wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 3, pp. 478–487, 2012.

[84] D. R. Myers, A. Weiss, M. R. Rollins, and W. A. Lam, "Towards remote assessment and screening of acute abdominal pain using only a smartphone with native accelerometers," *Scientific reports*, vol. 7, no. 1, pp. 1–12, 2017.

[85] E. Davarci, B. Soysal, I. Erguler, S. O. Aydin, O. Dincer, and E. Anarim, "Age group detection using smartphone motion sensors," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 2201–2205.

[86] S. Cho, J. Park, and O. Kwon, "Gender differences in three dimensional gait analysis data from 98 healthy korean adults," *Clinical biomechanics*, vol. 19, no. 2, pp. 145–152, 2004.

[87] A. Jain and V. Kanhangad, "Investigating gender recognition in smartphones using accelerometer and gyroscope sensor readings," in *2016 international conference on computational techniques in information and communication technologies (ICCTICT)*. IEEE, 2016, pp. 597–602.

[88] H.-F. Yanai and A. Enjyoji, "Estimating carrier's height by accelerometer signals of a smartphone," in *International Conference on Human-Computer Interaction*. Springer, 2016, pp. 542–546.

[89] J. Hua, Z. Shen, and S. Zhong, "We can track you if you take the metro: Tracking metro riders using accelerometers on smartphones," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 286–297, 2016.

[90] D. He, S. Chan, and M. Guizani, "User privacy and data trustworthiness in mobile crowd sensing," *IEEE Wireless Communications*, vol. 22, no. 1, pp. 28–34, 2015.

[91] H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh, "Mobile device identification via sensor fingerprinting," *arXiv preprint arXiv:1408.1416*, 2014.

[92] J. Zhang, A. Beresford, and I. Sheret, "Sensorid: Sensor calibration fingerprinting for smartphones," 2019.

[93] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi, "Accelprint: Imperfections of accelerometers make smartphones trackable." in *NDSS*, 2014.

[94] A. Primo, V. V. Phoha, R. Kumar, and A. Serwadda, "Context-aware active authentication using smartphone accelerometer measurements," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 98–105.

[95] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Cell phone-based biometric identification," in *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*. IEEE, 2010, pp. 1–7.

[96] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S.-M. Makela, and H. Ailisto, "Identifying users of portable devices from gait pattern with accelerometers," in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 2. IEEE, 2005, pp. ii–973.

[97] D. Gafurov, K. Helkala, and T. Søndrol, "Biometric gait authentication using accelerometer sensor." *JCP*, vol. 1, no. 7, pp. 51–59, 2006.

[98] A. Jain, A. Ross, and S. Prabhakar, "Fingerprint matching using minutiae and texture features," in *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, vol. 3. IEEE, 2001, pp. 282–285.

[99] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE transactions on Image Processing*, vol. 9, no. 5, pp. 846–859, 2000.

[100] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[101] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.

[102] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: a deep quadruplet network for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 403–412.

[103] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in neural information processing systems*, 2018, pp. 8778–8788.

[104] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*. Springer, 2008, pp. 1–19.

[105] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, "Query-adaptive late fusion for image search and person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1741–1750.

[106] D. Chen, Z. Yuan, B. Chen, and N. Zheng, "Similarity learning with spatial constraints for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1268–1277.

[107] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, "Joint discriminative and generative learning for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 2138–2147.

[108] L. Wu, Y. Wang, J. Gao, and X. Li, "Where-and-when to look: Deep siamese attention networks for video-based person re-identification," *IEEE Transactions on Multimedia*, vol. 21, no. 6, pp. 1412–1424, 2018.

[109] Z. Zheng, L. Zheng, and Y. Yang, "A discriminatively learned cnn embedding for person reidentification," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–20, 2017.

[110] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *Proceedings of the iEEE conference on computer vision and pattern recognition*, 2016, pp. 1335–1344.

[111] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[112] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6036–6046.

[113] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe, "Deformable gans for pose-based human image generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3408–3416.

[114] X. Li, A. Wu, and W.-S. Zheng, "Adversarial open-world person re-identification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 280–296.

[115] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3754–3762.

[116] R. Benjamin, "Race after technology: Abolitionist tools for the new jim code," *Social Forces*, 2019.

[117] F. Bacchini and L. Lorusso, "Race, again: how face recognition technology reinforces racial discrimination," *Journal of Information, Communication and Ethics in Society*, 2019.

[118] M. Kunz, K. Kasper, H. Reininger, M. Möbius, and J. Ohms, "Continuous speaker verification in realtime," *BIOSIG 2011–Proceedings of the Biometrics Special Interest Group*, 2011.

[119] N. N. An, N. Q. Thanh, and Y. Liu, "Deep cnns with self-attention for speaker identification," *IEEE Access*, vol. 7, pp. 85 327–85 337, 2019.

[120] J. Galbally and R. Satta, "Three-dimensional and two-and-a-half-dimensional face recognition spoofing using three-dimensional printed models," *IET Biometrics*, vol. 5, no. 2, pp. 83–91, 2016.

[121] "Fingerprint Biometrics Hacked Again," http://www.ccc.de/en/updates/2014/ursel, 2014, [Online; accessed 27-December-2014].

[122] L. Yang, Y. Guo, X. Ding, J. Han, Y. Liu, C. Wang, and C. Hu, "Unlocking smart phone through handwaving biometrics," *IEEE Transactions on Mobile Computing*, vol. 14, no. 5, pp. 1044–1055, 2014.

[123] B. Shrestha, N. Saxena, and J. Harrison, "Wave-to-access: Protecting sensitive mobile device services via a hand waving gesture," in *International Conference on Cryptology and Network Security*. Springer, 2013, pp. 199–217.

[124] G. Kambourakis, D. Damopoulos, D. Papamartzivanos, and E. Pavlidakis, "Introducing touchstroke: keystroke-based authentication system for smartphones," *Security and Communication Networks*, vol. 9, no. 6, pp. 542–554, 2016.

[125] B. Draffin, J. Zhu, and J. Zhang, "Keysens: Passive user authentication through micro-behavior modeling of soft keyboard interaction," in *International Conference on Mobile Computing, Applications, and Services*. Springer, 2013, pp. 184–201.

[126] T. Feng, X. Zhao, B. Carbunar, and W. Shi, "Continuous mobile authentication using virtual key typing biometrics," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 2013, pp. 1547–1552.

[127] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE transactions on information forensics and security*, vol. 8, no. 1, pp. 136–148, 2012.

[128] M. Shahzad, A. X. Liu, and A. Samuel, "Secure unlocking of mobile touch screen devices by simple gestures: you can see it but you can not do it," in *Proceedings of the 19th annual international conference on Mobile computing & networking*, 2013, pp. 39–50.

[129] M. Ehatisham-ul Haq, M. A. Azam, J. Loo, K. Shuang, S. Islam, U. Naeem, and Y. Amin, "Authentication of smartphone users based on activity recognition and mobile sensing," *Sensors*, vol. 17, no. 9, p. 2043, 2017.

[130] C. Zhang, W. Liu, H. Ma, and H. Fu, "Siamese neural network based gait recognition for human identification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2832–2836.

[131] A. Das, N. Borisov, and M. Caesar, "Tracking mobile web users through motion sensors: Attacks and defenses." in *NDSS*, 2016.

[132] A. Das, N. Borisov, and E. Chou, "Every move you make: Exploring practical issues in smartphone motion sensor fingerprinting and countermeasures," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 1, pp. 88–108, 2018.

[133] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," *arXiv preprint arXiv:0903.3276*, 2009.

[134] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A practical attack to de-anonymize social network users," in *2010 IEEE Symposium on Security and Privacy*. IEEE, 2010, pp. 223–238.

[135] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: Using social network as a side-channel," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 628–637.

[136] S. Nilizadeh, A. Kapadia, and Y.-Y. Ahn, "Community-enhanced de-anonymization of online social networks," in *Proceedings of the 2014 acm sigsac conference on computer and communications security*. ACM, 2014, pp. 537–548.

[137] V. Griffith and M. Jakobsson, "Messin'with texas deriving mother's maiden names using public records," in *International Conference on Applied Cryptography and Network Security*. Springer, 2005, pp. 91–103.

[138] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," 1998.