UiT The Arctic University of Norway

Faculty of Science and Technology
Department of Computer Science

User Profiles Inferred from Smartphone Sensor/Context Data

How Features and Attributes Contribute to a Smart Nudge System

Håvard Bengtson

INF-3981 Master Thesis in Computer Science - June 2021

Abstract

Some of the most prominent challenges in today's society include lifestyle diseases and mental health problems. These issues can often be prevented or resolved by leading a healthy lifestyle[1]. Many people who could choose to foster healthy lifestyles find themselves stuck in unhealthy habits, finding it difficult to make positive changes.

Nudging is a term first introduced in economic and political theory. A nudge's goal is to change people's behavior without removing choice or being overly persuasive[2]. Smart nudges build on this principle, but are in a digital domain, and considers user context. Smart nudges are thought to be an effective tool to help users make positive changes to their lives.

Most people in the developed world own a smartphone and will keep it by their side at most times. Modern smartphones contain a range of different types of sensors, that can be exploited to extract features and infer user profiles. This has encouraged researchers to investigate what types of information can be found about users based on sensor/context data.

In this thesis user profiling in the smartphone domain is described and the findings of researchers are outlined. Researchers have found that sensors such as the accelerometer sensor, GPS, proximity sensor, and gyroscope, can be used to extract features and to infer user attributes. Extracted features such as activity, location, screen time, and app usage, can be used to infer user attributes such as mood, stress level, activity level, personality traits, age, and gender.

How user profiles inferred from smartphone senor/context data can contribute to a nudging system is discussed, and a smart nudge design is proposed. This is to my knowledge the only research that exploits user profiles inferred from smartphone sensor/context data to design a smart nudge system. Based on the proposed design and relevant research, I argue that smartphones are preferred to use in a smart nudge system. Both features extracted and user profiles inferred from smartphone sensor/context data are highly relevant. Features and user profiles can be used to decide when to nudge, to build the nudge, and to evaluate the nudge. The smartphone is always close to the user and can therefore be the medium that receives smart nudges.

Acknowledgements

I would like to thank my supervisor Prof. Randi Karlsen for providing helpful advice throughout this thesis. I am very grateful for her feedback and for being an ideal supervisor. I would like to thank my brother Audun Bengtson for proofreading my work and providing good feedback.

Lastly, I would like to thank my family and friends for whose help and support have led me to where I am today.

Contents

Abstract i

Acknowledgements iii

List of Figures ix

List of Tables xi

My list of definitions xiii

1 Introduction 1
- 1.1 Motivation . . . 2
- 1.2 Goal . . . 2
- 1.3 Contributions . . . 2
- 1.4 Methods and Methodology . . . 3
- 1.5 Outline . . . 3

2 Background 5
- 2.1 The sensor rich smartphone . . . 5
- 2.2 User Profiling . . . 7
  - 2.2.1 User Profile Types . . . 8
- 2.3 User Profiling Process . . . 9
  - 2.3.1 User Information Collection . . . 9
  - 2.3.2 Data Preprocessing . . . 10
  - 2.3.3 Feature extraction . . . 10
  - 2.3.4 User profile modeling . . . 11
- 2.4 Nudging . . . 14
  - 2.4.1 Digital Nudging . . . 15
  - 2.4.2 Smart Nudging . . . 15

3 Methodologies and Methods 19

4 User Profiling with Smartphone Sensor and Context Data 21
- 4.1 App information . . . 21

4.1.1 Installed app list 22
4.1.2 App usage records 23
4.1.3 Installation behaviors 24
4.1.4 Challenges 24
4.2 Activity and Location 25
4.3 Physical Health and Mental Health 26
4.3.1 Mental health 26
4.4 Health 28
4.5 Multidimensional approach 29
4.6 Summary 30

5 Design and Architecture 33
5.1 Architecture 34
5.1.1 Sensor/context monitor 35
5.1.2 Sensor and Feature Store 35
5.1.3 User profile 35
5.1.4 Nudge Store 36
5.1.5 Nudge Decider 37
5.1.6 Nudge Builder 38
5.1.7 Nudge Evaluator 39
5.2 Data Collection 39
5.3 Storage 40
5.4 User Profile 41
5.4.1 Demographics Attributes 41
5.4.2 Lifestyle 41
5.4.3 Interests 41
5.4.4 Personality traits 42
5.4.5 Psychological and health information 42
5.4.6 Inferring New Attributes 42
5.5 User Modeling 43
5.5.1 Preprocessing and Feature Extraction 43
5.5.2 User profile modeling 44
5.6 Summary 47

6 Implementation 49
6.1 System Architecture 49
6.2 Enabling Technologies 52
6.2.1 Flutter 53
6.2.2 Firebase 53
6.3 Data Collection 53
6.3.1 App Data 54
6.3.2 Pedometer Data 54
6.4 Data Storage 54
6.4.1 Data Model 54

6.4.2 Feature Store 55
6.4.3 User Profile 56
6.5 User Modeler 57
6.5.1 Adaptivity 57
6.5.2 Feature Extraction 57
6.5.3 User Modeling 58

7 Discussion **61**
7.1 Challenges: User Profiling from smartphone Data 61
7.1.1 Population Bias 61
7.1.2 Ground truth Data Collection 63
7.1.3 Power Consumption 63
7.2 How can User Profiling from Smartphone Sensor/Context Data Contribute to a Smart Nudge System? 64
7.2.1 Data Collection 64
7.2.2 Nudge Decision 66
7.2.3 Nudge Building 66
7.2.4 Nudge Evaluation 67
7.2.5 The Smartphone as a Medium to Receive Nudges . . 67
7.3 Evaluation of Implementation 67
7.4 Evaluation of Research Method 68
7.5 Future work 69

8 Conclusion **71**

List of Figures

2.1 The evolution of smartphones[14] 7
2.2 Types of user attributes[17] 8
2.3 Transformation from 1-dimensional data to 2-dimensional data[27] 13
2.4 Smart nudge architecture[36] 16

4.1 Example of app metadata[17] 22

5.1 Smart nudge architecture 34
5.2 Example of information stored about nudge 37
5.3 Saved data 46
5.4 Extract of User Profile Ontology 46

6.1 System architecture 51
6.2 Extended System architecture 52
6.3 Example feature store 56
6.4 Example user profile 57

7.1 Information gain from different demographic attributes . . . 62

List of Tables

4.1 Reviewed Research 31

My list of definitions

1

Introduction

Smartphones have become a necessary appliance to own in the 21st century. In 2020 there exists 3.5 billion smartphone users[3], and this number is predicted to grow by another 200 million by 2021. Smartphones have, in their technical evolution, Incorporated more and more sensors. Smartphones also contain several sources of context such as phone logs, installed app list, and app usage data. These sensors, context sources, technical advances in the smartphone's processing power, and the ability to always stay online give developers and researchers the possibility to infer user profiles. Researchers have used this opportunity to infer user attributes such as age, gender, lifestyle, mood, personality traits, health conditions, and mental well-being. By creating user profiles, this information can be stored and used by applications to provide personalized content.

The focus of this thesis is to investigate how user profiles inferred from smartphone sensor/context data can contribute to a nudging system. The term nudge arrives from political and economical theory. A nudge tries to alter an individual's behavior without removing choice and without being overly persuasive. Digital nudges try to nudge individuals in a digital domain.

In this thesis, I argue that smartphone sensor/context data greatly contributes to a nudging system. I argue this by reviewing research on user profiling from smartphone sensor/context data. I also propose a smart nudge system contributed by user profiles inferred from smartphone sensor/context data, and by implementing a system inferring user profiles from smartphone sensor/context data.

1.1 Motivation

Some of the most prominent challenges in today's society include lifestyle diseases and psychological health issues. Heart disease is the biggest killer in the world[4], and the cause can often be attributed to leading an unhealthy lifestyle. An unhealthy lifestyle is also connected to psychological health issues. Fostering healthy habits such as exercising, eating healthy, and getting enough social interaction can counteract these types of issues and improve quality of life[1].

Changing one's habits is however known to be a difficult task. Many people who could choose a healthy lifestyle find themselves stuck in unhealthy habits, being unable to foster healthy habits. A smart nudge system can help individuals to make these positive changes by providing personalized nudges.

1.2 Goal

The goals of this thesis are to:

- The main goal is to investigate how user profiles inferred from sensor/-context data can contribute to a smart nudge system.

- The second goal is to investigate what research has been done in the area of user profiling in connection to smartphone sensor/context data. Here, methods used, findings, and why it can be useful are investigated.

1.3 Contributions

User profiling from smartphone sensor/context data is a relatively novel research area. In this thesis, I provide an overview of this research. I describe domains, methods, and results.

This is, to the best of my knowledge, the only research that has been done concerning digital nudging in relation to user profiles inferred from smartphone sensor/context data. In this thesis, a smart nudging system is proposed. It exploits smartphone sensor/context data where user profiles and extracted features are used to decide when to send nudges, to build nudges, and to evaluate nudges.

Another contribution is an implemented system. The system infers user profiles from smartphone sensor/context data that can be used in a smart nudge system.

1.4 Methods and Methodology

Håkanssons "Portal of Research Methods and Methodologies for Research Projects and Degree Projects"[5] provides an overview of methods and methodologies and was used to find relevant methods and methodologies for this thesis. This was done by navigating through Håkansson's "portal" and identifying relevant methods.

This thesis uses *qualitative research*. "Qualitative research concerns understanding meanings, opinions and behaviors to reach tentative hypotheses and theories or develop computer systems, artifacts, and inventions."[5]. Qualitative research uses smaller data sets than quantitative research. Sometimes qualitative research is used before or after quantitative research to get a complete picture of a problem.

This thesis uses the research method *applied research*. Applied research aims to answer specific questions or to solve known and practical problems. This thesis aims to answer the question, "how can user user profiles inferred from smartphone sensor/context data contribute to a smart nudge system". To answer this question, relevant research is examined, finding how smartphone sensor/context data has been used to infer user profiles in other domains. A smart nudge design is proposed, and a prototype system is implemented.

The *inductive research* approach is used. Inductive research develops theories and propositions with alternative explanations from collected data[5]. This is done by describing and discussing relevant research concerning user profiling from smartphone sensor/context data and research concerning nudging.

1.5 Outline

The second chapter of this thesis includes a background on sensor-rich smartphones and the general process of user profiling. The third chapter provides an overview of what research methods were used to investigate the two goals of the thesis. The fourth chapter is an in-depth investigation of how smartphone sensor/context data have been used to infer user profiles and what benefits it holds.

In chapter five, I propose a smart nudge design contributed by user profiles inferred from smartphone sensor/context data. Chapter six describes the implementation of a system that infers user profiles from smartphone sensor/context data that is useful for a nudging system.

In chapter seven, I discuss the findings of my research. This includes a discussion of challenges connected to user profiling from smartphone sensor/context data, how user profiling from smartphone sensor/context data can contribute to a smart nudging system, and an evaluation of the implemented system. Lastly, in chapter eight, the thesis is summarized and concluded.

2

Background

2.1 The sensor rich smartphone

The world's first smartphone was introduced in 1992 by IBM[6]. This device could make phone calls and send texts but was also capable of: using the internet, had a notebook to take notes, a calendar, had a touch screen, and could use third-party apps. Although not much known, the first smartphone "Simon" managed to sell 50,000 units. Smartphones have since then had rapid technological advances, and in their evolution, more and more sensors have been incorporated. figure 2.1 illustrates this evolution.

A sensor can be defined as "A device which detects or measures a physical property and records, indicates, or otherwise responds to it."[7]. Notable sensors typically found in today's smartphones include:

- Proximity sensor: A proximity sensor is built of an infrared LED light and an infrared radiation detector. Its purpose is to detect nearby objects and to calculate the distance between them. It has a working range of around ten cm[8].

- Ambient light sensor: An ambient light sensor detects the amount of light in the surrounding area. It is used in today's phones to adjust screen lightness appropriately. For example, if the phone user is out in the sun, the screen will be brighter so that the user can see the screen better[8].

- Global positioning system (GPS): GPS is made by the US department of defense and contains 31 satellites orbiting the earth. It became widely available to the public in the 1980s and can now be used free of charge. To find a device's "geolocation", the device receives signals from four satellites. The satellites contain extremely accurate atomic clocks. The receiving device uses this to compute its geolocation[9]. A device's geolocation can also be found by the use of cellular networks and wifi.

- Accelerometer sensor: The accelerometer sensor notices and measures dynamic forces such a walking, running, or driving acting on an object[10].

- Gyroscope: A gyroscope provides stability or maintains a reference of direction. It achieves this by a spinning wheel or circulating beam of light. The gyroscope sensor can measure the tilt and orientation of a smartphone[11][12].

- Magnetometer: The magnetometer sensor measures magnetic fields. Its most typical use is as a compass[13].

- Touch sensor: The touch sensor notices touch to a certain area and is used by smartphones in the form of a touch screen.

- Camera: The camera, although not strictly thought of as a sensor, can be used as a sensor. The camera can, for example, be used to notice and react to movement.

- Microphone: The microphone is as the camera not thought of as a sensor, but can be used as one to notice and react to sounds.

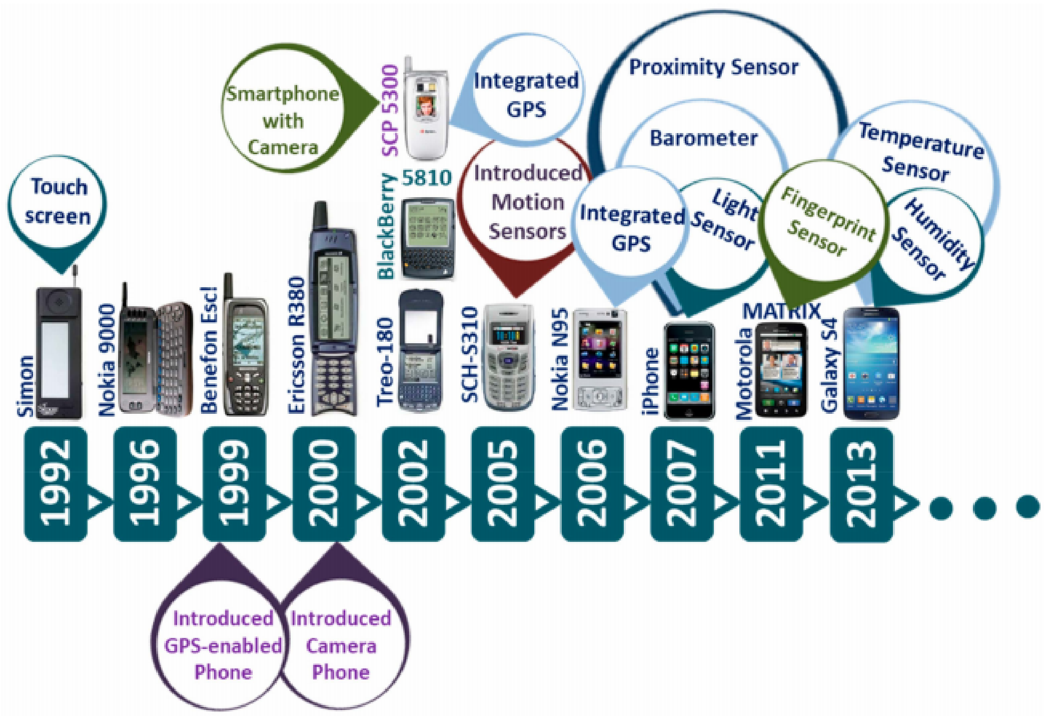

Figure 2.1: The evolution of smartphones[14]

2.2 User Profiling

User profiles are essential for any application that wants to provide personalized content for its users. A user profile can be defined as: "a digital representation of the unique data concerning a particular user."[15]. To create a user profile, you need to define what types of information are of interest and should be stored about your users. A user profile can either be static or dynamic and is created through the process of user profiling. User profiling is split into the phases of information collection, preprocessing, feature extraction, and user profile modeling[16].

Figure 2.2 shows different types of user attributes that are interesting when building user profiles.

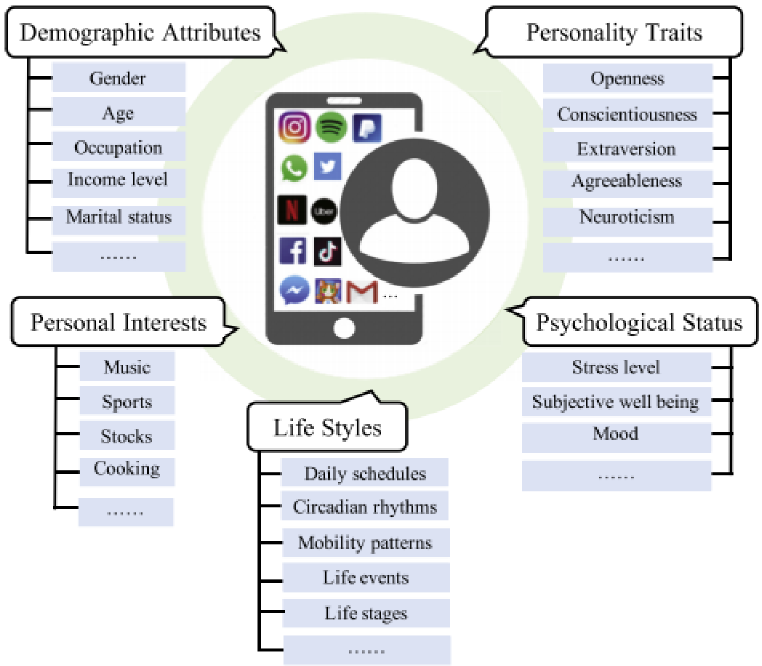

Figure 2.2: Types of user attributes[17]

2.2.1 User Profile Types

There are two main types of user profiles, the static, and the dynamic user profile.

Static User Profile

Static user profiles are mostly built out of static information. Demographic information such as age or gender are examples of these. The information is stored over long periods and is not changed after its creation. Data collection for static profiles is often explicit, e.g., through questionnaires[16]. Explicit and implicit data collection are discussed further in the section concerning data collection.

Dynamic User Profile

The dynamic user profile is ever-changing and is often used when the data is collected implicitly (e.g., inferred through the use of mobile sensors or web scraping). Dynamic user profiles contain information about the user that is likely to change. Some examples are mood, interest, and stress level. It is worth noting that some of these attributes change more often than others. Interests, for example, change a lot more slowly than mood. The dynamic user profile can also be referred to as a behavior or an adaptive profile[16].

2.3 User Profiling Process

The user profiling process consists of several steps. First of all, what user information we want to infer as well as the use case needs to be defined. The next step is to collect user information. Here, we need to define how the information should be gathered, either implicitly, explicitly, or a combination of both. After having gathered the information, it goes through a process of preprocessing, Feature extraction, and analysis. This step is referred to as the modeling phase to infer user attributes[16].

2.3.1 User Information Collection

User information can be gathered either explicitly or implicitly. The explicit method requires user interaction and is often used to find static information and build static user profiles. The explicit method relies on techniques such as questionnaires, where the user has to fill out a form. This technique typically seeks to find information such as the user's age and gender. Another explicit technique can be found in many recommender systems where the user is prompted to rate items.[16].

In the implicit method, the data is gathered by an intelligent agent or data mining techniques. This is achieved without asking for user feedback, but rather gathered and analyzed through user interaction with the system. One method is to scrape social media sites for user information. Many static attributes are stored here, such as age and gender, but there is also the possibility of gathering social media posts. These posts can be processed and analyzed by incorporating machine learning and statistical methods. Another example of implicit data collection is to collect sensor data from mobile devices, such as user movement, location, and app usage. This information can be modeled to infer user attributes.[16].

The implicit method is generally preferred in modern systems. This, to better avoid fake user profiles, where the user purposefully provides false information. Users also find the process of filling out forms tedious and are often reluctant to share their information with the system[16].

2.3.2 Data Preprocessing

The tasks of preprocessing and feature extraction are important to make the data both understandable for machines and humans and to make it ready for analyzes. The data often contains outliers, missing data, and duplicate values[18]. If a computational model is built based on this type of data, the inferred user profile will not correctly represent the user. There exist several libraries for programming languages such as Python to help preprocess data. Pandas[19] is an example of such a library.

- An outlier occurs when a value of a feature greatly differs from the other values. If, for instance, we have extracted the feature "income" from 1000 randomly selected people, and one of those people is Bill Gates, a computational model would not correctly represent the group of people as Bill Gates would skew the model. The value of Bill Gates should therefore be removed[18].

- Missing data can affect the computational model. This can, for example, happen as a result of an error in the data set. There are several techniques for solving this problem. Some problems might have occurred during the data collection phase. If this is the case, data can be collected anew to find the missing value. Other solutions are to remove the feature or to fill the value with the mean value of that type of feature[18].

- Duplicate values can be noticed with a library such as Pandas and should in some cases be removed[18].

2.3.3 Feature extraction

Before explaining the step of feature extraction, it is necessary to make a distinction between the word "feature" and the word "attribute". These words are often synonymous in the world of machine learning and statistics. In this thesis, however, I make a distinction to avoid any confusion. The word "feature" refers to information extracted from sensor data. This includes information such as step count, geolocation, and app usage. These features can be used in the modeling phase to find "user attributes". Activity level is a user attribute that can be found by analyzing the feature step count. If the user walks 20,000 steps daily, the activity level could, for example, be "very active".

The goal of the feature extraction phase is to reduce the set of raw data to smaller groups. This data must, however, still accurately and completely describe the original data set[20]. Accelerometer data can, for example, be used to calculate pedometer data(step counts). This data can be gathered into collections, such as Key-value pairs. Saving data like daily step count or this week's step count makes it easier to model and process the data.[16].

2.3.4 User profile modeling

A computational model is built using the extracted features. This step is called user profile modeling and is performed to infer user attributes. Some modeling techniques include machine learning, ontology-based, and filter approach[16].

Some of these techniques, like filter-based and ontology, require some user knowledge. If the user has not given this explicitly, it needs to be gathered implicitly. Then it might be necessary to infer user attributes using, for example, machine learning before performing the second technique.

Machine learning

"Machine learning focuses on applications that learn from experience and improve their decision-making or predictive accuracy over time."[21]. Machine learning algorithms come in two forms: either supervised learning or unsupervised learning. In supervised learning, the algorithm learns by using ground truth data. The ground truth data is labeled, meaning that the input data contains the correct output data[22]. This data is used to train a computational model. When the model receives unlabeled input data, it will have trained a logic to predict the correct output.

Supervised learning can be split into regression and classification problems. Regression algorithms aim to predict a numerical value. Classification algorithms aim to predict a discrete or categorical value[22].

Some common supervised learning techniques include:

- **Random forest** can be used on both regression and classification problems. It utilizes decision trees that answer a set of yes or no questions, creating a branch for "yes" and a branch for "no". After having answered the first question, each branch answers a new question and so on... These yes or no answers are made out of features. The features that cannot be presented as true/false are transformed.

  As an example, if we want to predict if an individual is more likely to own a house by looking at the features: income, in a relationship, and education, the features are transformed into yes/no questions. Income is made to above or below e.g the median salary, in a relationship is already a yes/no question, and education is set to having a university degree or not.

Then we find which of the yes/no questions most meaningfully shows whether a person owns a house or not. If 70 percent of people with an income higher than the median owns a house and the other features are split more evenly, the income should be the root node. The root node is split into two branches, and the process is repeated a given number of times.

Random forest is a collection of decision trees trying to predict the same attribute. To ensure that the decision trees are uncorrelated, bagging(bootstrap aggregating) and feature randomness can be used. Bagging is the act of creating smaller data sets out of the initial training data. Feature randomness makes sure that a random set of features are used on each decision tree. The prediction with the most votes is picked to be the correct prediction[23].

- **K-nearest neighbors** assumes a data set with known categories. When a new data point is added, it is categorized by looking at the k-nearest neighbors, k being a set value. The new data point is always placed in the category with the most votes[24].

- **Naive bayes** is a classification technique based on Bayes' Theorem[25]. It assumes that all features are independent and unrelated. Features are often related, and this is why the algorithm is referred to as "naive". Despite this, the algorithm works surprisingly well in practice[26].

  The navie bayes algorithm can be mathematically notated as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

  Which can be understood by using Bayesian probability terminology as:

$$posterior = \frac{priorXlikelihood}{evidence}$$

  A case where the Bayes algorithm can be used is when wanting to predict if a person will play outside based on the weather. First, the frequency of playing in different kinds of weather is found. Using this data, the likelihood of playing outside is found for each type of weather. Then the values are put into the naive Bayesian equation to predict if a person will play outside based on different types of weather.

- **Support Vector Machines**: Support vector machines aim to categorize a data point into one of two categories. This task can be difficult or impossible to achieve in a low-dimensional space. A hyperplane that clearly splits the categories is called a classifier. We want the hyperplane that creates the largest distance between the two categories.

  There are different "kernel functions" to transform the original data into higher dimensions to find relations between the transformed data points. These relationships are used to find a suiting classifier.

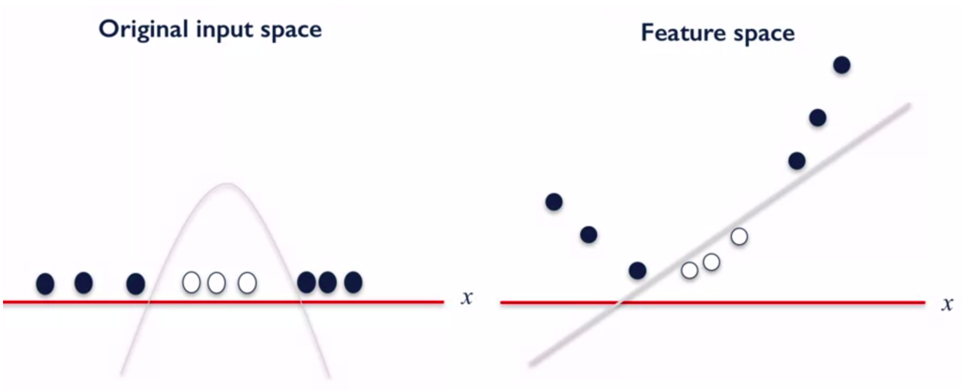

Figure 2.3: Transformation from 1-dimensional data to 2-dimensional data[27]

  As seen in 2.3 it was not possible to find a linear classifier to split the two categories of data marked in different colors. When moving the data into a higher dimension, however, it is possible[28].

In unsupervised machine learning, no data is labeled, and the model builds its logic without any external help. Whereas supervised learning is split into classification and regression problems, unsupervised learning consists of clustering and association problems[22].

K-means clustering is a clustering technique that aims to cluster a data set into k clusters. Initially, k random data points in the training set are selected as starting points for the clusters. The rest of the data points are placed into the cluster they are closest to. The mean of each cluster is calculated, and the data is re clustered based on the mean. This step occurs until it converges. All of these steps are repeated to find the combination that leads to the least amount of variation[29].

Neural Networks are loosely based on how the human brain works mimicking how biological neurons signal one another. Neural networks consist of an input layer, a range of hidden layers, and an output layer. Each layer consists of several nodes. Each node is connected to a number of nodes in the next layer and has an associated weight and threshold. Each node can be thought of as a linear regression model. If a node's result is above the threshold, it will pass the data to one or more of its connected nodes[30].

Ontology-based approach

Ontology-based approach: "Derived from its philosophical origin, an ontology in computer science is a data model that represents a set of concepts within a domain and the relationships between those concepts."[31]. The ontology is both human-understandable and machine-readable. The concepts can be represented in the form of classes. The classes are connected in a hierarchy by using "is-a" links forming parent/child connections. Any given class has properties that describe features and restrictions of the class[15].

Filter based approach

Filter-based approach: The filtering approach is found in recommender systems. It aims to recommend relevant items to its users and to filter out irrelevant items. This can be achieved e.g, by using content-based filtering, where items recommended are based on earlier liked items, or by using the collaborative-based filtering method. Here, user items are recommended based on what similar users have liked in the past[16].

2.4 Nudging

Nudging was first introduced in political and economical theory and is defined as "any aspect of the choice architecture that alters people's behavior predictably without forbidding any options or significantly changing their economic incentives."[2]. Nudging can often be found in everyday life. An example is signs along the road with pictures of people wearing seat belts. The sign does not remove the choice of not wearing seat belts. It does, however, increase the chances of people choosing to wear seat belts.

The practice of nudging has, not surprisingly, raised ethical questions. This because of the potential manipulating nature of the practice. Thaler and Sunstein[2] argue that nudging is in many ways inevitable and is therefore not ethically wrong. They argue that not nudging is often more ethically problematic than nudging. A grocery store, for instance, must be built in a certain way. There has to be an entrance and a shopping register. Somewhere in-between, there have to be items that the customer can purchase. Some of these items will, in turn, be closer to the register than others and are, as a result, more likely to be noticed by a potential customer. The fact that the item closer to the counter is more likely to be noticed and picked by the customer is an example of nudging. The architecture of the store makes this act of nudging inevitable. It has also been argued that nudging is ethical as long as it promotes welfare and autonomy[32][33]. There is proof that most people do not find nudges unethical as long as they are not overly manipulative or maleficent[34].

2.4.1 Digital Nudging

"Digital nudging is the use of user-interface design elements to guide people's behavior in digital choice environments."[35]. The way a user interface is designed has a large effect on how individuals end up using the app. This has caused the field of user interface design to become an important research area.

Colors can be used to nudge users. The color red, for example, incites feelings of urgency. An online store can take advantage of this to make users feel like they are in a rush to purchase items displayed on the screen. By incorporating other nudging techniques like saying an item is almost out of stock and that five other people are currently watching this item, the store can increase the chances of an impulsive purchase. This, would of course, not be considered an ethical nudge, as it is manipulative and its sole aim is financial gain.

For another, more ethically in-line example, imagine a meditation app. The goal of the app is to better users' life by creating a daily routine of meditation. It incorporates the color blue to make the user more calm and relaxed. It displays motivational messages when opening the app to make users motivated. It also sends daily push notifications to remind the user to meditate. All of these tools can be thought of as nudges.

2.4.2 Smart Nudging

Smart nudging further expands digital nudging by incorporating context. It was first defined in: "Recommendations with a Nudge"[36] as "digital nudging, where the guidance of user behavior is tailored to be relevant to the current situation of each user".

Karlsen and Andersen[36] proposed a general architecture for smart nudging illustrated in figure 2.4.

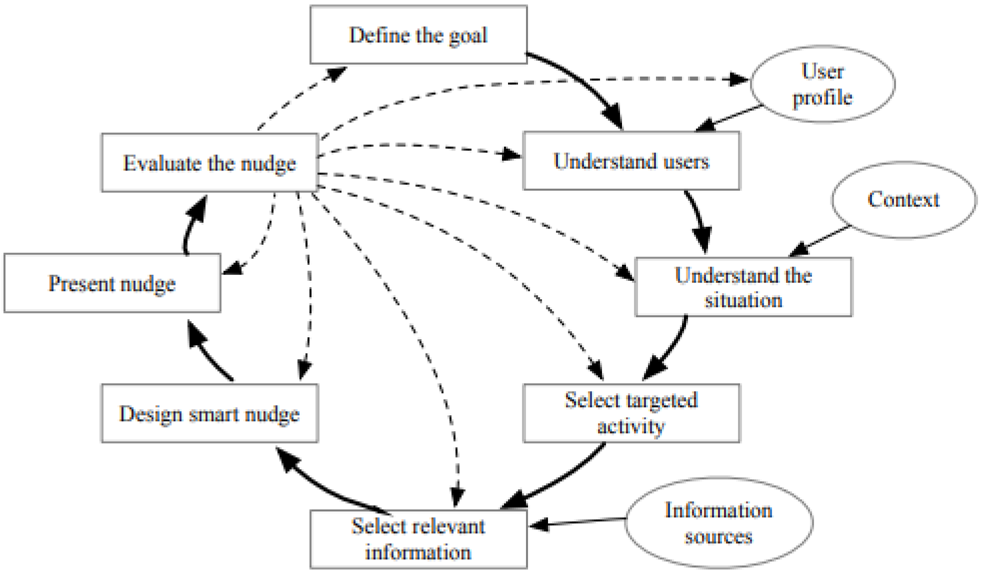

Figure 2.4: Smart nudge architecture[36]

The architecture can be summarized in 8 steps:

- Define the goal: In this step, a nudge goal is developed. A nudge goal is what the user should be nudged towards. It can, for example, be to encourage the user to be more active, eat more healthily, or be more mindful.

- Understand users: An understanding of users is achieved through the creation of user profiles. A user profile should contain relevant information in relation to the nudge goal. If the nudge goal is to make the user more active, for example, it is advantageous to store the user's activity level.

- Understand the situation: It is important to understand the current situation of the user to know whether or not it is appropriate to send a nudge. If the user is in church, for example, a nudge is probably not appropriate. If the user is sitting at home, watching TV, a nudge can be sent.

- Select target activity: What activity the user should be nudged towards is largely based on the nudge goal and current context. If the nudge goal is to make the user more physically active, and the user has to work in 45 min, a suggestion to bike to work might be a good nudge.

- Select nudge information: This is the information that will be used in the nudge. This information can be gathered from different information sources. If the nudge goal is to make the user more physically active, Context data, user attributes, and information from third parties, such as a weather service, can be collected. This information can be used to design the nudge, as described in the next step.

- Design smart nudge: Building on the example above, the nudge could be a message saying: "You have work in 45 minutes. The sun is out, and you have time to take the bike."

- Present to the user: A nudge can be presented to the user in two main ways. 1) Through push notifications and 2) through changes in user interface elements.

- Evaluate nudge: The last step is to evaluate whether the nudge was successful in influencing the user's behavior or not. This information can be used in the future to decide whether a nudge should be sent and what the nudge should contain.

3 Methodologies and Methods

The main aim of this thesis is to find how user profiling from smartphone sensor/context data can be used to contribute to a smart nudge system. Applied research was used to find an answer. The research approach is inductive, meaning that theories and propositions were proposed based on reasoning around collected data.

The case study strategy was used. "Case study is an empirical study that investigates a phenomenon in a real-life context where boundaries between phenomenon and context are not clearly evident."[5] A case study investigates the phenomenon using multiple sources of evidence. In this thesis, a literature review is performed. Here, research concerning user profiling from smartphone sensor/context data is described. The relevant research consists of both quantitative and qualitative research, all trying to find how user profiling from smartphone data can be used to infer user profiles, or how it can contribute to a specific domain, such as recommender systems, mental health, and physical health.

A smart nudge design based on Karlsen's design[36] is proposed that focuses on user profiling from smartphone data. A prototype system is implemented inferring user profiles from smartphone data.

Findings from the research review, the designed system, and the implemented system are used to strengthen the belief that user profiles inferred from smartphone sensor/context data can positively contribute to a smart nudge system.

4

User Profiling with Smartphone Sensor and Context Data

This section includes a description of selected relevant research done in the area of user profiling based on smartphone sensor/context data. Here different approaches and research areas are outlined.

4.1 App information

App information has been used to infer demographic attributes, personal interests, personality traits, and psychological information. These user attributes can be inferred by analyzing the installed app list, app usage records, installation behaviors, and app metadata[17].

Installed app list, app usage records, and installation behaviors are often combined with app metadata to provide some meaning to the data.

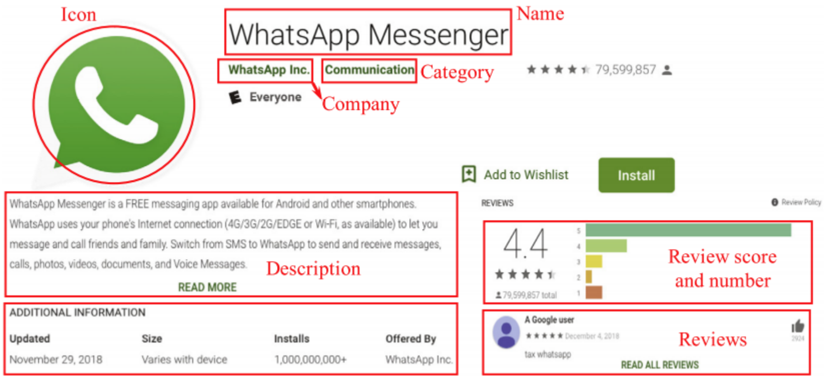

Figure 4.1: Example of app metadata[17]

Figure 4.1 Shows how app metadata can be scraped from the Google play store. Some examples of app metadata are app category, icon, reviews, and description.

4.1.1 Installed app list

The installed app list can easily be retrieved from android phones and contains a list of installed apps. Seneviratne et.al[37] were able to predict gender with a precision of 70 percent solely by looking at a snapshot of installed apps and their associated metadata scraped from the Google play store.

The authors collected a snapshot of the installed app list of 218 users(60 percent male and 40 percent female). They extracted a range of different features.

- Numeric Features: The authors extracted the number of installed apps, and cost of apps as numeric features. They found no significant difference between the number of installed apps in relation to gender. They did, however, find that men more often had paid apps installed in comparison to women.

- Category-based Features: Google play store categorizes apps into 30 categories. The authors extracted these features and found that the categories of "Libraries and demos" and "sports games" were more popular among men. They also found that the category "casual" was more popular among women.

- Item-based Features: The authors point out that some apps are largely more popular with one gender. Google Translate and Reddit is Fun are popular among men, but Pinterest and IbottaCash are more popular among women. Such information can also be found in market reports. One example is the app Pinterest which has 83 percent women users.

- Content-based Features: The authors point out the restriction of placing apps in one of 30 categories. To better describe apps, they extract content features. These features are made out of the app description scraped from the Google play store. They performed tf-idf[38] on the collected top terms after having prepossessed the data. They found that certain terms such as network and RSS were more popular among men, while the term decor was more popular among women.

The authors tested naive Bayes and SVM on different combinations of features. They found that SVM machine on the category features and naive Bayes on the content features gave the best results of around 70 percent. Adding more dimensions(features) did not improve the models.

They conclude that there exist more accurate methods for predicting gender, but that prediction using installed app list is a viable choice. This, because predictions can be made from solely one snapshot of the installed app list.

Despite the promising results by Seneviratne et.al[37], it is worth pointing out that over a quarter of installed apps are only used once and that most installed apps are seldom used[39]. Installed app list might therefore not be a good indicator of user interests.

4.1.2 App usage records

App usage records show how much time has been spent on each app. It can be collected either through 1) events, e.g, when an app is launched, or 2) by periodically pushing collected data[17]. MoodScope[40] for example, were able to predict mood based on app usage, phone logs, SMS, and email data. The authors used the circumplex mood model[41], a 2-dimensional model that measures activeness on one dimension and pleasure on the other. The app prompts the user to input their mood four times a day, with at least 3 hours of space between each input. The authors found, by using multi-linear regression, that phone logs and app usage were most effective at predicting mood. When training the model for two months, they were able to predict mood with 93 percent precision. The authors, however, point out the limitations of the app. Most participants of 32 individuals were young students, so the results might not represent other groups. They also point out that the app does not notice other important factors that might impact mood, such as stressful situations or the weather.

4.1.3 Installation behaviors

Installation behaviors include the actions of installing, updating, and deleting apps. Frey et al. [42] were able to predict life events(events such as marriage, buying a house, or getting children) with 65 percent precision by looking at installation behaviors. They achieved this by timestamping when apps were installed and sent out questionnaires to find life events over the last six months and the upcoming six months. They then split the set into training data and testing data evenly to train the model.

Lim et al. [43] surveyed 10,208 individuals from more than 15 countries and collected app data from 4,824 individuals. They found that installation behaviors varied among different countries. Americans are, for example, more likely to download medical apps, and Brazilians will more often download social networking apps. Chinese individuals tend to download the first app presented to them, and Mexicans are more likely to download paid apps. These findings suggest that installation behaviors can be used to predict what country an individual is from.

4.1.4 Challenges

Zhao et al. [17] point out challenges connected to user profiling through app data:

- Data collection: Most studies use a small dataset, and the duration of the collection is often quite short. This can cause problems during processing and modeling.

- Groundtruth collection: For classification and regression techniques, training data is needed. The authors point out that users are often not willing to provide this ground-truth data because of privacy concerns. Users might turn off the data collecting app or act differently because they know they are being monitored. Questionnaires suffer from the fact that they are prone to bias and that they are subjective.

- Population bias: Population bias is a problem when inferring user profiles. People of different age groups and different cultures are different. So if ground-truth data is collected from college students located in California, it will likely not correctly represent an elderly man from China.

- Pre-installed apps: Smartphones typically comes with a range of pre-installed apps. If we infer user profiles by looking at installed apps, it might cause some problems as some apps are not picked by the user.

4.2 Activity and Location

There are several papers on the subject of user profile inference using movement and location data collected from smartphones. The sensors used are accelerometer, gyroscope, and some sort of geolocation. User attributes found include lifestyle, points of interest(POI), and mental state.

State of art research using accelerometer and gyroscope have been successful in recognizing user activity. Khan et al. [44], for example, were able to detect and distinguish between human activities such as: sitting, walking upstairs, walking downstairs, and running. They achieved activity recognition of 96 percent by analyzing accelerometer data collected from smartphones.

Shoaib et al. [45] propose a system combining smartphone sensors and wrist-worn sensors to better recognize human activities. This approach is argued to be better at recognizing activities such as walking and running but is also able to recognize activities using hand gestures such as drinking, smoking, eating, and talking.

Geolocation has also been used to recognize activities. Natal et al[46] for example used geolocation to find POIs(points of interest). They combined this data with the user's profession and whether they have a car or not to recognize 13 different activities. Some activities include bank, breakfast, lunch, praying, in a car, or on a bus. They were able to accurately predict activities with 97.4 percent precision.

It seems as though lifestyle is a user attribute often found using movement and location data. Lifestyle can be thought of as daily routines and activities. Some examples of lifestyle features include office work, shopping, and being at the gym.

Geolocation can be used to identify lifestyles. Zion et al. [47], for example, collected daily trajectories of participants to identify and predict lifestyles. The data was collected using the Google location history. They gathered longitude, latitude, and timestamp data. The user then assigned semantic meaning like home, or work, to the POIs. Daily trajectories often appearing together were identified and thought of as lifestyles. This was achieved by using unsupervised neural networks. They then used the collected unique trajectories from all users to train a supervised neural network model to predict unseen lifestyles.

Wang et al. [48] developed a recommender system called Friendbook to recommend friends to its user based on similar lifestyles. They draw an analogy between daily lives and documents. As documents can be treated as mixtures of topics and topics as mixtures of words, "life documents" can be treated as a mixture of lifestyles, and each lifestyle as a mixture of activities. The authors claim that lifestyle is a better indicator of friendship than interest and shared friend-list, which is most often used today. This claim is supported by emerging sociology research[49]. Friendbook extracts activity features from the accelerometer and gyroscope sensors found in smartphones. Unsupervised learning is used on these features to identify lifestyles. They argue that unsupervised learning is better suited for this task as it is difficult to collect enough ground truth data for each activity, and the number of activities involved is unpredictable. They use the K-means clustering algorithm that split the activities into 15 clusters. The cluster centroids are distributed to users' smartphones which can then use it to recognize activities. The collected lifestyle documents are decomposed using the Latent Dirichlet Allocation (LDA) decomposition algorithm[50] to find users' lifestyle vectors. Similar users are then matched by comparing lifestyle vectors. Users are more likely to be matched if they share dominant lifestyles. By using geolocation, users close by with matching lifestyle vectors can be prioritized.

4.3 Physical Health and Mental Health

In this section, the two research areas of physical health and mental health are described.

4.3.1 Mental health

There has been some research on the area of sensor-rich smartphones in connection to mental health issues such as stress, depression, and anxiety. Ben-Zeev et al.[51] for example, extracted features of geospatiality, kinesthetic activity, sleep duration, and time of speech. All of these features were extracted using one or more of the sensors found in smartphones. These sensors were typically turned on periodically to collect data. Speech duration, for example, was found by activating the microphone every 2 min to check for speech using a speech detection system. Sleep duration was found using device "lock" duration, accelerometer, sound, and light levels.

The authors wanted to find whether there is a correlation between these features and the mental health issues: depression, stress, and loneliness. To investigate this, 47 individuals of age 19-30 years undertook a 10-week long study. Over this period, they used a smartphone provided by the authors that continuously collected sensor data. During the ten weeks, the individuals completed daily stress ratings. They also completed pre and post-tests for depression, stress, and loneliness. The authors found correlations between different types of sensor data and different types of mental health issues. This, by building mixed-effects linear models. The results showed that geospatial activity and sleep duration were correlated to stress levels. Speech duration and geospatial activity were associated with depression levels. Kinesthetic activity was associated with perceived levels of loneliness.

Another study by Gaggioli et al.[52] wanted to combine the power of smartphone sensors with the experience sampling method(ESM). ESM is a technique for collecting individuals' thoughts, feelings, and behaviors several times a day. This either periodically or at set times. Normally, individuals will have to carry with them a pen, self-report forms, and a beeper(to remind them to fill out the form). The addition of a smartphone removes all of these necessities.

One's psychological well-being can often be recognized physiologically. For example, if a person is in a stressful social situation, he/she might have an increased heart rate. This motivated the authors to collect sensor data from biosensors and to create an ESM smartphone app. Then they can combine individuals' subjective self-reported well-being with objective data collected from sensors, and thus give a better understanding between psychological and physiological mechanisms.

The authors tested their system on ten young individuals. They were provided with a smartphone and a biosensor, measuring heart rate, and equipped with an accelerometer. Data were collected at random intervals during waking hours over seven days. The authors found an association between both heart rate and heart rate variability in connection to self-reported stress. The accelerometer helped with identifying cases where the heart rate was high as a result of physical activity instead of a stressful situation.

The findings of Ben-Zev et al.[51], and Gaggioli et al.[52] show that sensor data from smartphones(and sometimes biosensors) can be of great use in the domain of mental health research. Ben-Zev et al.[51] showed that there is a connection between several features such as sleep duration and geospatiality, and mental health issues such as stress and depression. Gaggioli et al.[52] showed that there is a connection between physiological and psychological mechanisms by having individuals use an ESM app while being connected to biosensors. These findings are promising, and future research will likely include the prediction of mental health issues based on sensor data collected from smartphones.

4.4 Health

In their review: "smartphone sensors for health monitoring and diagnosis", Majumder and Deen[14] point out several possible use cases of smartphone sensors in the area of health. It can, for example, be used to monitor patients with disabilities and for noticing and diagnosing disabilities.

Monitoring heart rate and heart rate variability can be useful for monitoring and diagnosing cardiovascular diseases. The monitoring of heart rate can be achieved without the need for extra equipment by using a smartphone camera. By putting a finger on the camera, a photoplethysmogram(PPG) can be read. PPG is an optical technique to detect volumetric changes in blood in peripheral[53]. This signal can be used to get heart rate and heart rate variability. There have also been some attempts at reading the change of blood volume in the facial blood vessels by using the front-facing camera. This is another technique for getting heart rate and heart rate variability but is less reliable at this time.

The smartphone's microphone has been used in the field of pulmonary health monitoring. Larson et al.[54] were, for example, able to create a cough recognition algorithm with the precision of 92 percent. The user, however, has to have the phone in a chest pocket for the system to work. Another system was proposed by Chen et al.[55] to detect the actions of blowing the nose, sneezing, and having a runny nose.

Skillen et al.[56] propose a user profile ontology-based approach for assisting people with dementia in mobile environments. They argue that a context-aware system using an adapted user profile ontology can help lift the quality of life for people with dementia. They point out that current ontology techniques do not cover the aspect of personalization in enough detail. The GUMO model(General User Model Ontology), and the UPOS model(User Profile Ontology with Situation-Dependant Preferences Support) are lacking. GUMO focuses only on static user characteristics, while UPOS only focuses on the context surrounding a user.

The authors propose a design to better provide personalization to users. The user model consists of static information such as personal information, but also dynamic information such as social context, location, capability, context, activity, interest profile, preference profile, education profile, and health profile. Each class has properties. The HealthCondition class, for example, has the property hasHealthCondition. hasHealthCondition could be linked to the Capability class. So if an individual has the health condition "dementia" they will have the capability "lower memory".

The inclusion of context is important as it allows the system to adapt and the user ontology to be updated. It is also useful in situations where a person with dementia is lost as a health provider can quickly be contacted.

Although not having implemented a system, the authors illustrate a proposed application. Here, an individual with a mild form of dementia is used as an example. The system has learned that this individual goes grocery shopping every Monday and reminds her if forgetting to perform this task. The authors fail to mention how contextual data is gathered. Activity, context, location, and social context, for example, are difficult to identify.

4.5 Multidimensional approach

Yu et al.[57] use a range of smartphone sensor and context data, incorporating several features at once to predict user age, gender, and personality traits. This multidimensional approach led to promising results, as gender detection was 91.7 percent, and by using RSME[57], they found that their computational model was effective at predicting age and personality traits.

The proposed system architecture is a typical user profiling system architecture and includes the steps of data capture, feature extraction, and user profile inference. In the data capture phase, they collect a range of different sensor and context data found in smartphones. This includes the accelerometer, magnetic field, gyroscope, light sensor, app usage, battery, headset, mobile mode, network, and screen status event listener. They extract mobile phone usage features, preference features, and activity features. All user attributes were found using different types of supervised machine learning algorithms on one or more features. Using supervised machine learning makes sense as the detection of age, gender, and personality traits can be regarded as regression or classification problems. For ground truth data, they sent questionnaires to 523 individuals.

For age detection, the authors extract features using the accelerometer, gyroscope, and light sensors. They found that the way individuals open the phone screen can be used to predict age. The process of opening the screen is split into two actions. 1) picking up the phone, and 2) swiping the phone. For picking up the phone, the accelerometer sensor, and magnetometer is used to find the phone's elevation angle and slope angle. The researchers found that children like to pick up the phone with a back-and-fourth raise. Adults, on the other hand, have less fluctuation when picking up the phone, and the elderly are somewhere in-between children and adults. When swiping to open the phone, the elderly are the slowest and adults are the fastest. Elderly people also tend to swipe a longer distance. The authors extract the features: sliding start-end area, sliding angle, sliding distance, sliding duration, and sliding velocity. They also found a difference in how individuals of different age groups answered the phone. The Elderly took longer at answering the phone, and children moved more while having a phone call.

Users aged between 18 and 32 were sampled to predict gender. The authors found that the action of unlocking the screen could be used to predict gender. They found that women tend to shake their phone back and forth dramatically when picking it up. What apps are installed on the phone can also be used to predict gender. They analyzed the phones of 15 females and 15 males. Men were more likely to have installed many games, while women were more likely to have installed many photography apps.

Young adults were also used for training data when predicting personality traits. Feature extraction includes app usage, battery, screen status, mobile mode, network, and headset. The users needed to answer a questionnaire based on the five-factor personality model by Mcrae and John 1992[58]. The authors found that introverts spend more time on social applications to contact friends and that extroverts like traveling applications. The times the screen is lit up were less for extroverts than for introverts. Individuals high in openness used video and music applications more often.

4.6 Summary

Table 4.1: Reviewed Research

Reference	Sensors	Features	Methods	User Information	#Users	Duration	Result
Seneviratne et al[37]	-	Installed app list	Naive Bayes, SVM	Gender	218	-	70%
LiKamwa et al[40]	-	App usage records	Linear regression	Mood	32	2 months	93%
Frey et al[59]	-	App installation behaviors	key word-based classifier	Life events	2008	1 months	65%
Lim et al[43]	-	App installation behaviors	Correlation	Country differences	4824	2 months	-
Zhao et al[17]	-	-	Review user profiling smartphone apps	-	-	-	-
Khan et al[44]	Accelerometer, gyroscope	autoregressive coefficients, signal magnitude area, significant non-linear discrimination features	Artificial neural nets	Physical activity: sitting, walking, walking-upstairs, walking-downstairs, running	-	-	96%
Shoaib et al[45]	Accelerometer, gyroscope, linear acceleration sensor	time-domain features: mean and standard deviation	KNN, decision trees, Navie Bayes	Thirteen physical activities	-	-	97.4%
Natal et al[46]	GPS	Geolocation, POIs	Artificial Neural Network, decision tree,SVM, radio frequency machine learning	Thirteen activites	10	10 days	97.4%
Wang et al[48]	Accelerometer, gyroscope, GPS	Activites	K-means clustering, LDA, Filtering	Lifestyle	8	3 months	-
Zion et al	GPS	Geolocation, POIs	Unsuperviced/Superviced NN	Lifestyle	38	2 weeks	-
Ben-Zeev et al[51]	Microphone, accelerometer, GPS, WiFi receivers, Device "lock" time, light sensor	Geolocation, kinesthetic activity, sleep duration, sound, time of speech, light levels	Mixed-effects linear modeling, penalized functional regression	Depression, Stress levels, lonliness	47	10 weeks	-
Gaggioli et al[52]	Accelerometer, biosensor	Heart rate, heart rate variability	Pearson correlation	Stress levels			-
Majumder and Deen[14]	-	-	Review smartphone Sensors for Health Monitoring	-	-		-
Skillen et al[15]	-	-	Ontology	-	-	-	-
Yu et al[57]	Accelerometer, magnetic field, gyroscope, light sensor, battery	Mobile phone usage features, usage features, preference features, activity features	Random forest regression/classification, SVM	Age, gender, personality traits	84	2 weeks	Gender: 91.7%, Age: RSME 4.3696, personality traits: RSME openess 0.29, conscientiousness 0.3506, extraversion 0.465, agreeableness 0.3022, neuroticism 0.452
Peltonen et al[39]	-	App usage records	Kullback-Leibler divergence	Information gain demographic attributes	3293	-	-

Table 4.1 shows an overview of the relevant research that was described in this chapter.

This chapter described how powerful sensor-rich smartphones can be at inferring user profiles. The literature shows how static user attributes like age and gender can be inferred implicitly, whereas they usually are inferred explicitly. It also shows how dynamic user attributes such as mood and stress level can be predicted with high accuracy.

The literature shows that the choice of features and modeling techniques can have a large effect on the accuracy of prediction. Seneviratne et al.[37], for example, found that modeling category and content features performed better than modeling numeric and item features. They also found that different modeling techniques such as naive Bayes and SVM led to different results.

The number of dimensions(features) can have an effect on the result. Whereas Seneviratne et al.[37] found that expanding dimensions did not increase the accuracy of the model, Yu et al.[57] found that expanding dimensions increased accuracy.

Research shows how user profiles inferred from smartphone sensor/context data can contribute to different research domains such as psychology and medicine. Ben zev et al.[51], for example, found correlations between sensor data and mental health issues such as stress, loneliness, and depression. These findings suggest that sensor data can be used to predict periods of mental health issues without the need for a psychologist. Another example is Skillen et al.[56], who proposed a user profile ontology that can help raise the quality of life for individuals suffering from health issues such as dementia.

In the next chapter, I will investigate whether such advantages can be found by combining user profiles inferred from smartphone data with a smart nudge system.

5

Design and Architecture

In this chapter, I propose a smart nudge design. It is based on the smart nudge design formulated by Karlsen and Andersen[36] but is narrowed down to the domain of smartphones. Here, the focus lies on how user profiles inferred from smartphone sensor/context data can contribute to a smart nudge system.

The proposed design is innovative in the field of digital nudging. User profiling from smartphone sensor/context data provides advantages to a smart nudge system not found elsewhere. As described in chapter 4, smartphone sensor/-context data can be used to infer both static and dynamic attributes with high accuracy. I argue that smartphone sensor/context data contributes to a smart nudge system, not only by providing relevant attributes such as personality traits, mood, and age but also by providing relevant features such as movement data and geolocation. These attributes and features are used to 1) decide if a nudge should be sent, 2) build the nudge, and 3) evaluate if the nudge was successful.

Smartphones are readily available for nudges as they are always close to their users. This also provides the advantage of continuous data collection to build accurate user profiles. The continuous monitoring of users provides fresh context and highly adaptive user profiles. All of this without asking for user feedback.

In the following sections, I introduce the smart nudge design. First, the architecture is outlined, and its components are explained. Then an in-depth description of the user profiling process is given.

5.1 Architecture

Figure 5.1 illustrates the proposed smart nudge design. The sensor/context monitor continuously collects user data and notifies the nudge decider in certain events. It is also used by the nudge evaluator to decide whether a nudge was successful or not. The collected data is stored in the sensor store and the feature store. The user modeler component preprocesses data, extracts features, and performs modeling techniques on the extracted features. The inferred user attributes are saved in the user profile. The nudge decider and nudge builder have access to the feature store and user profile and use this information to decide when to nudge and to build nudges. Nudges and metadata concerning nudges are stored in the nudge store.

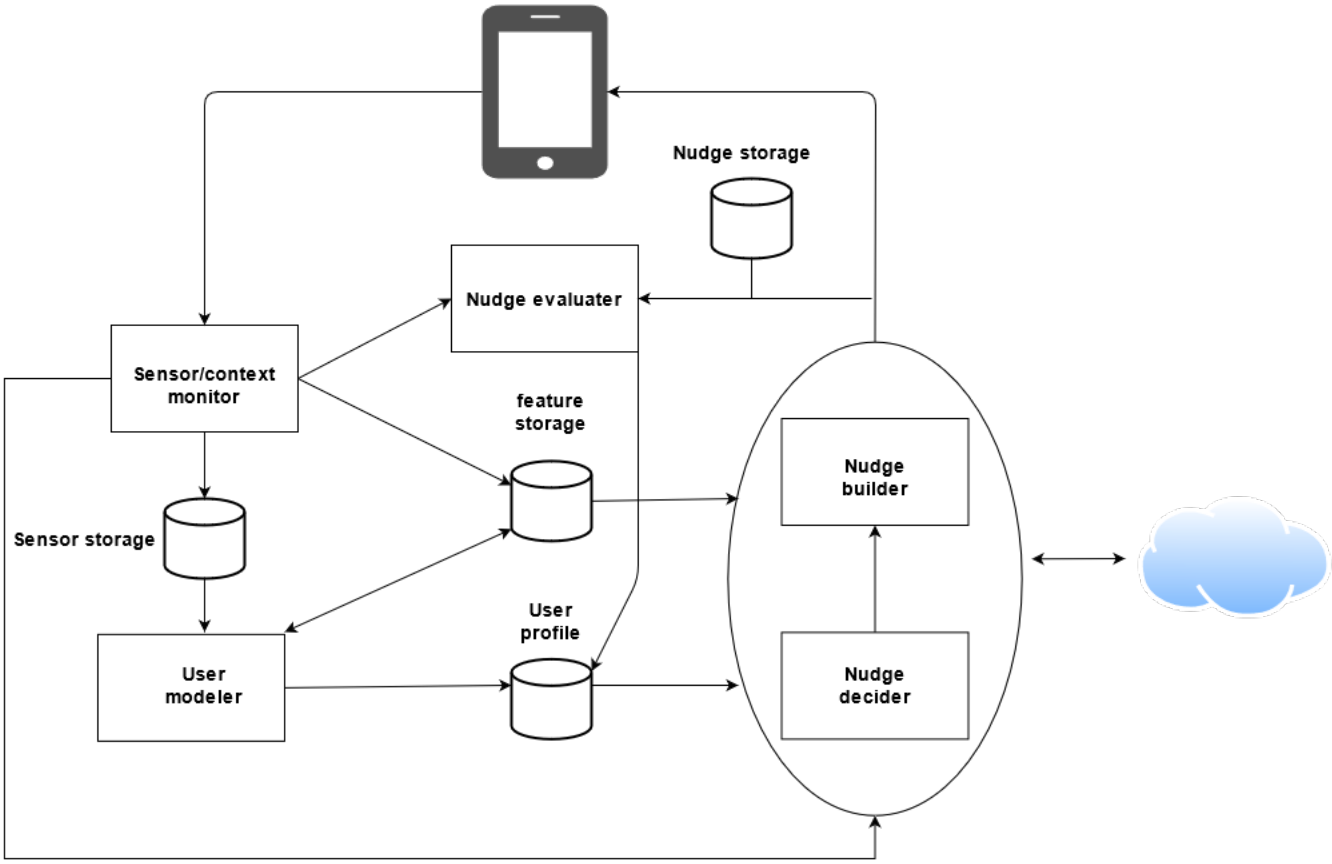

Figure 5.1: Smart nudge architecture

5.1.1 Sensor/context monitor

The sensor/context monitor component has several tasks. It can mainly be thought of as the data collection component where different sensor data are collected. Certain events can be defined that trigger a notification to the nudge decider. Then the nudge decider has to decide whether a nudge should be sent or not. An example of such an event is if the sensor monitor collects movement data and notices that the user has been idle for a long period. It can then notify the nudge decider, which might decide that a nudge is appropriate. It can also notify the nudge evaluator to notice if a nudge was successful in influencing the user.

5.1.2 Sensor and Feature Store

The sensor and feature store both reside on the smartphone and a server. Raw sensor data is stored in the sensor store, and extracted features are stored in the feature store. Feature extraction can be done on the client phone or the server-side. If a type of sensor data is used to find several features, it might be advantageous to perform feature extraction on the server-side. This, because performing many computational tasks on the client device might lead to more power consumption. If a type of sensor data is used to extract one type of feature, however, feature extraction should be performed on the client smartphone. This is because the act of feature extraction can be thought of as a compression technique, and the sending of compressed data drains less energy than sending raw data.

In some designs, there might be no need for a sensor store. This is the case when features are extracted on the client device and no sensor data is stored.

5.1.3 User profile

The user profile contains user attributes inferred from features by the user modeler. The user profile can be a mixture of static and dynamic attributes. These attributes are used to decide when to send nudges and how to build nudges.

As an example: A user profile contains interests, mood, and income level. The nudge goal is to help the user discover new experiences. The system has found that the user is interested in classic art, is currently in a good mood, and has a high income. A personalized nudge proposes a modern art exhibit. The system knows that the user can afford the experience and that the timing might be good, as the user is in a good mood.

5.1.4 Nudge Store

The nudge store contains previously issued nudges. Figure 5.2 shows important information that should be stored about a nudge in JSON format. This information can be analyzed to learn what motivates a user towards a nudge goal and in what contexts the nudges tend to be more successful.

The nudge store contains the information:

- Type: There are two types of nudges that can be received on a smartphone. Namely, push notifications and UI updates.
- Representation: This is the content of the nudge. The nudge always contains some sort of explanatory text but might also contain pictures.
- Action: This is the action the user was nudged to do.
- Reaction: Was the nudge successful, i.e., did the user perform the intended action.
- Information used: This is what information was used to both decide to nudge and to build the nudge. This information is gathered from the user profile, the feature store, and third-party sources. The information is typically a subset of the user profile and the feature store. By varying what information is used, the nudge evaluator can learn what information is of importance to the user.

```
{
  "Nudge": {
    "Type": "Push-notifcation",
    "Representation": "The sun is out, and we found you a close-by hiking trail!",
    "Nudge Goal": "Become more physically active",
    "Action": "Hike",
    "Reaction": "success",
    "Information Used": {
      "Attributes": {
        "Age": 22,
        "Mood": "Happy",
        "Health Information": "No known conditions"
      },
      "Context": {
        "Activity-level": "Idle",
        "Location": "Home",
        "Weather": "Sunny"
      },
      "Other": "Hiking trail"
    }
  }
}
```

Figure 5.2: Example of information stored about nudge

In the example nudge shown in figure 5.2, the user is nudged to be more physically active. The nudge suggests a close by hiking trail as the weather is good. The user profile contains age, mood, and health information. This information suggests that the user is capable of completing a hike. The user's context tells us that the user is idle, at home and that the weather is sunny. Included is a hiking trail that shows up when the user opens the notification.

5.1.5 Nudge Decider

The nudge decider pulls information from the feature store, the user profile, and third-party services. This information is used to decide if a nudge is appropriate in the current context. Both the feature store and the user profile are used to find user context. The feature store might contain geolocation and activity-level. The user profile might contain mood and stress levels.

The user profile also contains personal information describing what types of situations the user is open for nudges in. This is also based on previously successful nudges.

The nudge decider will in the early stages of the system's lifetime find appropriate times to nudge based on a predefined logic. This predefined logic is defined by the developer and based on common sense. We, for example, know that suggesting a mountain hike in the middle of a winter storm will not appeal to most people. It is, therefore, safe to assume that this would be a bad nudge. Sending it while it is sunny, on the other hand, is more likely to resort in a good nudge.

As the system adapts, it learns which nudges have been successful in the past and becomes able to predict which nudges will work in the future. As an example, let us say that a nudging system has the nudge goal of making a user more social. At the start, the system sends social nudges based on local events. The system learns that the user gets more stressed in social settings with many people. It learns this by monitoring heart rate via a smartwatch and collects metadata about the event(how many are attending the event) from a third-party service. It has, however, learned that the user responds well in smaller, more controlled social settings if they are already in a good mood.

In the example, a third-party service informs the nudge decider of an upcoming event. A decision to nudge is made by comparing earlier nudges analyzed by the nudge evaluator. The nudge decider also analyses information from the feature store, the user profile, and a third-party service to make a decision.

5.1.6 Nudge Builder

The nudge builder builds and presents nudges when notified by the nudge decider. There are two main ways of presenting nudges to smartphones. 1) to issue push notifications, or 2) to update UI elements.

Push notifications have the best effect when the content is personalized[60]. This system already saves sensor/context data, features, and user attributes that can be used to build personalized nudges. If the nudge goal is to make the user get more sleep, a push notification can be issued containing sleep statistics showing the user's progress. If the goal is to make the user more active, the nudge could be motivation to go for a walk. Information of close-by hiking trails or the weather forecast can, for example, be used to build and personalize the nudge.

UI elements can be updated similarly but assume that the user is using the app or will be soon. A nudge presented as a UI element can, for example, be a news feed relevant to the nudge goal. If the nudge goal is to make the user lead a more healthy lifestyle, the news feed can be linked to relevant health articles. Another example is motivational quotes towards the nudge goal. If the nudge goal is to make the user more productive, the motivational quote can be from a famous successful person pointing out the importance of working hard.

5.1.7 Nudge Evaluator

The nudge evaluator evaluates prior nudges to predict what nudges will work in the future. This is achieved by analyzing the information stored in the nudge store about earlier nudges. To check whether the nudge was successful or not, the user can be prompted to rate the nudge after having received it. Another solution is to exploit the sensor monitor or feature store to find if the user was successfully nudged.

For example, if the nudge aimed to make the user eat a healthy lunch and suggested a close by vegetarian restaurant. The nudge evaluator can pull the feature geolocation from the feature store to see check if the user went into the restaurant or not. Another example is in the case of a movement nudge. The nudge evaluator can check the feature step count to notice whether the user went for a suggested walk or not.

The findings of the nudge evaluator can be saved in the user profile.

5.2 Data Collection

User data is typically collected in two ways: explicitly, where the user provides information themselves, or implicitly, where information is gathered by an intelligent agent without user interference[16]. In this design, the implicit method is picked to take advantage of the readily available sensors found in today's smartphones. As described in chapter 4, sensor/context data can be used to successfully predict user attributes. The type of phone and type of data can affect the collection process. Android phones are typically more lenient than iPhones when it comes to the collection of user data, as Apple is more privacy-minded. Some tasks like collecting app usage data are therefore only possible on android unless using a jail-broken iPhone[40].

Several types of data can be obtained from both Android and IOS devices. Programming languages like Java[61], Swift[62], and the development kit Flutter[63] contain many third-party libraries that help collect, pre-process, and extract features from sensor data. Another possibility is to use state of art solutions concerning, for example, activity recognition that are often open-source.

Power consumption might become a problem from continuously collecting sensor data, but also from network calls, and performing tasks on the sensor data such as preprocessing, feature extraction, and modeling techniques.

A design decision needs to be made regarding how the sensor data is collected. If data are continuously collected from several sensors, the battery might drain out rather quickly. A solution is to periodically collect the data rather than collecting it continuously. Ben-zeev et al.[51] used such a solution in their app for collecting and inferring mental health data. When measuring speech duration, the microphone was turned on periodically every 2 minutes to check for speech using a speech detection system. It remained active if speech was detected, and was turned off if it was not.

Another solution might be for the system to learn when certain types of sensor data are unnecessary to collect. It is, for example, unnecessary to collect GPS data when the user is at work. GPS tracking can therefore be turned off during working hours.

Extracting features on the smartphone itself can lead to less power consumption. Miluzzo et al.[64] compressed sensor data before sending it to a server to save power on network calls. The extraction of features is a type of data compression and can thus lead to less power consumption.

5.3 Storage

It makes sense to create two storage layers. One local on the client's smartphone and one on some sort of server. This way, sensor data can be temporarily stored on the smartphone and be periodically pushed to the server. This will save storage and energy of the smartphone, as the nudge builder, nudge decider, modeler, and evaluator can issue network calls to the server instead of the smartphone.

The sensor and feature store are both present on the smartphone and a server. Other stores such as the nudge store and user profile are only present on a server. In this design, data is split into fresh and historical data. What is defined as fresh data is a design decision and can, for example, be data collected over the past two weeks. The data collected in the past two weeks will be stored in the fresh data store until being sent to the historical data store after two weeks.

All types of data stores will be split into fresh and historical data. This is useful as we want the system to be adaptive. It should adapt and change based on how the user changes. Mostly the fresh data is used. This, to reflect how the user is currently. Historical data can be used to see how the user changes over time and for tasks where a larger data set is needed.

5.4 User Profile

The user profile is inferred by creating a computational model on the extracted features. Information that can be stored in a user profile includes Demographic attributes, interests, lifestyles, personality traits, health information, and psychological information.

The nudge goal will have an impact on what information should be stored in the user profile. It is, for example, natural to save a user's activity level if the goal is to make the user more active.

5.4.1 Demographics Attributes

Demographic attributes such as gender, age, occupation, and income level are useful in a nudging system. Gender and age, for example, indicate what interests the user might have. A 15-year-old boy is more likely to be interested in football than a 75-year-old woman. Another example is income level. A person with a high-income level can be nudged towards more paid activities than a person with a low-income level.

5.4.2 Lifestyle

Lifestyle describes how individuals live day-to-day life and can be thought of as a collection of daily activities and habits. This information is highly relevant in a nudging system. Learning an individual's daily routines and activities can especially help to decide when a nudge should be sent. If the user's schedule is known, we know when the user goes to work, is at the gym, or home. The user should probably not be nudged when at work or the gym but can be nudged when at home.

5.4.3 Interests

Knowing an individual's interests can help to find what kind of activity the user should be nudged towards. If the nudge goal is to make the user eat a more healthy diet, and he/she is interested in Chinese cuisine, a healthy Chinese recipe can be proposed.

5.4.4 Personality traits

Personality traits are connected to how a person acts in different situations. The personality traits extroversion and introversion tell if the person likes to spend time in social settings or alone. Conscientious people are more structured and less impulsive than people high in openness, who tend to be more impulsive. People high in neuroticism are more easily stressed[65], something that needs to be taken into consideration when creating nudges.

Information about an individual's personality traits guides us in knowing what type of activities are more likely to succeed. An extrovert is, for example, more likely to be motivated by a social nudge. A person who is high in openness is typically open to trying new things[65]. This type of person should therefore get a wider range of nudges as we know that the individual is open to them, and he/she might get bored by only receiving a narrow range of nudges.

Neuroticism is an important personality trait to know of in a nudging system. Individuals high in the neuroticism personality trait experience more stress and generally feels more negative emotions[65]. To such an individual, nudges should be sent in caution as not to cause stress or anxiety. These types of individuals are also prone to sudden changes in mood[65]. By noticing such changes, nudges can be sent when the individual is in a good mood rather than in a bad one.

5.4.5 Psychological and health information

Psychological and health information: Information about an individual's psychology and health provides knowledge about the individual's psychological and physical capabilities. A person who suffers from a high degree of social anxiety should not be nudged to go into a socially stressful situation. A person who has lowered mobility capabilities should not be nudged to go on a mountain hike.

5.4.6 Inferring New Attributes

It is worth noting that information about one/several types of user attributes can be used to infer other types of attributes. Demographic information is, for example, tied to interests. We know that a young woman from California is likely to hold other interests than an old man from China. We also know that a young woman is likely to be physically capable of more activities than the old man. The old man might, on the other hand, have a higher income and is therefore capable of participating in more paid activities.

The personality trait extroversion indicates that a person is more likely to be interested in traveling. If a user is known to be extroverted, it is natural to test this by issuing a travel nudge. If a person is high in neuroticism, however, a travel nudge might cause stress.

It seems as though it gets easier to infer new user attributes the more user information is already known. Implementing a logic to infer new user information from already gained knowledge can, however, be challenging. It presupposes a knowledge of how people are and behave in society. Inference of new information should therefore be argued based on research, and maybe in collaboration with experts in other fields such as psychology and sociology.

5.5 User Modeling

User modeling refers to tasks such as preprocessing, feature extraction, and user profile modeling. Picking a modeling technique is a design choice based on extracted features and what user attributes are to be inferred. This design choice should be made based on testing and findings from research having solved similar problems with similar data.

5.5.1 Preprocessing and Feature Extraction

Preprocessing and feature extraction are two important tasks to perform before creating a computational model. The occurrence of false data, outlier data, missing values, and duplicate values can impact the model in such a way that it does not correctly represent the users.

Once a set of features have been extracted, the features can be used again to extract new features. This action is called dimension reduction and is performed to avoid "the curse of dimensionality". The curse of dimensionality happens when there are so many dimensions(features) that the model struggles to find meaning in it. This is because as the dimensions grow, the need for data to find meaning grows exponentially[66]. There are also times when there are too few dimensions to find meaning in the collected data. In such cases, expanding the number of dimensions can solve the issue.

Preprocessing can happen in several steps, before and after having extracted features. The act of feature extraction is often thought of as a part of preprocessing, where it would be referred to as the data cleansing stage. Preprocessing and feature extraction can be the most time-consuming task when building user profiles. Python is often used for such data science tasks, and libraries such as NumPy[67], pandas[19], and sklearn[68] can be used to identify and handle outliers, missing values, and duplicate data.

5.5.2 User profile modeling

There are different types of modeling techniques that can be used to infer user profiles. As seen in the literature, most implicit collection techniques require some sort of machine learning or statistical analysis to infer user attributes.

After having inferred the attributes, other user modeling techniques such as filtering can be used to, for example, group people together based on interest. Ontology-based techniques can also be used as a second modeling technique to define the structure of the user profile and to make the profile more human-understandable, and to infer new user attributes.

Existing solutions

In this thesis, several exiting solutions for inferring user profiles from smartphone sensor/context data have been addressed. If pre-existing open source solutions are available and adequately solve the problem, it might be advantageous to use them.

As seen in research, different modeling techniques can have a big impact on the quality of user attribute prediction. Choice of features and number of dimensions can also have an impact on prediction outcomes. Yu et al.[57], for example, found that more dimensions lead to better predictions, and Seneviratne et al.[37] found that more dimensions did not lead to better predictions.

From chapter 4, we know that:

- Mood can be predicated with more than 90 percent precision by using multi-linear regression on app usage data. This technique did, however, require user interference while teaching the model[40].

- Life events can be predicted with 65 percent accuracy by looking at installation behaviors[42].

- Activity can be recognized with more than 95 percent precision by building an artificial neural network model on activity features[44], and that such information can be used to predict lifestyle[48].

- Touch screen features and installed app list can be used to predict gender[57].
- Mental well-being can be inferred by analyzing sensor data like heart rate, accelerometer data, and geolocation[51][52].
- Different types of health information can be inferred from smartphone sensor data[14].
- Personality traits can be inferred from several types of features including app usage, battery, and screen status[57].

Picking a Technique

The most reliable way to pick a modeling technique is through testing. If machine learning is used, a technique can be picked by testing several machine learning algorithms on different features by using cross-validation[69]. Cross-validation helps to predict how well the machine learning algorithms will perform with different inputs.

Trade-offs have to be made. Is it more important that the algorithm is precise? or should it be fast? Is computational power an issue? And what model can be trained using the available data?

Unsupervised learning is argued to be a good solution if privacy is in focus. Privacy, although not the subject of this thesis, is of utmost importance in any system that stores user information. Highly sensitive information such as health data, demographic data, and daily routines are of interest in a nudging system.

Wang et al. [48], the creators of Friendbook, use unsupervised machine learning to infer lifestyle vectors. They use K-means clustering to discover 15 activities. The developers do not know the meaning behind these 15 activity types, and thus provide an additional privacy layer. Unsupervised learning is, therefore, a viable choice with the additional advantage of improving user privacy.

User Profile Ontology structure is used for knowledge representation and semantic visualization. By looking at relevant research, ontology-based user profiling sticks out as a potential technique that can contribute to a nudging system.

Skillen et al. [15] constructed a user profile ontology for the smartphone domain with an emphasis on context-awareness.

Class Name	*Class Description*	*Example Values*
User	Type of user involved.	“Physically-disabled”, “Normal”
User_Profile	Every user has one user profile.	
Activity	User activities, social or work related.	“Stamp collecting”, “Writing”
Activity_Type	The type of activity the user is involved in.	“Indoor” “Cooking”, “Outdoor”
Capability_Level	The level the user can cope with things.	“Severe”, “Low”, “Moderate”
Capability_Type	The type of capability that the user has.	“Cognitive”, “Emotional”
Interest	A user hobby or work-related interest.	“Swimming”, “Reading”
Interest_Type	The type of interest that the user has.	“Computing”, “Food”, “Sports”
Interest_Level	The level of interest the user has.	“Low”, “Medium”, “High”
Preference_Domain	The area of preference/likes.	“Exercise”, “Food”, “Travel”
Health_Condition	The health conditions associated.	“Dementia”, “Diabetes”
Health_Level	The current health level/status of the user.	“Mild”, “Normal”, “Severe”
Location	Where the user in situated at one time.	“House”, “Park”, “Cinema”
Time	The time of day associated.	“Morning”, “Evening”
Context	The environment that the user is in.	“Working”, “Social”, “Home”

Figure 5.3: Saved data

Figure 5.3 describes the types of data saved in the user Ontology. It stores, among other things: activity, interest, health data, context, and capability levels.

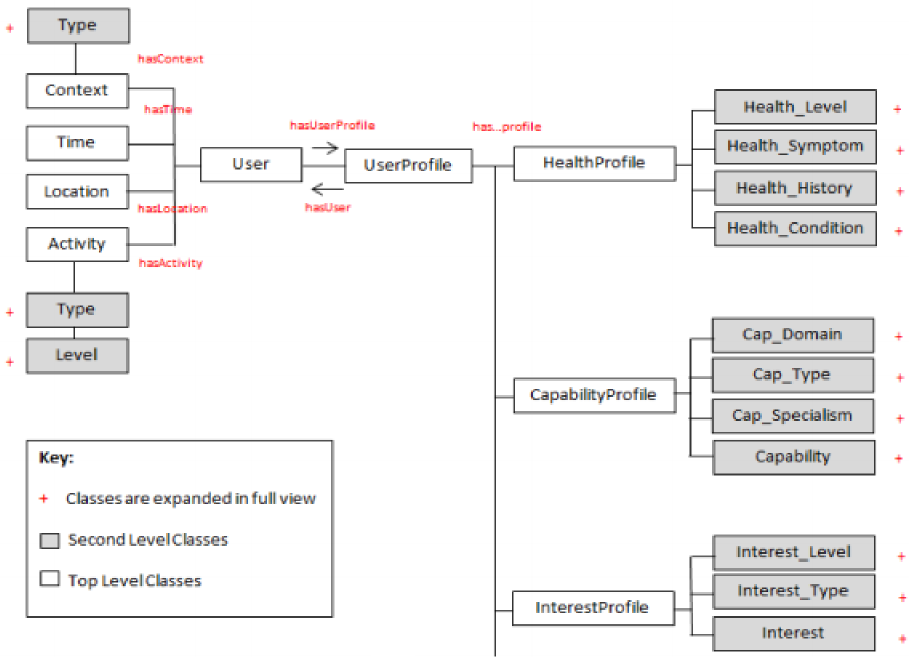

Figure 5.4: Extract of User Profile Ontology

Figure 5.4 shows an extract of the user profile ontology, including its hierarchical structure. One of the advantages of using a hierarchical structure is that knowledge of one user attribute can lead to the inference of another user attribute. If we know that the user has a health condition, we also know that they might have lower cognitive or physical capability. By having an is-a link between the class HealthProfile and CapabilityProfile, we can, for example, infer that a person who has the health condition "broken leg", has a lower movement capability. This type of information is of high relevance in a nudging system.

An interesting future research area is therefore to investigate whether the user profile ontology suggested by Skillen et al. [15] can be further extended to better support a nudging system. By collaborating with other scientific fields, several interesting connections might be found. Maybe two new linked classes can be personality traits and general stress level. If we have inferred that the user is high in neuroticism, we can infer that the user has a general high stress level. This is, of course, an assumption that we can not be sure of and must be made in collaboration with an expert in a relevant field.

5.6 Summary

I have, in this chapter, introduced a novel smart nudge design contributed by user profiles inferred from smartphone sensor/context data. The design was motivated by findings from chapter 4, which indicates that smartphone sensor/context data can be a powerful tool to infer user profiles.

The proposed smart nudge design reveals several reasons why such a system is useful. The advantages of using extracted features and inferred user profiles are thought to be unique to this system. Both features extracted from sensor/-context data and inferred attributes can be used to decide when to nudge, to build nudges, and to evaluate nudges. Movement data and geolocation are examples of such features. Personality traits and mood are examples of such attributes.

6 Implementation

An implemented prototype app is outlined in the following sections. Parts of a smart nudging system in the smartphone domain are implemented. Data collection, feature extraction, and user profile modeling are implemented. It describes how smartphone sensor/context data can be collected and used to extract features and infer user profiles relevant for a nudging system.

The first screen of the app is a typical register/login screen. When registering, a user id is created for the user connected to the email and password provided. The main screen is a step counter continuously updating as the user walks. The step counter is reset every day. The user will therefore be able to see daily steps on the main screen.

6.1 System Architecture

Figure 6.1 illustrates the implemented system. The authentication service provides a way of uniquely identifying users and creates a unique id tied to each user.

The feature store stores app features and movement features extracted from collected sensor/context data. The user profile stores user attributes found by analyzing the extracted features. User attributes inferred include user interest and activity level.

User interest was inferred by analyzing categorized app data. This was achieved by defining certain categories like "Beauty" as an interest. The number of apps and usage of a category was used to decide whether the user holds the interest or not.

Activity level was inferred from the extracted feature average step count. The average step is defined as the average daily steps the user walks over two weeks. Activity level is split into the levels: "high activity", "moderate activity", and "low activity".

The user modeler is implemented as a python server. It monitors the feature store in a publish-subscribe manner. This means that every time the client device pushes information, the Python server is notified. The user modeler extracts features and maintains the user profile.

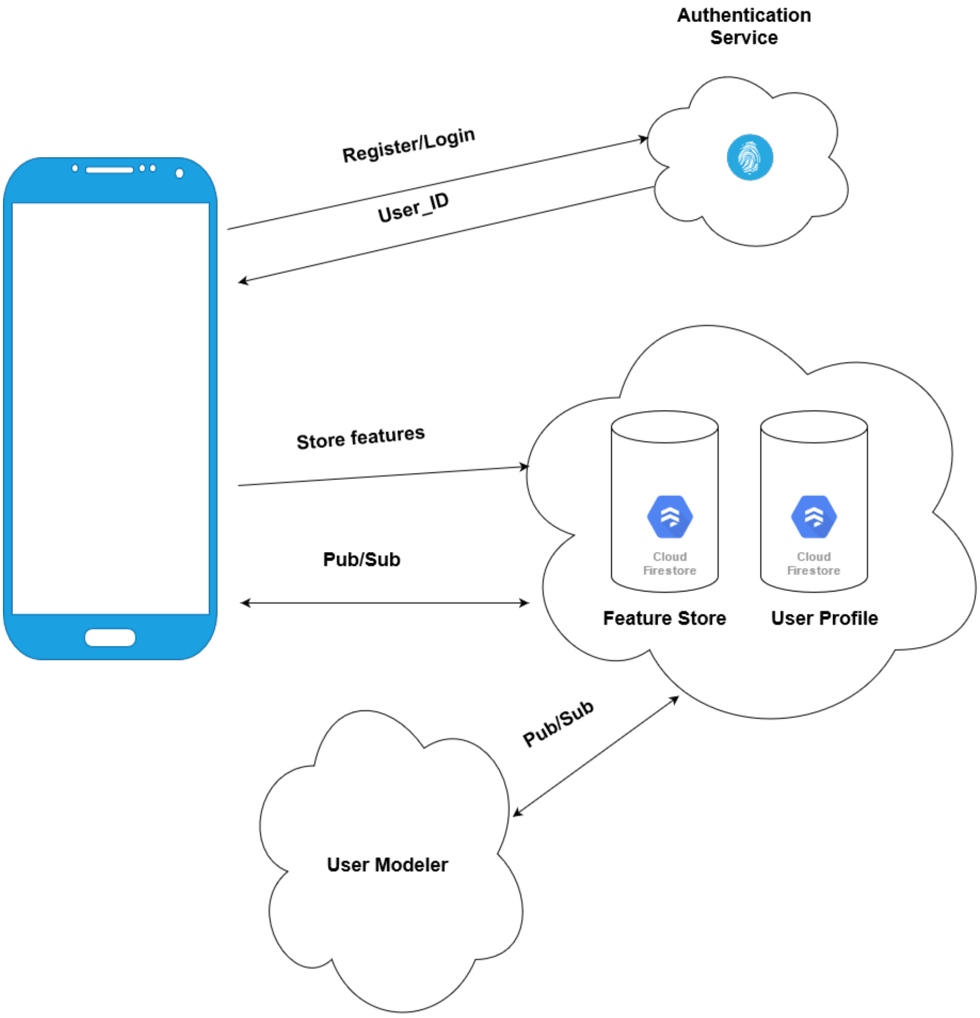

Figure 6.1: System architecture

The described architecture is part of a smart nudge system. The nudging aspects are however not implemented yet. Figure 6.2 shows how a complete implemented system would look like. Here a nudge store contains previously issued nudges. The nudge handler decides when to nudge, builds nudges, evaluates nudges, and sends nudges to the client smartphone. This is how a system can be expanded in the future.

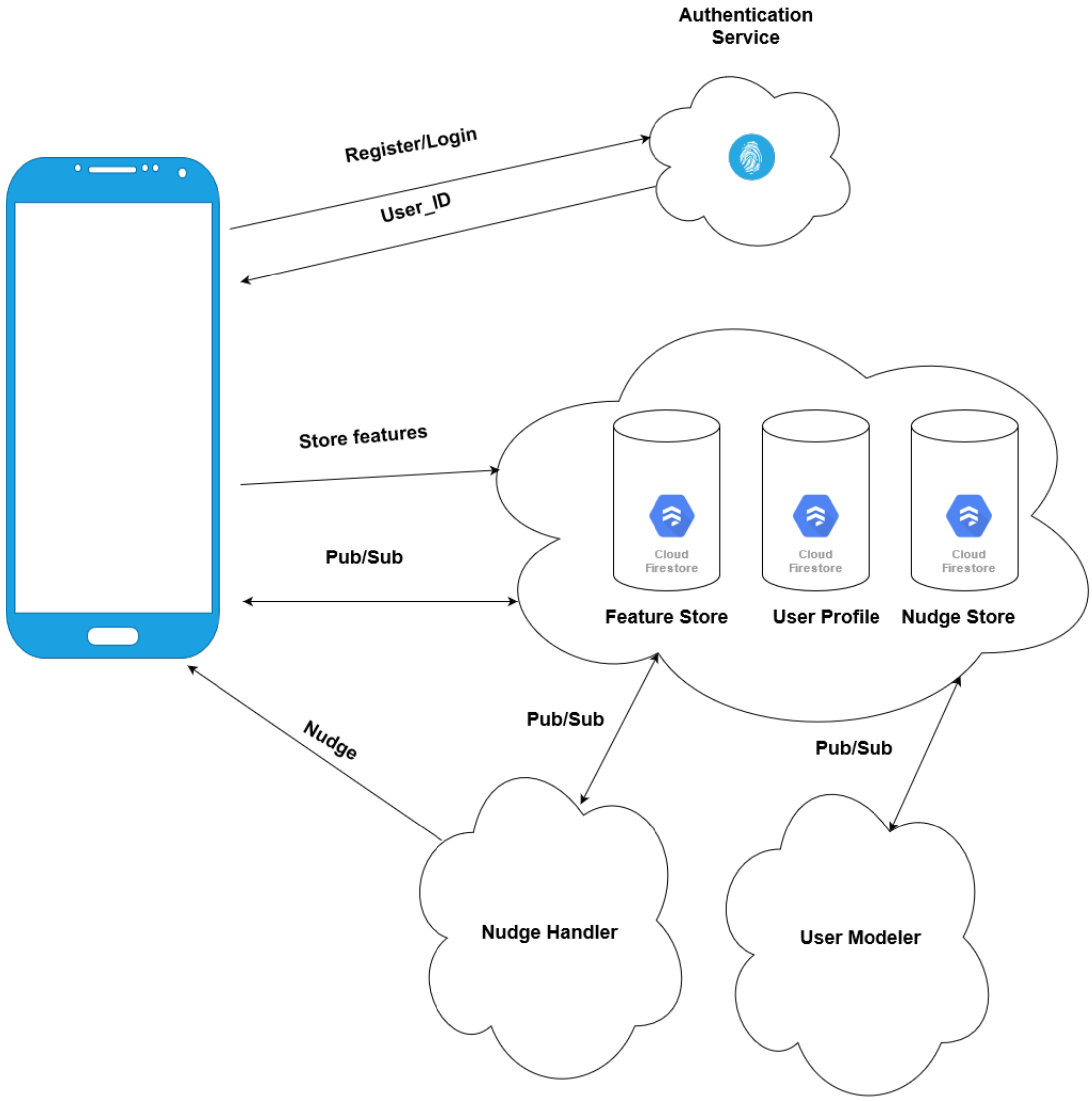

Figure 6.2: Extended System architecture

6.2 Enabling Technologies

The prototype app was realized using several enabling technologies, most of them provided by Google's Firebase.

6.2.1 Flutter

The smartphone app was realized using Flutter. Flutter is a software development kit used to create cross-platform apps for IOS and Android[63]. It is developed by Google and uses the programming language Dart[70].

6.2.2 Firebase

Firebase is Google's mobile platform[71]. It is used when creating mobile apps, and can among other things, store user information and authenticate users. A main motivation for picking Firebase is its integration with Flutter.

Firebase Authentication

Firebase Authentication[72] provides a way of uniquely identifying users. The user registers via email and password, which creates a unique id. This unique id is typically used to create and identify documents storing user information.

Firestore

Firestore[73] is a document-based database provided by Google. It consists of collections and documents. A collection contains a specific type of document. Each document can point to a sub-collection consisting of another type of document, resulting in a deeply nested tree-like structure.

Firestore provides the possibility of listening to document changes in a publish/subscribe manner. This was one of the main motivations for picking the technology as both the client smartphone and a server can notice real-time changes.

6.3 Data Collection

Installed app list, app usage data, and pedometer data are collected from the client device.

6.3.1 App Data

The app only runs on Android devices, even though flutter allows for cross-platform applications. This is because app data is only possible to collect from android devices.

Android devices make the act of collecting installed app list and app usage data trivial tasks. Both installed app list, and app usage data are stored on android devices and can be retrieved by using a library. In this implementation, a flutter library was used to collect both types of data.

6.3.2 Pedometer Data

Pedometer data was collected on the client device and is calculated from accelerometer data. As discussed in the design chapter, features such as pedometer data can be extracted on a client device or a server.

In this implementation, step count is extracted on the client smartphone and is pushed as "daily steps" every day and as "hourly steps" every hour. There already exist libraries to calculate step counts, one of which was picked for this implementation. By calculating step counts on the client device, less storage and energy is used. It also lessens the complexity of the code, as the calculation of step counts is done by the imported library.

Although not taking advantage of the hourly step count as of yet. It can be used as context in a nudging system to decide if a nudge should be issued or not.

6.4 Data Storage

Firestore is used as the main storage unit in this system. The feature store and user profile are stored here. The feature store contains extracted features, some of which are extracted on the client device and some of which are extracted on the server. The user profile contains inferred user attributes.

6.4.1 Data Model

There are several data models for storing data in Firestore. One way is to store all data in one collection. Then data concerning a user would be stored in one document in a JSON style format. This model, however, does not scale well as there is a size limit of 1 MB per document.

Another model exploits sub-collections. This approach allows the feature store to be a top-level collection, with the user profile as a sub-collection. In this approach, however, the upper collection is fetched whenever the sub-collection is fetched.

The model picked for this implementation is having two upper-level collections. These collections are connected through the user id. This means that each user has one document located in the feature collection and one located in the user profile collection, both named after the user id. This model is picked as it splits the user profile and feature store, allowing one to be fetched without the other to be fetched. This is advantageous as there are cases where only one of the stores is queried or changed.

6.4.2 Feature Store

The feature store contains pedometer features and app features. An example feature store is depicted in Figure 6.3. The feature store contains:

- Hourly steps: hourly steps are pushed hourly by the client device. This feature is not used yet but can be used as context to decide if the user should be nudged.

- Day: The day is stored to make the system adaptable. We want the user profile to reflect how the user is at the moment, and therefore only take features from the last two weeks into account when maintaining the user profile.

- avgSteps: Average daily steps over the two past weeks are stored to find the activity level of the user.

- appCat: app categories are a collection of categories that can be translated into interest. It contains how many apps from the installed app list fall under the category and how much time the user has spent on apps falling under the category.

- appUsage: The app usage feature is extracted on the client device. It contains the installed app list and usage of each app. The app package name is stored. This is because the package name can be used to query the Google play store to find which category the app belongs to.

- twoWeekSteps: Daily steps over the two past weeks are stored and are used to extract avgSteps. Note that Figure 6.3 only contains three days of steps. This should, in reality, contain 14 days but is not included in the figure to save space.


```
{
  "Fetaures": {
    "id": "1231000aasdfa",
    "hourlySteps": 100,
    "day": 12,
    "appCat": {
      "Art & Design": {
        "num": 4,
        "usage": 120
      },
      "Game": {
        "num": 7,
        "usage": 300
      },
      "Lifestyle": {
        "num": 2,
        "usage": 0
      }
    },
    "appUsage": {
      "com.facebook.katana": 0,
      "com.h8games.DoodleRun3D": 60,
      "com.ballzy.game": 20,
      "com.sec.android.app.popupcalculator": 0
    },
    "twoWeekSteps": {
      "1": 1500,
      "2": 3000,
      "3": 4500
    }
  }
}
```


Figure 6.3: Example feature store

6.4.3 User Profile

The user profile contains inferred user attributes. Namely, the attributes: activity level and interests. An example user profile is depicted in 6.4. The user profile is connected to the feature store from figure 6.3. Both the interest "Game" and "Art and Design" were inferred from the fact that the user has installed a specific amount of apps underlying each category and has used these apps for some hours.

Activity level was inferred from the average step count feature. The user has a low activity level, meaning that they move less than a threshold for being specified as moderately active.


```
{
  "User_profile": {
    "id": "1231000aasdfa",
    "interests": [
      "Game",
      "Art & Design"
    ],
    "activity_level": "low_activity"
  }
}
```


Figure 6.4: Example user profile

6.5 User Modeler

The user modeler extracts features and infers user attributes. The user modeler is on a server and listens for changes on the feature store. A logic is developed for inferring user profiles from extracted features.

6.5.1 Adaptivity

As discussed in the design chapter, the system needs to be adaptive to provide relevant nudges. In this system, the user profile is updated every two weeks. The average step count over the two weeks is used to infer activity level. Installed app list and app usage over the two weeks are used to infer user interest.

6.5.2 Feature Extraction

Feature extraction performed by the user modeler are extractions on already extracted features. Features such as installed app list, usage data, and daily step count are extracted on the client phone. These features are used to extract new features such as average step count and app categories.

Categorizing the apps is done by the user modeler. Every app in the installed app list is looked up to check if it is on the Google play store. If it is, the category is saved, and the number of apps in that category is incremented. The app's usage is also added to the category.

Querying the Google Play store helps categorize the apps and to ignore some of the pre-installed apps on the device. Pre-installed apps are a problem when modeling user profiles. This is because the pre-installed apps were not picked by the user and do not reflect the user's interest. Some of these apps were ignored by not finding them on the Google Play store.

Other pre-installed apps, however, are found on the Google Play store and therefore not noticed. There are ways to solve this problem, and for some phones, there exists a list of pre-installed apps. This list can be queried and ignored by the feature store. It is also worth pointing out that it would be easier to remove pre-installed apps if the app was implemented using Java. This, because there exists a library for noticing these apps.

In this implementation, the problem was solved by considering both the installed app list and app usage. As seen in literature[17] app usage is a better indicator of user interest than the installed app list, and unused apps are filtered out by defining a usage threshold the user must reach for him/her to be interested in the category.

6.5.3 User Modeling

Activity level is categorized into three possible activity levels. If the user averages 10,000 steps per day, he/she is categorized as very active. If he/she averages between 5000 and 10,000, he/she is categorized as moderately active. The person is categorized as little active if he/she averages less than 5000 steps a day.

A similar strategy is used to infer user interest. A sub-set of the listed Google play categories is picked to signify interest. Categories such as "Beauty" and "Books" are picked to signify interest. Categories such as "Tools" and "Weather" are not included, as they can not be translated into an interest. Games are also thought of as an interest, and all game categories are merged into the interest "Game". Which categories are translated to interests and which are not, are depicted in the two following lists.

Can be translated into interest :

- Auto and Vehicles
- Beauty
- Books and Reference
- Business
- Comics
- Dating
- Education
- Entertainment
- Events
- Finance
- Food and Drink
- Health and Fitness
- House and Home
- Lifestyle
- Medical
- Music and Audio
- News and Magazines
- Parenting
- Photography
- Productivity
- Shopping
- Social
- Sports
- Travel and Local

Can not be translated into interest :

- Libraries and Demo
- Maps and Navigation
- Personalization
- Tools
- Video Players and Editors
- Weather

Interests are as described in chapter 5 of value in a smart nudge system. Several of the inferred interests in the implemented system can contribute to a smart nudge system. The category "social", for example, suggests that the user can be affected by social nudges. The category "Health and Fitness" suggests that the user can be affected by activity nudges.

A user is thought to be interested in a category based on a combination of app usage data and the number of apps installed belonging to the category. In this implementation, the user is interested in a category if he/she has more than three apps installed from the category, and he/she has used apps from the category for more than 60 minutes the past two weeks.

7

Discussion

In this chapter, user profiling in a smartphone domain is discussed. The emphasis lies on how user profiling from smartphone sensor/context data can contribute to a nudging system. Challenges connected to user profiling in the smartphone domain discovered through relevant research are also discussed.

7.1 Challenges: User Profiling from smartphone Data

Some of the most prominent challenges discovered in relevant research are discussed in this section.

7.1.1 Population Bias

A large data set is often needed to make a reliable user profile model. This data can either be gathered from users of the system or a third party. There exist several large databases on, for example, app usage. Using a third-party data set to train the model might not be as advantageous as it seems.

Peltonen et al. [39] found by analyzing 25,323 android users from 44 countries that the demographic attribute that gave the most information gain was what country they lived in. The authors tested this by performing a survey on 3293 individuals. They created a binary category vector for each user based on whether a category was used or not. They used Kullback-Leibler divergence[74] to find similarities in probability distributions of different countries.

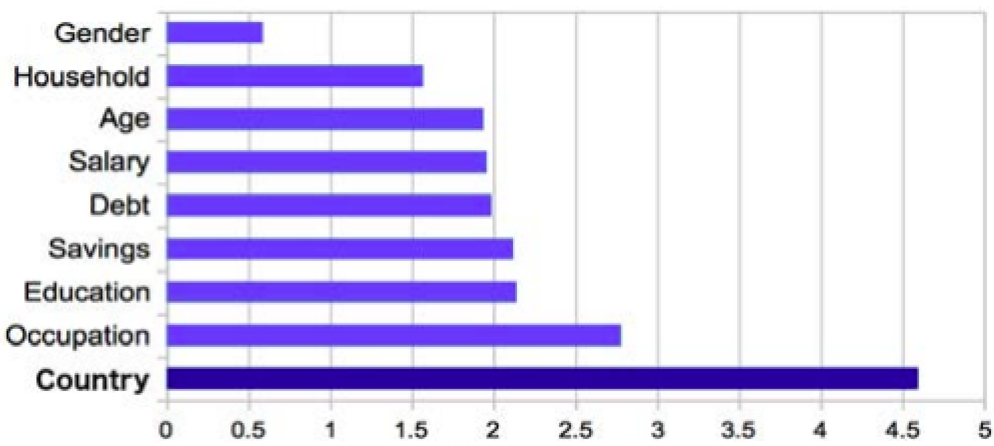

Figure 7.1: Information gain from different demographic attributes

They found that country gave significantly more information about app usage than any other demographic attribute. As illustrated in 7.1 country gave almost twice as much information gain as occupation. This suggests that individuals from different countries will have widely different app usage habits. They also found that certain demographics such as being a student were similar independent of country.

Based on the findings from Peltonen et al. [39] I suggest that a model should not be trained from third-party data. This, because the data may not represent the userbase of the app.

This has other implications as well. It means that findings from other researchers, such as Yu et al.[57] who predicted gender from app usage, among other features, might not be relevant in other systems. They collected all their data from Chinese individuals. One of their findings is that males rarely installed more than seven photo editing apps. This fact was used to predict gender. This fact might, however, only hold for Chinese individuals.

The model should therefore be trained based on data collected from the users of the nudging system. A predefined logic should be set in place to provide relevant nudges until the system adapts to the users.

7.1.2 Ground truth Data Collection

Several techniques discussed in chapter 4 uses supervised machine learning techniques to predict user attributes. This requires a set of ground truth data. As already discussed, this should be collected to represent the users of the system. The collection of ground truth data can, however, be a difficult task. Users often need to fill out forms that are prone to personal bias and fake information. If the user is aware of the ground truth data collection, they might also act differently than they would normally. If they know that activity data is being collected, for example, the user might be more active to appear more active. If app usage data is being collected, the user might avoid using certain apps they are embarrassed to use.

7.1.3 Power Consumption

Before starting work on the thesis, I presumed that a problem would be the power consumption of continuously collecting sensor data from sensors such as accelerometer, GPS, gyroscope, and magnetometer.

This problem, however, seems to be less of an issue than initially thought. Wang et al. [48] found that their app consumes relatively little energy. Their app drops the battery to 15 percent in about 13 hours. They argue that users will be affected by energy consumption if the battery runs out in less than 10 hours, which they claim is the typical hour of usage for a day. Yu et al. have similar findings. Their app, which collected a range of information from a range of sensors, used less than 6 percent of the power when the user used their phone normally.

If power consumption is a problem, however, there exist solutions to solve the problem. CenceMe, for example, uses low rate sampling to save energy and switch to a higher rate when an interesting event occurs. They also compress data before sending it to a server.

Another example is Ben-Zeev et al. [51] who turned on sensors periodically to collect data. When collecting speech data, the sensor would only keep collecting if it noticed speech using a speech detection system.

7.2 How can User Profiling from Smartphone Sensor/Context Data Contribute to a Smart Nudge System?

The research review in chapter 4, the design in chapter 5, and implementation in chapter 6 revealed several advantages of a smart nudge system based on smartphone sensor/context data.

7.2.1 Data Collection

Inferring user profiles from sensor/context data is a type of implicit data collection. Here, user profiles are inferred by modeling features extracted from sensor/context data. The alternative is the explicit approach, where the user is prompted to fill out forms with his/her information.

Explicit Approach

The explicit approach is still widely used to infer user profiles. This approach is especially useful when inferring static attributes such as age and gender but is also sometimes used to infer more dynamic attributes such as interests.

The explicit approach, however, is lacking for several reasons:

- Explicit data collection is prone to fake user profiles[16]. The user is often concerned with his/her privacy and might purposefully provide false information to protect it. Users also find the process of filling out forms tedious and might fill out wrong information to rush the process. Users might not believe that the personalization the app can provide outweighs their need for privacy and the time it takes to fill out the forms.

- The user profile is prone to subjective bias. In nudging systems, personal information such as mood, mental health information, and personality traits are of high relevance. The user might have wrong insights into these types of information and might be prone to bias. Inferring these types of attributes from objective sensor data might lead to more correct user profiles.

- Maybe the most relevant disadvantage of explicit user profiles is its lack of adaptability. Explicit data collection is only relevant for static attributes such as demographic attributes that do not change over time. Explicit data collection of dynamic attributes such as mood will become overly imposing on users, as it will need to be filled out several times during the day.

There are, however, several cases where the explicit approach might be preferred over the implicit approach. Static attributes such as age and gender for example should, in my opinion, be inferred explicitly. Consider a system that infers a range of static and dynamic attributes. Having to build a computational model with ground truth data for each attribute is thought to be quite a complex task. In a trade-off between building a complex computational model and simply asking the user for his/her age and gender, I believe that the explicit approach is advantageous, even with the risk of false information.

Implicit Approach

In this thesis, the implicit approach is picked to infer user profiles. This was motivated by the wide range of sensors found in today's smartphones, which, as described in chapter 4, have been used to infer user profiles with high accuracy.

Implicit data collection holds several prominent advantages.

- Implicit data collection allows for inference of both static and dynamic attributes with high precision as described in chapter 4, and thus provides the opportunity of creating adaptive user profiles.

- Implicit data collection from smartphone sensor/context data entails the extraction of different types of features. Features such as movement features and location features can be used to find user context, which is of high relevance in a nudging system.

- Most users will carry their smartphone where ever they might go. This provides the advantage of continuous data collection, which can be used to develop more accurate user profiles.

One of the main drawbacks of implicit data collection is that it, in many cases, relies on ground truth data and thus might cause some of the same problems found in explicit data collection. Personal bias is one of these problems.

Some techniques in chapter 4 built models based on data collected from each user. Moodscope[40], for example, tested personalized models, an all-user model, and a hybrid model. Personalized models mean that a model is built for each user. This technique implies that each user has to provide ground truth data. In MoodScope, users had to provide mood samplings several times a day for two months before the model reached an accuracy of 93 percent.

The all-user model aimed to build one model to fit all users. This model led to an accuracy of around 66 percent, significantly worse than the personalized models. The hybrid model used a combination of the all-user model and the personalized model. Here, less ground truth data was collected from users to form the personalized model. This approach was more accurate than the all-user model but did not outperform the personalized model.

All though some user attributes such as mood require a personalized model to provide high accuracy predictions, others can be inferred by using a general model for all users. Yu et al. [57], for example, were able to predict gender with 91.7 percent accuracy by building a general model used on all users.

It is worth noting that implicit data collection can be performed from other data than smartphone sensor/context data. It is, in fact, more normal to do this from sources such as social media. This does, however, not provide some of the advantages that smartphone sensor/context data does. Some examples are movement and location features and continuous data collection.

7.2.2 Nudge Decision

User profiling from smartphone sensor/context data provides several advantages when deciding whether to nudge or not. Both the adaptive user profiles and extracted features can be used to find user context. User attributes such as mood and stress level are of high relevance when deciding if a nudge should be sent. Movement and location features are also of high relevance. Knowing if the user is moving and where he/she is, provides powerful information in deciding if the user should be nudged.

7.2.3 Nudge Building

Both attributes and features are of relevance when building nudges. Research shows that personalized push notifications are more likely to affect the user than non-personalized push notifications[75][60]. Movement and location features can be used to personalize push notifications. If we want the user to eat more healthily and know his/her location, we can send the push notification: "You are close to a five-starred vegetarian restaurant. Why not give it a try?". The movement features could be used to encourage the user to be more active with a push notification like: "This has been one of your most active weeks. Keep it up!".

Attributes such as interests can be used to personalize the content of a nudge. If we know that the user is interested in a certain type of sports and the nudge goal is to make the user more active, the nudge could suggest a close-by sporting event.

7.2.4 Nudge Evaluation

Extracted features can be used to evaluate the nudge without the need of asking for user feedback. Movement and location features can, for example, tell if the user went for the walk he/she was nudged to do, and if the user went to the restaurant, he/she was nudged to visit.

7.2.5 The Smartphone as a Medium to Receive Nudges

The smartphone is an ideal medium to receive nudges on. This is because the user always carries it around and because the device can receive push notifications. Other digital mediums such as the computer require the user to actively use it. It is also less common to receive push notifications on a computer than on a smartphone.

Other digital mediums to receive nudges on are IoT devices. Devices such as smartwatches, smart fridges, and dashboard screens in a car are such mediums. Most of these devices, however, are less available than the smartphone and are only available for nudges on rare occasions.

7.3 Evaluation of Implementation

A user profiling system was implemented. App features and movement features were collected and analyzed to infer user interest and activity level.

A logic was set in place to infer the user attributes. The user was categorized into a specific activity level based on average steps taken in the last 2 two weeks, and the user was thought to be interested in a category based on the installed app list and app usage over the two last weeks.

This technique might be a bit naive when inferring attributes. Many variables may come to play that are not considered. For activity level, for example, the user might be very active without being a person who walks or runs a lot. He/she might, for example, use a bicycle, go to the gym, go mountain climbing, and so on. These types of exercise can be noticed by implementing activity recognition and by using geolocation to notice when the user is at the gym.

The translation from categories to interests might also be a bit naive. It is only able to notice interest that can be translated from the relatively small number of categories found from the Google Play store(31 categories). An improvement can be to extract content-based features as Seneviratne et.al[37] did when predicting gender from the installed app list. They gathered app descriptions from the Google Play store and performed tf-idf to find the top terms in the descriptions. Maybe this technique can be used to extract more "categories" that can be translated into interests.

It is also worth mentioning that most techniques described in chapter 4 used some type of machine learning or statistical method to predict user attributes. This was not implemented in this system. These types of methods are, as described, often highly reliable. These techniques, however, often require a lot of time and computational power to train the models. These systems also tend to become quite complex.

These things need to be considered when implementing the system. For static attributes such as age, gender, and marital status, it might be more advantageous to ask the user than to build complex machine learning models. For other tasks such as inferring mood and personality traits, however, machine learning is more appropriate.

7.4 Evaluation of Research Method

This thesis was conducted using qualitative research. The notion that user profiles inferred from smartphone sensor/context data can contribute to a smart nudge system was investigated through several sources of information. A research review was conducted, a smart nudge design was proposed, and a system for inferring user profiles from smartphone sensor/context data was implemented. Findings from these sources suggest that user profiles inferred from smartphone sensor/context data can greatly contribute to a smart nudge system. There are, however, limitations to how much this notion can be supported using qualitative research methods.

A limitation with the chosen research methods is that no user testing was conducted. This means that there is no actual way of knowing whether such a system will work in practice.

7.5 Future work

The implemented system infers user profiles from smartphone sensor/context data but does not use this data for nudging. A natural future work is, therefore, to implement the remaining components of the smart nudge system proposed in chapter 5. Then nudges can be issued and be built based on activity level and interests.

Other future work includes inferring more types of user attributes. Personality traits and mood, for example, are attributes that are thought to be of great value in a smart nudge system. Both of these attributes have been inferred by other researchers by using supervised machine learning on app usage records[57][40].

The limitations of the chosen research methods encourage user testing as future work. This can strengthen the belief that user profiles inferred from smartphone sensor/context data can contribute to a smart nudge system.

8 Conclusion

This thesis examined how user profiles inferred from smartphone sensor/context data can contribute to a smart nudge system. First, relevant research concerning user profile inference from smartphone sensor/context data was examined. Relevant research showed that smartphone sensor/context data can accurately predict static user attributes such as age, gender, personality traits, and dynamic user attributes such as mood and stress level. Research also showed how this type of information can contribute to research areas such as physical and mental health.

A smart nudge design contributed by user profiles inferred from smartphone sensor/context data was proposed. Several advantages of such a system were discovered, some of which are thought to be unique to this design. The unique advantages include the inference of highly adaptable user attributes such as stress level and mood, which can be inferred implicitly. These attributes provide user context. User context is also provided by extracted features such as movement data and geolocation. These attributes and features can be used to decide when to nudge, build personalized nudges, and evaluate nudges.

Smartphones are always close to their users and are, as a result, open to receive nudges at most times. Smartphones, being close to their users, also provides the possibility of continuous data collection.

A prototype system was implemented that infers user profiles from smartphone sensor/context data. The implemented system gave insight into how user attributes such as interests and activity level can be inferred implicitly without the need for machine learning, as often used in research.

Problems tied to implicit user profiling from smartphone sensor/context data were identified. The most prominent issues seem to be ground truth data collection and population bias. It is important to keep these issues in mind when implementing the proposed design. I believe that it is advantageous to collect ground truth data from the users of the system rather than using a data set from a third party. This, to make sure that the data represents the users of the system.

The proposed smart nudge design seems promising but can in future work be further supported through implementing the remaining components of the proposed design and through user testing.

Bibliography

[1] F. J. Penedo and J. R. Dahn, "Exercise and well-being: A review of mental and physical health benefits associated with physical activity," *Current opinion in psychiatry*, vol. 18, no. 2, pp. 189–193, 2005.

[2] R. H. Thaler and C. R. Sunstein, *Nudge: Improving decisions about health, wealth, and happiness*. Penguin, 2009.

[3] *Number of smartphone users worldwide from 2016 to 2021*, https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, Accessed: 2021-03-02.

[4] *The top 10 causes of death*, https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death, Accessed: 2020-28-05.

[5] A. Håkansson, "Portal of research methods and methodologies for research projects and degree projects," in *The 2013 World Congress in Computer Science, Computer Engineering, and Applied Computing WORLDCOMP 2013; Las Vegas, Nevada, USA, 22-25 July*, CSREA Press USA, 2013, pp. 67–73.

[6] *The world's first smartphone, simon, was created 15 years before the iphone*, https://www.businessinsider.com/worlds-first-smartphone-simon-launched-before-iphone-2015-6?r=US&IR=T, Accessed: 2020-28-05.

[7] *Meaning of sensor in english*, https://www.lexico.com/definition/sensor, Accessed: 2021-03-02.

[8] *Sensors on a phone/tablet and their functions*, https://consumer.huawei.com/eg-en/support/content/en-us00685236/, Accessed: 2021-03-02.

[9] *Satellite navigation - gps - how it works*, https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/howitworks/, Accessed: 2021-07-02.

[10] *What is an accelerometer?* https://www.fierceelectronics.com/sensors/what-accelerometer, Accessed: 2021-10-02.

[11] *Gyroscope*, https://www.lexico.com/definition/gyroscope, Accessed: 2021-28-05.

[12] *Gyroscope*, https://www.britannica.com/technology/gyroscope, Accessed: 2021-28-05.

[13] *Magnetometer*, https://www.dictionary.com/browse/magnetometer, Accessed: 2021-28-05.

[14] S. Majumder and M. J. Deen, "Smartphone sensors for health monitoring and diagnosis," *Sensors*, vol. 19, no. 9, p. 2164, 2019.

[15] K.-L. Skillen, L. Chen, C. D. Nugent, M. P. Donnelly, W. Burns, and I. Solheim, "Ontological user profile modeling for context-aware application personalization," in *International conference on ubiquitous computing and ambient intelligence*, Springer, 2012, pp. 261–268.

[16] C. I. Eke, A. A. Norman, L. Shuib, and H. F. Nweke, "A survey of user profiling: State-of-the-art, challenges, and solutions," *IEEE Access*, vol. 7, pp. 144 907–144 924, 2019.

[17] S. Zhao, S. Li, J. Ramos, Z. Luo, Z. Jiang, A. K. Dey, and G. Pan, "User profiling from their use of smartphone applications: A survey," *Pervasive and Mobile Computing*, vol. 59, p. 101 052, 2019.

[18] *Data preprocessing*, `https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.467.29&rep=rep1&type=pdf`, Accessed: 2021-28-05.

[19] *Pandas*, `https://pandas.pydata.org/`, Accessed: 2021-28-05.

[20] *What is feature extraction?* `https://deepai.org/machine-learning-glossary-and-terms/feature-extraction`, Accessed: 2021-10-02.

[21] *Machine learning*, `https://www.ibm.com/cloud/learn/machine-learning`, Accessed: 2021-12-02.

[22] *Supervised vs. unsupervised learning: What's the difference?* `https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning`, Accessed: 2021-28-05.

[23] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[24] O. Kramer, "K-nearest neighbors," in *Dimensionality reduction with unsupervised nearest neighbors*, Springer, 2013, pp. 13–23.

[25] J. Joyce, "Bayes' theorem," 2003.

[26] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, 2001, pp. 41–46.

[27] *Svm (support vector machines)*, https://machine-learning-and-data-science-with-python.readthedocs.io/en/latest/assignment4$_s up_m l.html$.

[28] W. S. Noble, "What is a support vector machine?" *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[29] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[30] W. S. Sarle, "Neural networks and statistical models," 1994.

[31] *What is ontology in computer science*, `https://www.igi-global.com/dictionary/ontology-in-computer-science/21124`, Accessed: 2021-12-02.

[32] C. R. Sunstein, "The ethics of nudging," *Yale J. on Reg.*, vol. 32, p. 413, 2015.

[33] ——, “Nudging and choice architecture: Ethical considerations,” *Yale Journal on Regulation, Forthcoming*, 2015.

[34] ——, “Do people like nudges,” *Admin. L. Rev.*, vol. 68, p. 177, 2016.

[35] M. Weinmann, C. Schneider, and J. Vom Brocke, “Digital nudging,” *Business & Information Systems Engineering*, vol. 58, no. 6, pp. 433–436, 2016.

[36] R. Karlsen and A. Andersen, “Recommendations with a nudge,” *Technologies*, vol. 7, no. 2, p. 45, 2019.

[37] S. Seneviratne, A. Seneviratne, P. Mohapatra, and A. Mahanti, “Your installed apps reveal your gender and more!” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 18, no. 3, pp. 55–61, 2015.

[38] J. Ramos *et al.*, “Using tf-idf to determine word relevance in document queries,” in *Proceedings of the first instructional conference on machine learning*, Citeseer, vol. 242, 2003, pp. 29–48.

[39] E. Peltonen, E. Lagerspetz, J. Hamberg, A. Mehrotra, M. Musolesi, P. Nurmi, and S. Tarkoma, “The hidden image of mobile apps: Geographic, demographic, and cultural factors in mobile usage,” in *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2018, pp. 1–12.

[40] R. LiKamWa, Y. Liu, N. D. Lane, and L. Zhong, “Moodscope: Building a mood sensor from smartphone usage patterns,” in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, 2013, pp. 389–402.

[41] J. A. Russell, “A circumplex model of affect.,” *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[42] R. M. Frey, R. Xu, and A. Ilic, “Reality-mining with smartphones: Detecting and predicting life events based on app installation behavior: Research-in-progress,” *ICIS 2015 Proceedings*, 2015.

[43] S. L. Lim, P. J. Bentley, N. Kanakam, F. Ishikawa, and S. Honiden, “Investigating country differences in mobile app user behavior and challenges for software engineering,” *IEEE Transactions on Software Engineering*, vol. 41, no. 1, pp. 40–64, 2014.

[44] A. M. Khan, Y.-K. Lee, S.-Y. Lee, and T.-S. Kim, “Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis,” in *2010 5th international conference on future information technology*, IEEE, 2010, pp. 1–6.

[45] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, “Complex human activity recognition using smartphone and wrist-worn motion sensors,” *Sensors*, vol. 16, no. 4, p. 426, 2016.

[46] I. da Penha Natal, R. d. A. C. Cordeiro, and A. C. B. Garcia, “Activity recognition model based on gps data, points of interest and user profile,” in *International Symposium on Methodologies for Intelligent Systems*, Springer, 2017, pp. 358–367.

[47] E. B. Zion and B. Lerner, "Identifying and predicting social lifestyles in people's trajectories by neural networks," *EPJ Data Science*, vol. 7, no. 1, p. 45, 2018.

[48] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Wang, "Friendbook: A semantic-based friend recommendation system for social networks," *IEEE transactions on mobile computing*, vol. 14, no. 3, pp. 538–551, 2014.

[49] A. Giddens, *Modernity and self-identity: Self and society in the late modern age*. Stanford university press, 1991.

[50] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[51] D. Ben-Zeev, E. A. Scherer, R. Wang, H. Xie, and A. T. Campbell, "Next-generation psychiatric assessment: Using smartphone sensors to monitor behavior and mental health.," *Psychiatric rehabilitation journal*, vol. 38, no. 3, p. 218, 2015.

[52] A. Gaggioli, G. Pioggia, G. Tartarisco, G. Baldus, D. Corda, P. Cipresso, and G. Riva, "A mobile data collection platform for mental health research," *Personal and Ubiquitous Computing*, vol. 17, no. 2, pp. 241–251, 2013.

[53] *Photoplethysmography (ppg)*, `https://www.news-medical.net/health/Photoplethysmography-(PPG).aspx`, Accessed: 2021-15-03.

[54] E. C. Larson, T. Lee, S. Liu, M. Rosenfeld, and S. N. Patel, "Accurate and privacy preserving cough sensing using a low-cost microphone," in *Proceedings of the 13th international conference on Ubiquitous computing*, 2011, pp. 375–384.

[55] N.-C. Chen, K.-C. Wang, and H.-H. Chu, "Listen-to-nose: A low-cost system to record nasal symptoms in daily life," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 590–591.

[56] K.-L. Skillen, L. Chen, C. D. Nugent, M. P. Donnelly, and I. Solheim, "A user profile ontology based approach for assisting people with dementia in mobile environments," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2012, pp. 6390–6393.

[57] Z. Yu, E. Xu, H. Du, B. Guo, and L. Yao, "Inferring user profile attributes from multidimensional mobile phone sensory data," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5152–5162, 2019.

[58] R. R. McCrae and O. P. John, "An introduction to the five-factor model and its applications," *Journal of personality*, vol. 60, no. 2, pp. 175–215, 1992.

[59] R. M. Frey, R. Xu, and A. Ilic, "Mobile app adoption in different life stages: An empirical analysis," *Pervasive and Mobile computing*, vol. 40, pp. 512–527, 2017.

[60] B. Tan, K. Sangaralingam, V. K. Singh, C. S. Saripaka, and G. Manai, "Clairvoyant-push: A real-time news personalized push notifier using topic modeling and social scoring for enhanced reader engagement," in

2015 IEEE International Conference on Big Data (Big Data), IEEE, 2015, pp. 2913–2915.

[61] *Java documentation*, `https://docs.oracle.com/en/java/`, Accessed: 2021-08-04.

[62] *Swift*, `https://developer.apple.com/swift/`, Accessed: 2021-08-04.

[63] *Flutter*, `https://flutter.dev/`, Accessed: 2021-08-04.

[64] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "Cenceme–injecting sensing presence into social networking applications," in *European Conference on Smart Sensing and Context*, Springer, 2007, pp. 1–28.

[65] *The big five personality traits*, `https://www.verywellmind.com/the-big-five-personality-dimensions-2795422`, Accessed: 2021-08-04.

[66] *Curse of dimensionality*, `https://deepai.org/machine-learning-glossary-and-terms/curse-of-dimensionality`, Accessed: 2021-04-04.

[67] *Swift*, `https://numpy.org/`, Accessed: 2021-28-05.

[68] *Pandas*, `https://scikit-learn.org/stable/`, Accessed: 2021-28-05.

[69] S. Arlot, A. Celisse, *et al.*, "A survey of cross-validation procedures for model selection," *Statistics surveys*, vol. 4, pp. 40–79, 2010.

[70] *Dart*, `https://dart.dev/`, Accessed: 2021-08-04.

[71] *Firebase*, `https://firebase.google.com/`, Accessed: 2021-28-05.

[72] *Firebase authentication*, `https://firebase.google.com/docs/auth`, Accessed: 2021-28-05.

[73] *Cloud firestore*, `https://firebase.google.com/docs/firestore`, Accessed: 2021-28-05.

[74] J. R. Hershey and P. A. Olsen, "Approximating the kullback leibler divergence between gaussian mixture models," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, IEEE, vol. 4, 2007, pp. IV–317.

[75] N. Bidargaddi, D. Almirall, S. Murphy, I. Nahum-Shani, M. Kovalcik, T. Pituch, H. Maaieh, and V. Strecher, "To prompt or not to prompt? a microrandomized trial of time-varying push notifications to increase proximal engagement with a mobile health app," *JMIR mHealth and uHealth*, vol. 6, no. 11, e10123, 2018.