

LOW-POWER ARTIFICIAL INTELLIGENCE OF THINGS(AIOT) SYSTEMS

By

Li Liu

A DISSERTATION

Submitted to  
Michigan State University  
in partial fulfillment of the requirements  
for the degree of

Computer Science—Doctor of Philosophy

2023

## ABSTRACT

Internet-of-Things (IoT) is another excellent innovation after the Internet and mobile networks in the information era, aiming at connecting billions of end-devices across scales. A multitude of IoT applications often operate under conditions of constrained energy resources, which has rendered low-power IoT systems a subject of considerable research interest. The increasing need for AI in complex scenario-based composite tasks has led to the rise of Artificial Intelligence of Things(AIoT), which encompasses research in two major directions: AI for IoT that solves problems in IoT systems with AI techniques and IoT for AI that adopts IoT infrastructure/data to advance the development of AI models. While AIoT systems in low-power scenarios offer significant benefits, they also face specific challenges that are inherent to their design and operational requirements.

This dissertation delves into low-power AIoT from both angles. 1) We endeavor to harness the capabilities of AI to predict and analyse the communication channels of dynamic long links in LoRaWAN which is one of the Low-power Wide-area Networks(LPWANs). DeepLoRa adopts Deep Neural Networks based on Bi-directional LSTM(Long-Short-Time-Memory) to capture the sequential information of environmental influence on LoRa link performances for accurate LoRa link path-loss estimation. It reduces the path-loss estimation error to less than 4 dB, which is 2× smaller than state-of-the-art models. LoSee extends the contributions of DeepLoRa. It measures the real-world fine-grained performance, including detailed coverage study and feasibility analysis of fingerprint-based localization, of a self-deployed LoRaWAN system with temporal dynamics and spatial dynamics. 2) We design energy-efficient IoT systems that facilitate the deployment of AI models for practical applications. FaceTouch enables accurate face touch detection with a multimodal wearable system consisting of an inertial sensor on the wrist and a novel vibration sensor on the finger. We leverage a cascading classification model, including simple filters and a DNN, to significantly extend the battery life while keeping a high recall. FaceTouch achieves a 93.5% F-1 score and can continuously detect face-touch events for 79 – 273 days using a small 400 mWh battery depending on usage.

In general, this dissertation studies both theoretical and practical aspects in the field of low-

power AIoT systems, including LoRaWAN link behavior analysis and building practical wearable systems. These advancements not only underscore the feasibility of deploying low-power AIoT in real-world settings but also pave the way for future research and development in this domain, aiming to bridge the gap between IoT and AI for the creation of smarter, sustainable, and more efficient technologies.

Copyright by  
LI LIU  
2023

## ACKNOWLEDGEMENTS

I am profoundly grateful for the guidance, support, and opportunities provided to me during my journey as a Ph.D. candidate.

First and foremost, I would like to extend my deepest appreciation to Dr. Yunhao Liu, who not only gave me the chance to embark on this academic voyage but also led me into the enriching world of scientific research in the AIoT area.

My sincere gratitude goes to Dr. Zhichao Cao, my advisor during my Ph.D. program. His unwavering support, guidance, and patient tolerance have greatly facilitated my scientific pursuits. His mentorship has been a beacon of light, guiding me through the challenges and triumphs of academic research.

I am also deeply thankful to my graduate committee members, Dr. Li Xiao, Dr. Mi Zhang, and Dr. Tianxing Li. They have witnessed the milestones throughout my Ph.D. program. Their presence and input have been invaluable to my academic and personal growth.

To my parents, who have always been my sanctuary, my haven of warmth, and my steadfast support, my gratitude is boundless. They are the cornerstone of my strength, providing me with love and encouragement every step of the way. I wish them happiness, health, and consistent pride in me.

I owe a debt of gratitude to Manni Liu, my senior colleague and dearest sister without blood, who has been my comrade in this journey. Together, we have faced challenges and setbacks, forging an unbreakable bond in the pursuit of our dreams in life and scientific research. Our friendship is eternal, and we'll always be on the same side.

Special thanks to my best friend, Maolin Gan, who has been a source of joy, hope, and motivation in my life. Time is the touchstone of our friendship. Let's continue to be there for each other, find happiness in life, and shine together in our shared career path.

I also want to acknowledge my other friends who have generously supported and helped me along this journey. Their belief in me has made me stronger and more determined to move forward, reminding me that I deserve the best and can overcome any obstacle.

Lastly, but certainly not least, I want to thank myself for the courage to embrace the unknown and the perseverance to withstand the ups and downs. I hope to always believe in myself, love my career, cherish the people I meet in life, and continue to make further progress throughout the long journey of life.

## TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION . . . . .	1
1.1	Conducted Studies and Proposed Techniques . . . . .	2
1.2	Low-power And Accurate Face Touch Detection . . . . .	3
1.3	Organization . . . . .	5
CHAPTER 2	DEEPLORA: LEARNING ACCURATE PATH LOSS MODEL FOR LONG DISTANCE LINKS IN LPWAN . . . . .	6
2.1	Introduction . . . . .	6
2.2	Related Work . . . . .	9
2.3	Preliminary and Empirical Study . . . . .	11
2.4	System Design . . . . .	18
2.5	Experiment Details . . . . .	23
2.6	Evaluation . . . . .	25
2.7	Conclusion . . . . .	30
CHAPTER 3	IS LORAWAN REALLY WIDE? FINE-GRAINED LORA LINK- LEVEL MEASUREMENT IN AN URBAN ENVIRONMENT . . . . .	31
3.1	Introduction . . . . .	31
3.2	Related Work . . . . .	34
3.3	System and Dataset Overview . . . . .	36
3.4	Link Behavior Study . . . . .	41
3.5	Coverage Area Study . . . . .	47
3.6	Localization Accuracy Study . . . . .	50
3.7	Observations, Insight, and Discussion . . . . .	54
CHAPTER 4	FACETOUGH: PRACTICAL FACE TOUCH DETECTION WITH A MULTIMODAL WEARABLE SYSTEM FOR EPIDEMIOLOGICAL SURVEILLANCE . . . . .	56
4.1	Introduction . . . . .	56
4.2	System Overview . . . . .	61
4.3	Vibration-based Surface Touching Classification . . . . .	62
4.4	Wrist-IMU based Face-Touch Gesture Detection . . . . .	68
4.5	Implementation . . . . .	72
4.6	Evaluation . . . . .	74
4.7	Discussion and Future Work . . . . .	82
4.8	Related Work . . . . .	84
4.9	Conclusion . . . . .	85
CHAPTER 5	CONCLUSION . . . . .	87
BIBLIOGRAPHY	. . . . .	89

# CHAPTER 1

## INTRODUCTION

Internet-of-Things (IoT) is another excellent innovation after the Internet and mobile networks in the information era, aiming to connect billions of end-devices across scales, among which many devices operate in an energy-constraint unattended manner. The development of low-power IoT systems has thus become a focal point of research, driven by the need to efficiently manage energy while maintaining functionality. The integration of AI with IoT has given rise to the concept of AIoT, a field that has become increasingly relevant in complicated composition tasks. This integration brings both opportunities and challenges to low-power IoT systems. AIoT research branches into two primary directions: 1) **AI for IoT** This direction focuses on applying AI techniques to enhance IoT systems. AI algorithms are used to process and analyze the vast amounts of data generated by IoT devices, leading to more efficient operations, predictive maintenance, intelligent decision-making, and energy consumption optimization. 2) **IoT for AI** Conversely, this approach utilizes the infrastructure and data generated by IoT systems to improve and advance AI models. The real-world data collected by IoT devices provide a rich, diverse, and often real-time dataset for training and refining AI algorithms. This collaboration is crucial in developing more accurate and robust AI models that can adapt to various scenarios and environments.

In the realm of **AI for IoT**, my research focuses on harnessing the power of AI for better measurement of network conditions and optimization of network deployment for low-power IoT systems. LoRa (Long Range) is an emerging technique that enables long-distance communication and keeps low power consumption. It supports IoT applications in lots of large-scale environments where various types of land covers usually exist. However, due to the expensive cost of densely deploying end-nodes, the understanding of LoRa link channel is still coarse-grained. There are numerical empirical studies [1–7] conducted to measure LoRa coverage ranges. The specific range varies with experiment environments. Also, these works are not able to address the link dynamics, fine-grained networking coverage, and localization accuracy of LoRaWAN. Besides, the complexity of the environment makes it challenging to build an accurate LoRa link model for further anal-

ysis of the channel and related applications. There are link models that integrate environmental factors [1,8–11], but they did not fully utilize the fine-grained environment information. Also, they can not transfer well to new environments due to fixed environment modeling. In this dissertation, we design DeepLoRa that adopts deep learning for accurate path loss estimation of long-distance LoRa links using environment information. DeepLoRa achieves less than 4 dB estimation error which is 2× smaller than state-of-the-art models and is more transferable because of using less-information-lost raw environment data and highly generalizable RNN models. We also conduct a fine-grained link-level measurement in LoSee that shows spatial-temporal link dynamics, coverage, and link-information-based localization of LoRaWAN in an urban area which can benefit the deployment of LoRa gateways, service quality in mobile applications, and network management in practice.

Conversely, in the realm of **IoT for AI**, I design energy-efficient IoT systems that enable AI for real-world applications. Face touch is an unconscious and high-frequency behavior that most of us have [12–15]. Amidst the global pandemic of COVID-19, face-touch detection becomes imperative for reducing epidemiological risk [16–19]. Prior work has investigated a variety of emerging sensing techniques [20–29] to measure the distance between the hand and the face to detect face-touch events, or adopts on-body sensors to extract features from the movement of hands to classify face-touch gestures [30–34]. However, these approaches suffer from gestures similar to face touch or drain the battery really fast. Therefore, there is a significant need for an accurate and low-power face-touch monitoring system. In this dissertation, we propose FaceTouch, a low-power and versatile method that enables accurate face touch detection with a multimodal wearable system and AI model. FaceTouch achieves a 93.5% F-1 score of face touch detection with low power consumption.

## **1.1 Conducted Studies and Proposed Techniques**

### **1.1.1 DNN-based LoRa Link Model**

As LoRa enables long-distance communication in diverse IoT applications in lots of environments where various types of land covers usually exist, it is challenging to conduct thorough field

measurements on a large scale or precisely predict a LoRa link's path loss. A few models are integrating environmental factors to reflect the difference in the rates of path loss increasing [1, 8, 9], but those models only adopt regional environment information for prediction. State-of-the-art models [10, 11] adopt remote sensing techniques to analyze the composition of land covers along LoRa links quantitatively and select the corresponding empirical model to use for prediction, which does not fully utilize the fine-grained environment information. Also, they can not transfer well to new environments due to fixed environment modeling.

In Chapter 2, we present DeepLoRa. DeepLoRa adopts a deep learning-based approach to accurately estimate the path loss of long-distance LoRa links in complex environments. Specifically, DeepLoRa relies on remote sensing to automatically recognize land-cover types along a LoRa link. Then, DeepLoRa utilizes Bi-LSTM (Bidirectional Long Short Term Memory) to develop a land-cover aware path loss model.

We implement DeepLoRa and use the data gathered from a real LoRaWAN deployment on campus to evaluate its performance extensively in terms of estimation accuracy and model transferability. The results show that DeepLoRa reduces the estimation error to less than 4 dB, which is  $2\times$  smaller than state-of-the-art models.

### **1.1.2 Fine-grained LoRa link-level Measurement**

In Chapter 3, we further present LoSee which shows a fine-grained LoRa link-level measurement via a self-deployed LoRaWAN system consisting of 2 gateways and 6 mobile end nodes in a  $6\times 6\text{ km}^2$  urban area. By such measurement, LoSee studies three fundamental research issues and draws the following conclusions: 1) The spatial and temporal behavior of LoRa links is quite dynamic due to environmental factors; 2) The coverage of LoRa gateways is anisotropic; 3) The median error of RSSI-fingerprint-based localization in given setting is about 400 m. Without densely deployed LoRa gateways, the SOTA LoRa localization can support road-level localization.

## **1.2 Low-power And Accurate Face Touch Detection**

During the COVID-19 pandemic, protecting ourselves from virus infection has been of vital importance. There are many studies conducted in both IoT and AI to provide health management

applications, and face-touch is one of them. The mainstream automatic face-touch monitoring is currently performed by recognizing face-touching gestures using emerging sensing techniques and wireless signals, including acoustic [21–23, 35–38], radio frequency signals [24–26, 39], and magnetic signals [27, 28], to measure the distance between the hand and the face and recognize potential hand-to-face gestures. On-body sensors like inertial sensors have also been investigated to extract features from the movement of hands to classify face-touch gestures [30–34]. However, many similar gestures (e.g., picking up the phone, wearing a hat, or adjusting eyeglasses) can significantly degrade their system performance and generate lots of false alarms, causing unnecessary panic and/or bringing medical resources to a place where they are not needed. To filter out these false-positive gestures, a recent work leverages sensors in the ear to accurately detect touch events [29]. However, since it relies on always-on sensing and signal processing to guarantee high recall, the battery life is extremely limited (e.g., the system requires multiple charging times per day), increasing the user burdens and degrading the user experience. Therefore, there is a significant need for an accurate and low-power face-touch monitoring system.

To fill the gaps, in Chapter 4, we propose to leverage the wrist inertial sensor to detect the face-touch gesture that the hand moves towards the face area and utilize the channel response of chirp vibration signal propagating through the human body to detect events that touch the skin to compensate for the ambiguity of gesture classification. To achieve this goal in a computation-efficient manner, we develop a cascading classification model including three classifiers (one of which is a DNN) to filter out irrelevant gestures to significantly extend the battery life while keeping a high recall. Once a face-touch gesture is triggered, we activate the vibration sensor to detect touch events. We implement FaceTouch using commercial off-the-shelf hardware components and evaluate its performance with various user activities and false-positive behaviors. FaceTouch achieves a 93.5% F-1 score of face touch detection. The entire system only consumes  $60.89 \mu\text{W}$  power on average in normal daily usage and  $209.15 \mu\text{W}$  in extremely heavy usage, which is several magnitudes lower than the state-of-the-art systems, and FaceTouch can continuously detect face-touch events for 79 – 273 days using a small 400 mWh battery depending on usage.

### **1.3 Organization**

The remainder of this dissertation is as follows, in Chapter 2, we discuss the LoRa link model that exploits fine-grained environment info; in Chapter 3, we show LoRa link-level measurement results that address fundamental issues for deploying LoRa in real-world; in Chapter 4 we propose a low-power system that detects face-touch events utilizing vibration chirp signal integrated with IMU data; in Chapter 5 we conclude this dissertation.

## CHAPTER 2

### DEEPLORA: LEARNING ACCURATE PATH LOSS MODEL FOR LONG DISTANCE LINKS IN LPWAN

#### 2.1 Introduction

The development of Internet of Things (IoT) has witnessed broader applications, increasing of IoT devices and the expansion of network size. In many scenarios (e.g., agriculture, industry, logistics, city, home, healthy-care), a large amount of unattended IoT devices are deployed, sending small volume of data periodically or sporadically, which are expected to last working for years given limited energy budget. To simultaneously fulfill all these requirements, various short-range and low-power wireless techniques (e.g., BLE, 802.15.4) have been widely adopted in body-area and local-area IoT. To extend the scale of these networks, ad-hoc architecture (e.g., wireless sensor networks [40–42]) is further utilized in the past decades, but suffers from dramatically increasing deployment and maintenance cost with the increasing of network scale. To mitigate this gap, long-distance and low-power wireless techniques (e.g., LoRa, Sigfox, NB-IoT) have recently emerged to enable LPWANs (Low-Power Wide Area Networks). Due to low-cost COTS radio/gateway (e.g., Semtech) and open-source development (e.g., LoRa Alliance), LoRa is gaining popularity in both industry and academy areas [1, 43, 44]. One of the most popular LPWAN, resulted from LoRa, is called LoRaWAN.

LoRa operates at license-free frequency bands, thus LoRaWAN saves the cost subscribing to any telecommunication operator. LoRaWAN consists of end nodes equipped with LoRa radio and LoRa gateways. An end node directly connects with several LoRa gateways in its communication range. LoRa physical layer adopts chirp spread spectrum (CSS) modulation [45] to enable data packet reception under low signal-to-noise ratio (SNR) (e.g., -20dBm) while keeping low power consumption (e.g., 400mW transmitting at 20dBm, 5 $\mu$ W in idle mode) as its low duty cycle and narrow bandwidth. Moreover, COTS LoRa radio and gateway usually have a high signal sensitivity to receive the potential weak signals. Hence, LoRa obtains large link budget [1, 7, 46] which accounts for its high maximum feasible power loss along the signal propagation between an end node

and a gateway. The sufficient link budget is capable of providing reliable coverage spanning from several kilometers to tens of kilometers in various environments (e.g., urban area, rural area).

Though LoRa establishes long-distance link, we observe that the communication distance may vary greatly in real-world deployment. When an end node is deployed at different directions regarding to a gateway, the power attenuation of the link between them, dubbed as *path loss*, changes due to different types of land-covers (e.g., tree, buildings, road, river) along the path. An accurate path loss model is vitally important for LoRaWAN applications such as gateway deployment and end-client localization. Specifically, since path loss correlates to the packet delivery probability of a link [47], if we can accurately predict the path loss associated with a LoRa gateway before it is deployed, we can optimize the LoRaWAN coverage by selecting gateway locations. Moreover, in LoRaWAN, end node localization [11, 48–50] relies on the matching of the signal fingerprint (e.g., received signal strength indicator (RSSI)) observed by several gateways. If we can accurately predict path loss without exhausted site survey, the localization system will be deployed and maintained with low overhead. However, facing the environment diversity in different and large coverage areas of LoRaWAN, it is challenging to develop such an accurate and general path loss model with low overhead [10, 11].

Most existing methods of path loss estimation depend on various physical models (e.g., Friis [51], Bor [1], Okumura-Hata [8]), which depict the influence of environmental reflection, refraction and diffraction on wireless signal attenuation. Friis transmission formula can be used to calculate path loss when wireless signal is propagating in free space, but free-space transmission is hard to meet in various field studies [2–7, 47, 52]. Moreover, Petajajarvi et al. [2] and Bor et al. [1] explore the log-normal shadowing model [53] to estimate path loss. An environment related signal shadowing is modeled as log-distance attenuation and measured by field study, and the derived parameters are combined to adjust Friis transmission formula. Okumura [8] and Hata [9] study the path loss of cellular signal in urban, suburban, rural areas based on the data collected in Tokyo, then propose empirical formulas for different areas. Although these fine-tuned physical models consider the influence of surrounding environments, the accuracy of path loss estimation may vary significantly

since their environment classifications are too coarse-grained to model the diverse per-link environment characteristics such as the types and order of land-covers appearing along the path. How to break the ceiling of these physical models and accurately estimate path loss of long-distance wireless links is still a challenging problem.

In this paper, we propose DeepLoRa, a learning framework for accurate path loss estimation of long distance LoRa links. We have two key observations. First, some public remote sensing images [10, 11] can be utilized to recognize the fine-grained land-covers distributed along a link. Second, the influence of the land-covers on path loss is actually very complicated that not only the types of land-covers matter but also their appearing order along the link makes a difference (Section 2.3). Then, we resort to deep learning technique [54–56] to model the influence of a specific land-cover distribution on path loss.

Specifically, instead of considering the environment of a LoRa link as a whole, DeepLoRa divides it into an ordered sequence of short links (called *micro link*) with the same length. The detailed land-covers of each micro link are recognized by utilizing remote sensing images. Then we apply supervised Long-Short-Term-Memory (LSTM), which is one kind of Recurrent Neural Network(RNN) for sequence analysis, to learn a path loss model based on the measurements collected from the area of interest. Our LSTM model inherently capture the relationship between the types and order of land-covers and path loss. When we have trained a LSTM model using the data collected from a gateway, with only a few extra data collection and model training, the model can be directly transferred to accurately estimate path loss for other gateways in the areas with similar land-cover composition.

We implement DeepLoRa and extensively evaluate its performance on the dataset collected from a campus LoRaWAN deployment spanning  $6 \times 6 \text{ km}^2$  area in urban scenario. The dataset includes data recorded by two gateways  $G_1$  and  $G_2$  placed at the roofs of two buildings with different locations and heights. 6 mobile end nodes mounting on 5 bicycles and a car are used to collect GPS and wireless signal data. 16071 data records are collected for  $G_1$  and 15192 records are collected for  $G_2$ . The experimental results show that DeepLoRa achieves a mean error of 3.56dBm which is  $2 \times$

smaller than state-of-the-art models. When we transfer the model to the other gateway and fine-tune it with less than 200 data records, the model achieves a mean error of 4.79dB. Our contributions are summarized as follows:

- Instead of physical model, we first propose to utilize deep learning for path loss estimation of long-distance LoRa links across large area in outdoor scenarios.
- We empirically study the influence of detailed land-cover sequence on path loss in a real LoRaWAN system. We propose DeepLoRa utilizing adaptive LSTM model to learn the relationship between path loss and the corresponding types and order of land-covers.
- We implement DeepLoRa and evaluate its performance in real LoRaWAN deployment. The experimental results show that the mean error is  $2\times$  smaller than state-of-the-arts and the LSTM model can be generalized with low overhead.

The rest of the paper is organized as follows. Section 3.2 introduces the related work. We present the preliminary knowledge and illustrate our empirical study in Section 2.3. The system design of DeepLoRa is followed in Section 2.4. Section 2.5 and Section 4.6 exhibit the implementation and evaluation, respectively. We conclude our paper in Section 4.9.

## 2.2 Related Work

The characteristics of long-distance wireless links have been empirically studied and theoretically modeled in the past decades. We summarize the existing efforts from the following three aspects.

**LoRaWAN field studies:** In LoRaWAN, path loss estimation is facilitated by the study of LoRa coverage in real world. The LoRa radios and gateways are usually adopt the commodity products from Semtech. LoSee [47] deploys a LoRaWAN network consisting of one gateway and a mobile end node in campus environment of Tsinghua University and utilizes log-normal shadowing model to predict the path loss. It shows two gateways are needed to ensure the full coverage of the  $4.5km^2$  campus. Liando et al. [52] deploys 3 gateways and 50 end nodes. The maximum line-of-sight (LOS) and non-line-of-sight (NLOS) communication distance is approximate 9.08km and 2km when the packet delivery ratio is higher than 70%. Numerous other empirical studies [1–7] have

been conducted to measure the LoRa coverage ranges. The specific range varies with experiment environments. For example, Centenaro et al. [5] observe a range of 2km in an area of high-buildings. Bor et al. [1] obtain a range of 2.6km in rural areas and a range of 100m in an environment concentrated buildings. Wixted et al. [7] observe of 1km to 20km in the central business district. However, these empirical studies do not model and answer the question how path loss increases with communication distance at different rates in various environments. In DeepLoRa, we deploy 2 gateways and 6 mobile end nodes in campus environment to study the detailed relationship between the types and order of land-covers and signal attenuation.

***Land-cover and environment aware models:*** There are a few models integrating environmental factors to reflect the difference in the rates of path loss increasing. Empirical models Okumura [8] and Hata [9] can be applied to LoRa path loss estimation which are originally used in cellular scenarios. The Okumura-Hata model provides ready to use formulas that are suitable for different environments (e.g. urban, suburban, rural areas). Bor et al. [1] adopts the well known log-normal shadowing path loss model [53]. Different from free-space path loss, Bor model estimates the absolute path loss with the reference path loss plus relative path loss between two distances. And it introduces a parameter, called path loss exponent, that accounts for the rate of path loss increasing in diverse environments, but estimating this value requires extra on-site measurements. Demetri et al. [10] and Lin et al. [11] use remote sensing to quantitatively analyze the composition of land-cover types along a signal propagation path, then based on the types of land-cover, they select appropriate combinations of Okumura-Hata model and Bor model for further path loss estimation, respectively. Instead of adopting physical path loss model, DeepLoRa utilizes deep learning to develop a learning model which can depict the complex relationship between the path loss and the types and order of land-cover along the path. Thus, DeepLoRa can achieve more accurate path loss estimation.

***Machine learning based models:*** Some works [57–61] use machine learning to model the path loss regarding to the influence of surrounding environments. Oroza et al. [57] adopt random forest algorithm to predict the path loss for the wireless links in American River Hydrologic Observatory (ARHO) system. The model takes link-specific features as inputs and achieves average prediction

error of 3.74dB with standard error deviation of 3.40dB. Zhang et al. [58] propose path loss models for evaluating the unmanned aerial vehicle (UAV) communication channels based on machine learning models (e.g., random forest and KNN). They take propagation distance, Tx altitude, Rx altitude, path visibility (binary parameter indicating if there exists LOS path between the Tx and Rx UAVs), elevation angle as features. Cheng et al. [60] associate the floor plan of a building to RSSI values in each indoor Wi-Fi measurement. They trained Convolutional Neural Networks (CNNs) to capture the underlying path loss model. The model takes images (e.g., floor plan) as input and generates predictions of received RSSI, achieving Root Mean Square Error (RMSE) of 3.9404 dBm and good generalizability. The existing learning models, however, cannot guarantee the accuracy and generalizability at the same time, especially for long-distance links. In comparison, DeepLoRa utilizes Bi-LSTM to depict the sequential influence of different landcover along the link path and shows good generalizability.

## 2.3 Preliminary and Empirical Study

### 2.3.1 Physical Path Loss Model

When the transmitter and receiver antennas are put in ideal free space, Friis transmission formula [51] gives the free-space path loss (FSPL) as follow:

$$FSPL(d) = 10 \log_{10}(d) + 20 \log_{10}(f) - 27.55 \quad (2.1)$$

where  $d$  is the distance between the transmitter's antenna and the receiver's antenna and  $f$  is the frequency whose unit is MHz.

Based on Friis model shown in Equ. 2.1, some models such as Bor model and Okumura-Hata model require integrating environmental information so that can provide more accurate path loss estimation. Specifically, for Bor model, the use of environmental information is reflected in an introduced parameter:

$$PL(d) = PL(d_0) + 10 \cdot n \cdot \log_{10}\left(\frac{d}{d_0}\right) + X_{\sigma} \quad (2.2)$$

where  $PL(d)$  indicates the path loss when the distance between receiver's antenna and transmitter's antenna is  $d$ .  $PL(d_0)$  is the path loss at a known reference distance  $d_0$ .  $n$  indicates the path loss

exponent that is environment-specific and needs to be estimated by empirical data.  $X_\sigma$  is a zero-mean Gaussian random noise with standard deviation  $\sigma$ .

Okumura-Hata model requires selecting one of the formulas based on the surrounding environment of end node. The main formulas involved in our following discussion is given by:

- The formula used in urban environments is indicated as follows:

$$L_U(d) = 69.55 + 26.16 \log_{10} f - 13.82 \log_{10} h_B - C_H + (44.9 - 6.55 \log_{10} h_B) \log_{10} d \quad (2.3)$$

where  $h_B(m)$  is the height of LoRa gateway.  $h_M(m)$  is the height of LoRa end node.  $C_H$  is the antenna height correction factor which is defined as follows:

$$C_H = 0.8 + (1.1 \log_{10} f - 0.7) h_M - 1.56 \log_{10} f \quad (2.4)$$

- Similarly, the formula used in suburban environments is depicted as follows:

$$L_{SU}(d) = L_U(d) - 2(\log_{10} \frac{f}{28})^2 - 5.4 \quad (2.5)$$

To acquire the environmental information to support the aforementioned two models, one typical way is on-site measurements which are usually labor exhausted. Demetri et al. [10] and Lin et al. [11] adopt public remote sensing images for quantitatively analysing the composition of land-covers along propagation route remotely, then based on the recognized types of land-covers, they train and select appropriate physical model for further path loss estimation.

### 2.3.2 Remote Sensing based Land-cover Recognition

Literally, remote sensing is acquiring information of large-scale area on the earth, which can be the surface, the atmosphere, or the oceans, using air-crafts or satellites equipped with sensors that detect radiation reflected or emitted from target objects.

To recognize different types of land-covers from remote sensing images, as shown in Table 2.1, Demetri et al. [10] define the types of land-covers which are representative enough in characterizing the environment factors that affect LoRa signal attenuation. The land-covers are divided in two groups according to whether they may lead NLOS signal attenuation or not (i.e., LOS transmission).

Table 2.1 The types of land-covers.

NLOS	BUILDING GREENHOUSE TREES	buidings greenhouse structures trees
LOS	Field SOIL ROAD WATER	farming field or glassland bare soil streets,roads and highways lakes and rivers

A few features are extracted from the multi-spectral images [62, 63] which contain the different radiation reflected by the land-covers. Then, each  $10 \times 10m^2$  area in geographic space can be classified into one type of land-cover by applying a classifier trained with Support Vector Machines (SVM) [64–66]. With the types of land-covers along a LoRa link, they decide which Okumura-Hata formula should be used based on the dominating land-cover type. They choose to use suburban formula if dominating type belongs to LoS category and use urban formula otherwise.

Similarly, Lin et al. [11] classify different types of land-covers by using Random Forest. They can achieve an area resolution of  $0.6 \times 0.6m^2$  on their map due to the fine-grained remote sensing dataset. After extracting a sequence of land-covers along a link, they separate the link into segments by the boundaries between adjacent different land-covers and adopt Bor model to estimate the overall path loss segment by segment. The path loss exponent for each type of land-cover is trained by site-surveyed data. To evaluate the influence of surrounding environment on signal attenuation, we deploy a LoRaWAN system in a campus environment, where is full of different types of land-covers. We also adopt remote sensing techniques for land-cover recognition.

### 2.3.3 Campus LoRaWAN System and Dataset

Figure 2.1 shows the overview and hardware of our campus LoRaWAN system. The system is built on the LoRaWAN protocol. In our system, we deployed 2 gateways  $G_1$  and  $G_2$ . Each of them is equipped with a MCU, a SX1276 transceiver and a Raspberry Pi 3 for programming remotely. They are located at the roofs of two different buildings in a campus environment as shown in Figure 2.1(a). Their altitudes are 84m and 68m, respectively. The ground altitude of the campus area is about 52m. Our LoRa end nodes are implemented with a MCU, a SX1278 transceiver and a

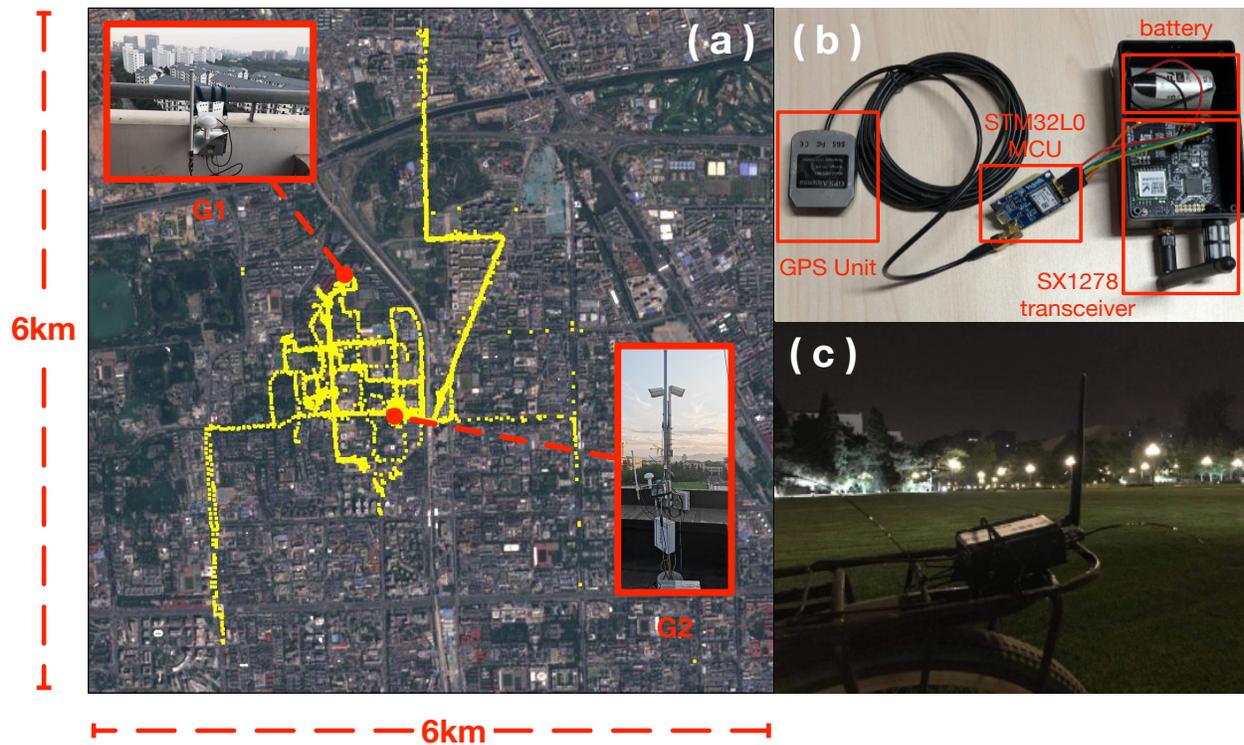


Figure 2.1 The overview of the deployment and dataset of our campus LoRaWAN system.

GPS unit as shown in Figure 2.1(b). They are mounted on 5 bicycles and a car as shown in Figure 2.1(c). While the bicycles and the car are moving, the LoRa end nodes will send packets to the gateways. All the packets are transmitted with spreading factor  $SF = 12$ , bandwidth  $BW = 125\text{kHz}$ , and coding rate  $CR = 4/5$ . TX power together with antenna gains is about 19dB. The 6 end nodes use channels of 486.3kHz, 486.5kHz, 486.7kHz, 486.9kHz, 487.1kHz and 487.3kHz, respectively. The interval between two adjacent packets is 5s. A packet includes the GPS coordinates, timestamps and sequence number. The corresponding SNR and RSSI are logged at gateway.

We completed deploying the system in Dec, 2018. All the data were collected in the campus or surrounding area from Dec 22, 2018 to Mar 15, 2019. We logged over 30,000 records at the two gateways in total. Via GPS readings, we can calculate the link distance  $d$  and the height difference  $h$  between end node's antenna and gateway's antenna. As shown in Figure 2.1(a), the measurement locations are along the main roads in or around the campus. The whole region of interest is a  $6\text{km} \times 6\text{km}$  square area in where the land-covers include buildings, roads, parking lots, lakes, a

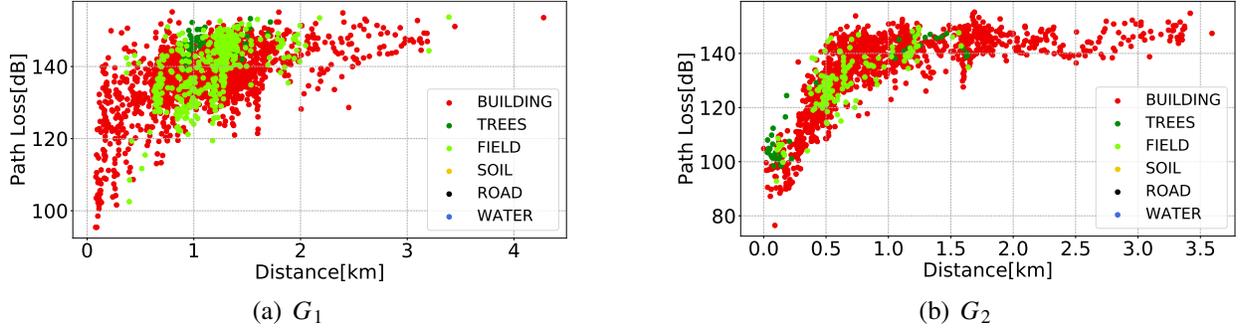


Figure 2.2 Path loss vs. distance for different land-cover dominated links regarding to different gateways  $G_1$  and  $G_2$ .

river, glassland, trees and playground. The red points are the locations of our two gateways  $G_1$  and  $G_2$ . The yellow points are the locations of all packet transmitted by the moving end nodes.

To study the environmental effects, we adopt the way that Demetri et al. [10] proposed to get a sequence of land-covers for each LoRa link. The detailed implementation refers to Section 2.5. We clean the data and remove redundancy in the way section 2.5.2. Finally, we obtain a dataset that consists of over 4,000 unique records regarding to two gateways. 2,301 records are from  $G_1$  and 1,780 records are from  $G_2$ , respectively.

In practice, given a received LoRa packet, we can obtain its RSSI and SNR from gateway, but the RSSI is the power combination of LoRa signal and various noises. To eliminate the influence of the noises, we use Expected Signal Power (ESP) as a metric to indicate actual received signal power, which can be derived from the following equation:

$$ESP = RSSI + SNR - 10 \cdot \log_{10}(1 + 10^{0.1SNR}) \quad (2.6)$$

Then, we use the following equation to obtain the ground truth of signal path loss.

$$PL = P^t + G^r + G^t - ESP \quad (2.7)$$

where  $P^t$  is the power fed into the transmitter's antenna,  $G^r$  and  $G^t$  are the power gains at receiver and transmitter sides, respectively. In our system, the sum of  $P^t$ ,  $G^r$  and  $G^t$  is 19dB.

### 2.3.4 Empirical Study

We already know that free-space path loss does not take attenuation caused by environment into account, thus underestimating the true path loss. How to model the impact of environment on path

loss is the focus of designing path loss models. We conduct empirical study based on our LoRaWan system measurements to find laws in which the environment information affect the path loss of LoRa links. As shown in Figure 2.2, we color the measured points according to their dominating land-covers along the link. We can see different links have different dominating land-covers (e.g., buildings, trees, roads and field). Moreover, even under the same distance, the distribution of path loss varies for different types of dominating land-covers. The buildings and fields make the path loss more dynamic than other types of land-covers. Therefore, the results show that different types of land-covers will lead diverse effects on the path loss.

The main problem of existing environment aware path loss models is that the land-cover information they use is either extracted from the surrounding environment of end node or from statistics of the whole link, which does not make full use of the fine-grained land-cover information. In Figure 2.2, we notice that even for the links with the same type of dominating land-cover, their path loss variance is still very significant, especially for  $G_1$ .

To discuss the problem in detail, we select four links  $R_1$ ,  $R_2$ ,  $R_3$  and  $R_4$  from our dataset as shown in Figure 2.3(a). The properties of those links can be found in Table 2.2.  $R_1$  and  $R_2$ ,  $R_3$  and  $R_4$  have something in common: 1) their length are nearly the same; 2) the type of dominating land-cover of both links are BUILDING; 3) the percentages of NLoS land-cover of both links are very close. If we adopt aforementioned empirical models, we should get very close path loss estimation of each pair of links. However, as it shown in our real measurements, the differences between path loss of the two pairs are more than 20dBm, which cannot be ignored.

Table 2.2 The properties of two link pairs.

Link Index	Length[m]	Dominating Land-cover	NLoS Land-cover Percentage	Path Loss[dB]
$R_1$	46.04	BUILDING	0.61	140.61
$R_2$	47.27	BUILDING	0.62	114.74
$R_3$	76.85	BUILDING	0.52	149.34
$R_4$	75.45	BUILDING	0.52	127.40

We plot the detailed types of NLoS and LoS land-covers of these links at different distance along the path from their end nodes to the gateway in Figure 2.3(b) and Figure 2.3(c). We can see that

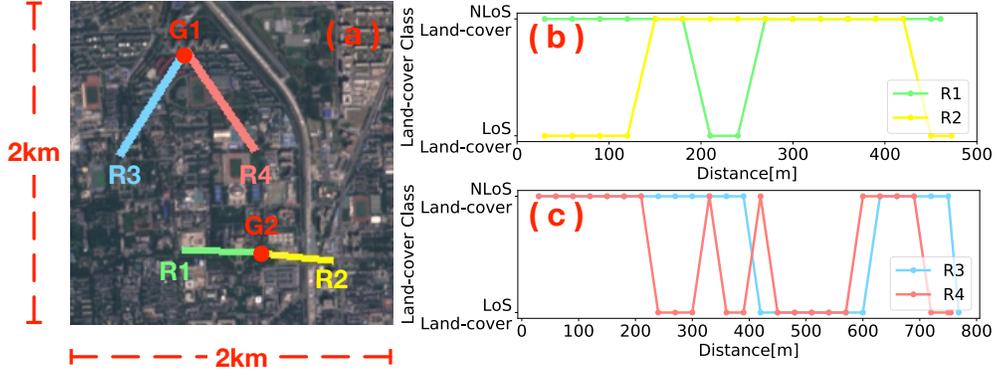


Figure 2.3 Four example links in our area of interest.

each pair of links show significant difference in the order that NLoS and LoS land-covers present along their paths. For those links (e.g.,  $R_2$  and  $R_4$ ) with less path loss, they have less NLoS land-covers that present near the end node. As other properties of the links remain similar, we believe that not only the types of land-covers affects the path loss, but their order along a link also matters. The reason is that for the distance between a obstacle and the end node, the closer the distance is, the more probability the signal can be blocked. Although buildings may block the signal between the end node and the gateway, the limited height of building has less probability to block the signal if it is far away from the end node.

Though Lin et al. [11] divides the whole link into segments by different land-covers and calculate the overall path loss segment by segment, they only use the type of land-cover in each segment to determine the path loss exponent for Bor model. The result would be of no difference if the order of those segments change in the link. There is still much room for us to improve in LoRa path loss estimation by fully exploit the fine-grained order in which land-covers present in the link.

### 2.3.5 Recurrent Networks

To model the observation mentioned in last section, we look into the signal-propagation link from finer granularity other than a whole. Link path loss can be regarded as a result of traversing a sequence of micro-environments. Based on this understanding, we need sequence analyze techniques to build the model. Finally we resort to Recurrent Neural Network (RNN), a main Deep Neural Network (DNN) approach for tackling with sequence data. Though effective on some

tasks [67,68], the simple architecture suffers from vanishing and exploding gradients problem [69], making the gradients hard to back-propagate through long sequence of hidden units. Thus, it is difficult for the traditional RNN to learn long-time dynamics. Since LoRa link can be very long, traditional RNN may fail on long distance cases. To address the vanishing and exploding gradients problem, LSTM [69] was proposed with a memory and forgetting mechanism. Besides, Bidirectional Recurrent Neural Network (BRNN) [70] can encode the temporal information in both the sequence order and the reverse order, which better captures the properties of the sequence. Considering all those factors, we choose to adopt Bidirectional LSTM(Bi-LSTM) in our model to capture the fine grained and order-dependent environmental information of LoRa links.

## 2.4 System Design

### 2.4.1 Overview

Figure 2.4 shows the overall workflow of our system. DeepLoRa consists of three parts. To start with, given a location where we intend to deploy a LoRa gateway at, we get free multi-spectral images of related area of interest via Sentinel-2 open access API. Next, we generate a land-cover map from multi-spectral images through **land-cover classification**. Each pixel in the land-cover map is the class label that represents the true land-cover type in the real map. For regional estimation, we assume that an end node can be deployed at any possible point in the area, so we can get a list of coordinate pairs of the gateway and an end node, which is exactly a list of LoRa links. Then, **Link segmentation and embedding** produces formalized descriptions of the land-covers traversed by LoRa link based on land-cover map, which are in the format of sequences. Moreover, our **path loss model** based on DNN takes the sequences together with experimental specific parameters as inputs, and predicts corresponding path loss such that the ESP received by the gateway can be calculated. Finally the regional estimation of ESP received by the gateway can be visualized as a heatmap.

### 2.4.2 Land-cover Classification

Land-cover classification is the first step of DeepLoRa, it provides us fine-grained knowledge of the land-cover information traversed by LoRa link. To formalize this problem, given an area of

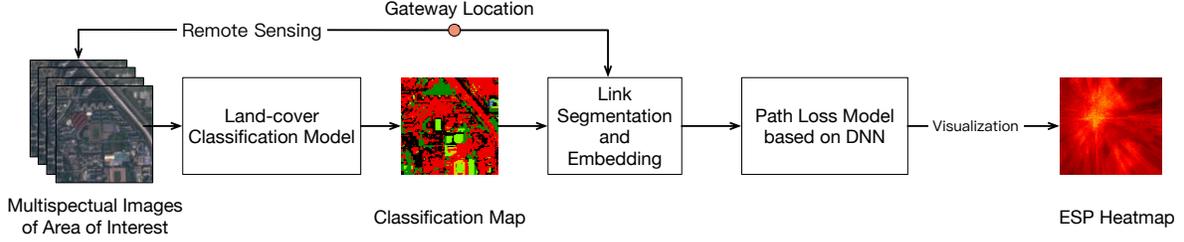


Figure 2.4 Overview of our system design.

interest that is  $10M \times 10Nm^2$  large and its multi-spectral images (10m is the pixel resolution of the multi-spectral images,  $M$  and  $N$  are the width and height of the images, respectively), we want to obtain a land-cover map of which each pixel  $p_{ij}$ ,  $0 \leq i < M$ ,  $0 \leq j < N$  indicates the corresponding land-cover type  $c_k$ ,  $0 \leq k \leq 5$  that the corresponding area  $a_{ij}$  of  $10 * 10m^2$  in the real map belongs to. We consider 6 land-cover types in total as shown in Table 2.1 except GREENHOUSE since this class does not present in our experiment area. Actually it is a per-pixel classification problem.

For each unit area  $a_{ij}$ , we extract a feature vector  $\mathbf{f}$  including the raw spectral values of corresponding pixel, the Normalized Difference Vegetation Index (NDVI) and the Normalized Difference Water Index (NDWI) from corresponding multi-spectral images. Then we feed the feature vector to SVMs with Radial Basis Function (RBF) kernel that predicts whether  $a_{ij}$  belongs to land-cover type  $c_k$  or not. We train 6 binary classifiers. Each classifier is trained for one specific land-cover type, and we select the one with the highest confidence score as final prediction. Altogether we can get the land-cover classification map.

### 2.4.3 Path Loss Estimation

Once we are done with land-cover classification, we can exploit the detailed environment information to design our DNN based path loss model. We need to first select a region that can represent the LoRa link. Then according to discussions in Section 2.3.4 and Section 2.3.5, we need to mount the types of land-covers in the link-region as a sequence of micro-environments and further formalize it as the inputs of the DNN learning framework with Bi-LSTM units.

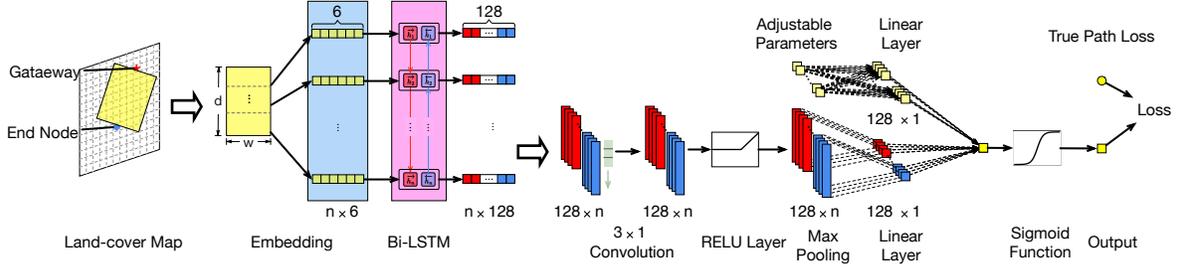


Figure 2.5 Deep neural network based on Bi-LSTM for path loss prediction.

### 2.4.3.1 Link Segmentation and Embedding

To represent the land-cover composition of a LoRa link, we do not just take a "line" but a rectangular area connecting the end node and the gateway from land-cover classification map as shown in Figure 2.6. Since in our scenario, usually the direct link is NLoS path and the attenuation caused by the environment can be quite complex due to reflection, diffraction, diffusion and so on. Moreover, factors like the orientations of transmitter's antenna and receiver's antenna can affect the actual propagation route so that make the "line" is hard to be determined. Besides, the misclassification of a few pixels on the line would affect the whole sequence if just take one line of pixels into account. Selecting rectangular area can provide fault tolerance to above concerns. The width of this rectangle needs to be selected according to experiment and empirical knowledge. Then, we segment the rectangular area and embed it into sequence format. Take a closer look at the embedding process in Figure 2.5 as Figure 2.6. We divide the extracted link region of length  $d$  and width  $w$  into micro-link regions of length  $d'$  from the end node to the gateway. We can get  $n = \lceil d/d' \rceil$  micro rectangles in total. If the remainder  $r \neq 0$  we still regard the rest part as a micro-link. The granularity and length of the sequence is actually determined by  $d'$  and have direct impact on estimation accuracy. We will discuss the selection of  $d'$  in section 2.6.2. Say that micro-link region  $l_i, 0 \leq i < n$  contains  $m_k, 1 \leq k \leq 5$  pixels for each land-cover type  $c_k$ . Then each micro-link region is embedded into a  $1 \times 6$  vector  $v_i$  by counting the proportion of 6 land-cover types.

$$v_i = [v_i^0, v_i^1, v_i^2, v_i^3, v_i^4, v_i^5]$$

$$v_i^k = m_k / \sum_{j=0}^5 m_j \quad (2.8)$$

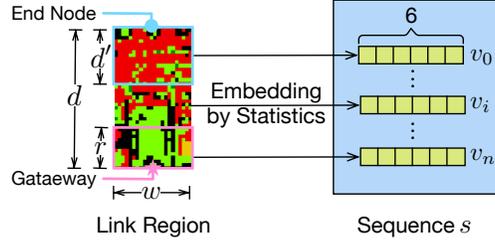


Figure 2.6 Link segment and embedding.

Such that we transform the rectangle area into an ordered sequence  $s = [v_1, v_2, \dots, v_n]$  consists of land-cover vectors. After embedding, we input the sequence to our designed deep neural network.

#### 2.4.4 DNN based Path Loss Model

The architecture of our neural network based path loss model is shown in Figure 2.5. The sequence of vectors is first input to Bi-LSTM unit to extract order dependencies. As RNNs can unfold along time axis (in our case, distance), they enable information flow traverse from the start of the sequence to the end of the sequence thus capture the forward dependency and connect the output of current frame (timestamp, location, etc.) to previous frame. This capability suits our demand that we want to estimate the path loss at gateway which is the last frame in our sequence considering the attenuation from the start of the sequence. One concern is that RNNs are not good at learning long-term dependencies. But LoRa links can span extremely long distances, resulting in quite long sequences. Even if we adopt quite coarse link segmentation granularity, e.g.  $d' = 3$  (i.e., corresponding to a distance of 30m in reality), we'll get a sequence of length 100 for a link of distance 3km. That sequence is equivalent to some paragraphs in machine translation task, which we already know that is hard to process with more complicated neural architectures. Not to say longer link with finer segmentation granularity. LSTM is more capable of learning long-term dependencies like sequences contains hundreds of elements. To better relate the information from the start of sequence to the end as we have discussed that land-covers near the end node are more likely to block the signal, we further adopt Bi-LSTM.

All RNNs have a chain form information flow from the former frame to the next frame as the forward layer related flow (e.g., blue arrows) in Figure 2.7. Bi-LSTM contains information flow

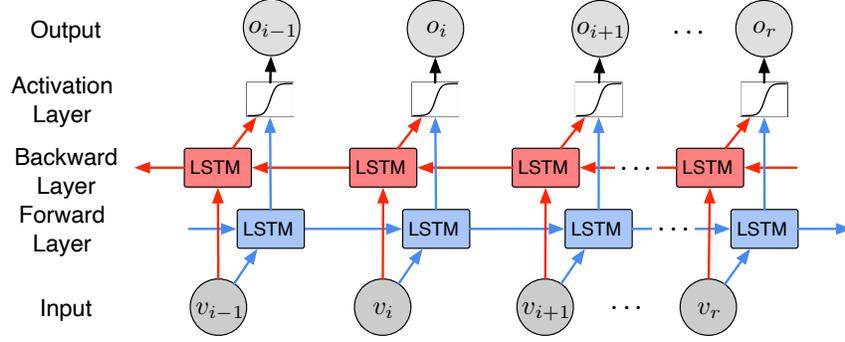


Figure 2.7 The information flow of Bi-LSTM.

from the other direction as the backward layer related flow (e.g., red arrows) in Figure 2.7. This ensures that the land-cover information from both the start and end of the sequence can be captured instead of 'forgot' when length of the sequence is very long.

The outputs of Bi-LSTM are input into convolution layers to extract local features and context dependency. Rectified Linear Unit (ReLU) layer [71] introduces non-linearity to the model. After max pooling, the output features are down-sampled and the dimensionality is reduced. Then we linearly map the features to the path loss. To be noted that, while doing linear mapping, we can add extra parameters to the network which can be any factor that has an effect on the path loss (also we can introduce non-linearity with other units). In this way our network becomes extendable when we have other properties of LoRa Link, e.g. weather condition, temperature, etc. So that we can quantitatively study new influencing factors in the future. For now, we just input the link distance as well as the height difference of transmitting antenna and receiving antenna.

The actual path loss of a LoRa Link has boundaries, it can not be less than 0, it can not exceed the maximum link budget constraint to the maximum transmitting power and end node sensitivity. So we curve our final estimation to a value between 0 and 1 using sigmoid function. In this way we can control the range of losses for the sake of training convenience. We just need to scale the estimation with the upper boundary to get the predicted path loss. Path loss larger than the upper boundary then will be curved to the upper boundary which indicates failure of packet delivery. In our experiments, we take 160dB as the upper boundary.

An important concern of DeepLoRa is whether it can be used in new environments. Our system

design enables 3 levels of generality and can be transferred to new environment painlessly.

- We do not manually select features but use sequence reorganized from real land-cover map with other factors as inputs, so that our model can learn a mapping which approximates to the law of signal propagation. This ensures the model generality of the first level.

- When we train our model, we endeavour to select training data that covers the links of various distances and land-cover compositions as introduced in section 2.5.2, then our training set span much room in the full feature space. This ensures the model generality of the second level.

- We adopt a Bi-LSTM based DNN model in our path loss model. Neural networks that are trained on a large history dataset can be fine-tuned with a small dataset contains new data to adjust its weights to fit new observations. So when we fine-tune our model with just a few data from the new environment, it can achieve higher accuracy than original model in the new environment. This is one advantage over many other machine learning based models since they need to retrain their model with fixed data from scratch and not promise to get better results. This ensures the model generality of the third level.

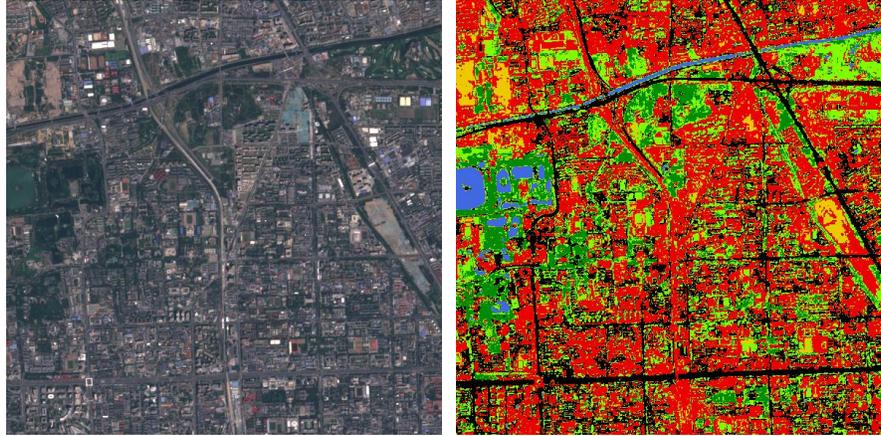
The first two levels of generality enable fair transferability of the original model while the last level of generality provide a feasible way to enable model fine-tune when we have higher demand for accuracy and would like to do lite on-site measurements for it. We evaluated the generality and tranferability of our model in Section 2.6.3.

## **2.5 Experiment Details**

### **2.5.1 Land-cover Classification**

We obtain multi-spectral images (hereafter *tile*) of resolution 10m from sentinel-2 open hub. We build a dataset from those images, which consists of small portions of micro-areas on the map to train and evaluate our land-cover classifiers. We manually labeled these areas using the online tool semantic-segmentaion-editor [72] which support annotating pixels in RGB images by checking areas. We obtained 900 labeled pixels in total, 180 pixels per class. The dataset then is divided into training set and testing set by 9 : 1. Training set is used for training and model selection, testing set is used for evaluating model performance.

■ Building   
 ■ Tree   
 ■ Field   
 ■ Soil   
 ■ Water   
 ■ Road



((a)) RGB Map

((b)) Land-cover Map

Figure 2.8 RGB map and land-cover map of the area of interest.

The testing result shows a quite high overall classification accuracy of 97.4% for all land-cover types which indicates that we can regard the obtained land-cover map as a reflection of true environment condition. Figure 2.8(b) shows the generated land-cover map of our area of interest. Different colors indicate different types of land-covers. We can see most parts of the area are full of buildings. Trees, fields and roads occupy a large part of the rest area. Water and soil only appear in a few parts of the area.

### 2.5.2 Path Loss Estimation

Our path loss model is based on DNN. We implemented it using Pytorch [73]. We need to use our collected data for model training. While training the model, we have to make sure that two same inputs can only be mapped to the same output, otherwise our model will be confused. So we clean our data before training. Since our data is continuously collected as bike moves, the locations logged by GPS unit are continuous on the map. Due to the 10m resolution of multi-spectral images we use, we regard every area of size  $10 \times 10m^2$  in reality as a pixel on the map. When we transform the GPS coordinates into coordinates on the map, many locations in reality are mapped to the same pixel with different ground truth path loss. To remove redundancy and get a unique ground truth for each input to the path loss model, we calculate a mean path loss for those measurements with

locations fall into the same pixel.

To train and evaluate our path loss model, we split the dataset into training set and testing set by 9 : 1. Since the principle behind our model is sequence processing and the length of sequence has significant impact on path loss, we separate our data into bins based on their sequence lengths before we split the dataset. In this way, we make sure that our training set contains sequences of diverse lengths, and their percentages in training set are close to these in testing set. Such a balanced training set promises a more general model. However, there is still gap of the link composition between training sets of two gateways and the two gateways are located at different altitudes, so it remains a challenge to apply the model trained for one gateway to a new gateway or new environment directly. We also extract data from training set for two gateways in several proportions, and conduct experiments on model transferability between different environment or gateways with those data. We will discuss more about it in Section 2.6.3.

For the link segmentation and embedding, we tried with multiple values of rectangle width  $w$  and micro-link length  $d'$ , the comparison and evaluation can be found in section 2.6.2. We finally select  $d = 3$ ,  $w = 7$ (represent 30m, 70m respectively) in our following experiments. We train our model with learning rate  $lr = 0.0001$ , batch size  $train_{bs} = 16$ , and test the model every 5 epochs.

## 2.6 Evaluation

### 2.6.1 Overall Performance

We evaluate our model accuracy in comparison with free-space model, Bor model, and two models PATH and INTERSECTION proposed by Demetri et al. [10] on the same testing set by calculating the absolute difference between path loss estimation and ground truth value. The path loss exponent of Bor model is fitted by the same training set as our model. PATH and INTERSECTION use settings in the original paper [10]. The result is as follows: Among all these models, DeepLoRa achieves the lowest error of less than 4dB for both gateways with the best performance of as low error as 3.29dB, which outperform those models in comparison by at least 50%. Also the standard deviation of DeepLoRa is limited to be 3.xdBm, the stability of estimation is ensured. We plot the raw estimation errors of these models except free-space model(since its error is too large) on the

Table 2.3 Absolute average estimation error on path loss. Average (avg[dBm]) and standard deviation (std[dBm]) among different models.

$G_{id}$	$G_1$		$G_2$		all	
	avg	std	avg	std	avg	std
DeepLoRa	3.29	3.12	3.94	3.21	3.56	3.17
INTERSECTION	19.91	7.13	17.55	10.00	18.88	8.58
PATH	20.93	7.68	17.35	10.35	19.36	9.12
Bor	8.70	7.18	12.24	8.83	10.25	8.14
free-space	52.23	6.04	47.52	9.64	50.17	8.16

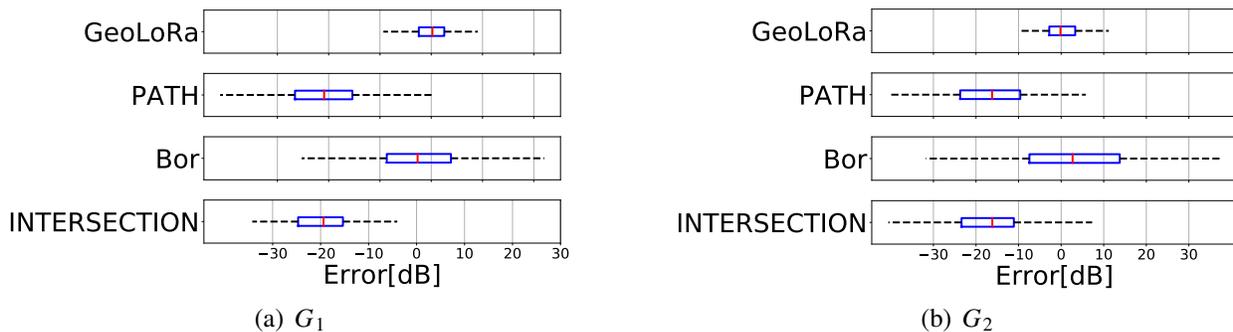


Figure 2.9 The distribution of the estimation errors on the full testing set.

full testing set in a box plot as Figure 2.9. We can see that the raw estimation errors of DeepLoRa is centered around 0dB which means that it has no tendency of underestimating or overestimating path loss, while estimation of other models all show obvious offset towards one side of 0. The error distribution of DeepLoRa is way narrower than other models, the magnitude of the largest error is less than 10dB and the magnitude of 50% errors are less than 5dB. It further proves that DeepLoRa achieves higher estimation accuracy with low variance.

## 2.6.2 Link Segmentation

To study the impact of different link segmentation parameters on the final performance of the path loss model, we train models for several combinations of link region width  $w$  and micro-link length  $d'$ . The performance of those models are compared based on absolute average estimation error. The results is reported in Table 2.4 Though the variance of estimation error among all the settings is not significant, we can still find some changing rule from it. When we fix the width  $w$ , and vary length  $d'$ ,  $d' = 3$  gives lower error than  $d' = 5, 7$ , this is consistent with our intuition

Table 2.4 Absolute average estimation error with different link segmentation parameters.

$d'(10m)$	$w(10m)$	avg(dB)
1	7	3.79
3	7	3.29
5	7	3.75
7	7	3.54
3	3	3.85
3	5	3.28
3	11	3.30

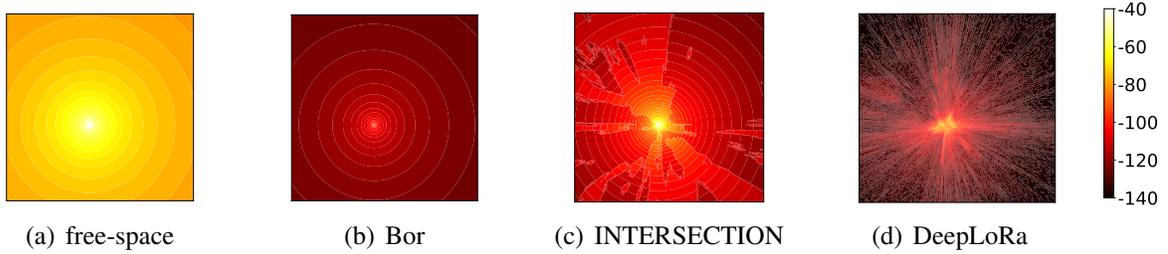


Figure 2.10 ESP heatmaps of a  $6 \times 6 km^2$  area with regard to gateway  $G_2$ .

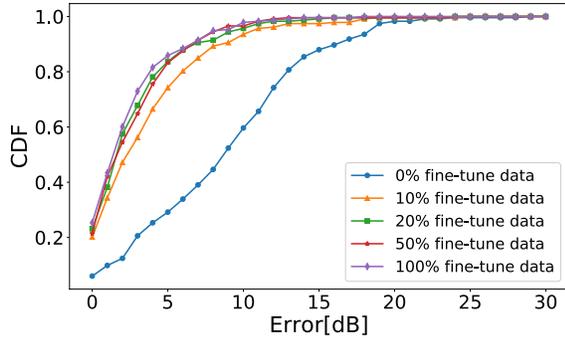
that the finer the granularity of the sequence, the better the result. But  $d' = 1$  does not produce the best result. The reason mainly lies in two folds: 1) for the same link, the length of sequence with  $d' = 1$  is 3 times longer than that of sequence with  $d' = 3$ . Long sequences are harder to learn; 2) smaller  $d'$  means smaller micro-link region, there are so few pixels in such region for embedding that it provides little fault tolerance for land-cover classification. So we choose  $d' = 3$ . When we fix the length  $d'$ , and vary width  $w$ ,  $w = 3$  seems to give the highest error while the model achieves similar results with  $w \geq 5$ . Actually setting  $w = 3$  is similar to selecting a "line" to represent the link, it is just too narrow as we already explained in Section 2.4.3.1. But it is not always the wider the better since wider link region means more calculation during embedding. To balance between fault tolerance and computation, we select  $w = 7$ .

### 2.6.3 Model Generality and Transferability

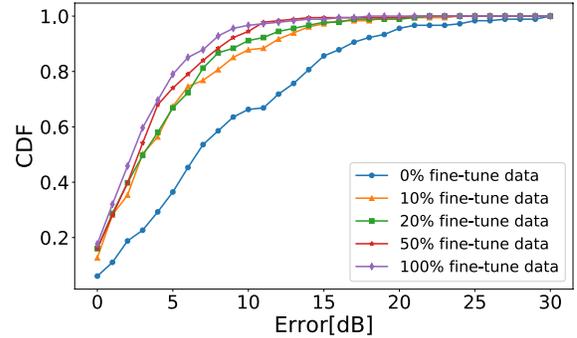
To compare the performance between DeepLoRa and other models when applied in new environment, we conduct testing of model trained with training set of one gateway on the testing set of the other. Since free-space model, PATH and INTERSECTION do not apply in such scenario (their performance should be the same as in Table 2.3), we just compare our model with Bor model

Table 2.5 Absolute average estimation error during model transfer.

$G_{id}$	$G_1 \rightarrow G_2$		$G_2 \rightarrow G_1$	
	avg (dBm)	stdd (dBm)	avg (dBm)	stdd (dBm)
DeepLoRa	9.58	0.76	8.92	6.51
Bor	10.20	9.16	10.00	8.91



(a)  $G_1$



(b)  $G_2$

Figure 2.11 CDF of absolute estimation error when apply model in new environment with different amount of fine-tune data.

as in Table 2.5.

We can see that when DeepLoRa model trained on  $G_1$  training set is applied for  $G_2$  testing set, the average estimation error and standard deviation of DeepLoRa is 9.58dBm and 0.76dBm which is lower than those of Bor model. The same things happen if we reverse  $G_1, G_2$ . This result indicates that DeepLoRa guarantees good generality. When we transfer DeepLoRa to new environment it still retains satisfactory estimation accuracy.

Above result just shows DeepLoRa generality of the first two levels, we also conduct experiments to verify its generality of the third level. Before apply the model directly in new environment, we use different percentages of training data belonging to the new gateway to fine-tune the base model. Adding 0% of data means using the base model directly without fine-tuning. The result is given by testing the fine-tuned model on the testing set of new gateway as shown in Figure 2.11. We can see from the CDF that when test on  $G_1$  data, using 10%  $G_1$  training data for fine-tuning can control 80% of estimation errors within 7dBm, when test on  $G_2$  data, using 10%  $G_2$  training

data for fine-tune can control 80% of estimation errors within 8.5dBm, which approximates the performance when using 100% training data of the new gateway for fine-tuning (i.e., equivalent to train the model from scratch for the new gateway).

In Table 2.6, we report the absolute average estimation error of above experiment. We can see that using 10% of training data to fine-tune can improve the estimation accuracy up to  $2\times$  when compared with no fine-tuning. And we can see greater improvement when fine-tune  $G_2$  model and test on  $G_1$  data. This is because dataset collected for  $G_2$  is more diverse than that of  $G_1$ , resulting in a more general base model. The extra accuracy benefit brought by increasing the amount of fine-tuning data can be ignored when we already use 20% or more fine-tune data. In our context, 10% training data is around 200 records, which can be easily collected with our LoRaWan system. Actually we may not even need 10%, 5% or less would be enough. Based on this result, we suggest first training a base model with large-scale history data which can be obtained from existing real-world deployments, and fine-tuning the base model with a few data collected in the new environment for seek of higher accuracy demand.

#### 2.6.4 Generating ESP Heatmap

In order to show the performance of DeepLoRa more intuitively, we do per-link path loss estimation using DeepLoRa for each unit area in the  $6 \times 6 km^2$  area shown in Figure 2.1. Finally we draw the ESP heatmap of this area with regarding to gateway  $G_2$  ( $G_1$ 's is similar to  $G_2$ 's). We also draw heatmaps using free-space model, Bor model and INTERSECTION model for comparison purpose. Figure 2.10 shows the heatmaps. In these heatmaps we use the same color scale of  $[-40, -140]$ dBm for all models, darker color means lower ESP value (i.e., larger path loss). ESP value equal or lower than 140dBm means unable to deliver the packet/no coverage. It is clearly that free-space model and Bor model only provide isotropic path loss estimation with lower accuracy. INTERSECTION model reflects the anisotropy to some extent, but the granularity is not fine enough. When it comes to DeepLoRa, we can see the difference between each single link clearly, many holes of coverage that are hidden in former heatmaps now show up.

Combine the quantitative experiments results and the visualization of estimation for large-scale,

Table 2.6 Absolute average estimation error with model fine-tune.

$G_{id}$	$G_1 \rightarrow G_2$ (dBm)	$G_2 \rightarrow G_1$ (dBm)
0% fine-tune data	9.58	8.92
10% fine-tune data	4.06	5.37
20% fine-tune data	3.42	4.80
50% fine-tune data	3.40	4.21
100% fine-tune data	3.15	3.99

we can prove that DeepLoRa is a more accurate and robust path loss model which is capable of providing ESP/coverage estimation for the area of interest in fine granularity.

## 2.7 Conclusion

To conclude, we propose DeepLoRa, a learning framework enabling accurate and general path loss estimation for long distance wireless links in LPWAN. By deploying a real LoRaWAN system in campus environment, we empirically study the relationship between the path loss of a link and the land-covers along the link. We have observed that not only the types of land-covers lead different signal attenuation, but also the order of these land-covers has significant influence. Given the position of an end node, we utilize remote sensing images to recognize the types of land-covers between the end node and a gateway. Then, we use Bi-LSTM to develop a learning path loss model which captures the influence of both type and order of these land-covers on the path loss. We implement our learning model and evaluate it based on our dataset. In comparison with state-of-the-art physical models, the experimental results show that DeepLoRa achieves more accurate and fine-grained path loss estimation and needs few transferring training overhead.

## CHAPTER 3

### IS LORAWAN REALLY WIDE? FINE-GRAINED LORA LINK-LEVEL MEASUREMENT IN AN URBAN ENVIRONMENT

#### 3.1 Introduction

Low-power Wide Area Network (LPWAN) is an emerging IoT paradigm aiming for low-power wireless communication over kilometer links. Several LPWANs (e.g., Long Range (LoRa) [74], Narrow-Band(NB)-IoT [75], SIGFOX [5]) with different physical layer designs have been commercialized, enabling city-scale IoT applications at a low cost. For example, NB-IoT [75] and LTE-M operate on the LTE band as a part of 5G for the massive IoT. SIGFOX [5] uses an unlicensed band but is a proprietary network. In contrast, LoRaWAN [74] operates at an unlicensed spectrum and follows an open-source standard, attracting much attention from academia and industrial communities.

LoRa networking stack adopts the Chirp Spread Spectrum (CSS) modulation at the physical layer (LoRa-PHY). By suppressing the background noise on the spectrum in CSS, LoRa-PHY can successfully demodulate a symbol even if its SNR level is as low as -20 dB [76, 77], making it a representative of the low-power and long-distance communication. With such LoRa links, spatial-temporal link dynamics, coverage, and link-information based localization are three fundamental research issues [78] which can be formulated as follows:

- For spatial-temporal link dynamics, the critical questions are whether different links with the same distance show similar link performance and whether a link's performance is stable over a long period.
- For coverage, the critical question is whether the conceptual "long-distance" can be realized in a wide area with a few LoRa gateways, enabling smart-city applications (e.g., vehicle sharing [47], environment monitoring [40, 41], metering, logistics)?
- For link-information based localization, the critical question is whether an end-node can be accurately localized with LoRa link fingerprint in a wide area and sparse deployment.

With the answers to these questions, a fine-grained link-level measurement can benefit the de-

ployment of LoRa gateways, service quality in mobile applications, and network management in practice.

**Status Quo and their Limitations:** Several works [10, 79, 80] have observed the spatial diversity of LoRa links, but lack detailed analysis in different distance scales. To our best knowledge, no work reports the temporal performance of LoRa links. Similarly, to answer the coverage question, some measurement studies [2, 47, 52, 81–84] deployed real LoRaWAN systems to study the coarse-grained communication range in real environments. For example, Liando et al. [52] deployed three LoRa gateways and more than 50 static LoRa end nodes in a  $3 \times 3 \text{ km}^2$  campus environment to conduct a coverage measurement. And they further use a 70% packet delivery ratio (PDR) as a threshold to approximate the communication range of a LoRa link. The results show that the maximum communication range is 9.08 km and 2 km in Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) scenarios. However, with only a communication range, the communication heterogeneity [79, 85] will cause significant uncertainty in the coverage area for a gateway. Thus, the coverage problem is not fully addressed.

Compared to the energy-consuming GPS-based localization, LoRa link fingerprint based localization consumes much less power at the expense of accuracy. To answer the localization question, the SOTA LoRa localization method, SateLoc [11] reported a median localization error of 43.5 m in a  $350 \times 650 \text{ m}^2$  urban area with three gateways. However, the size of the evaluated area is limited, and the cost of dense gateways deployment is unaffordable. Thus, whether we can achieve the same localization accuracy in a larger area and with sparsely deployed gateways is still questionable.

**Challenges:** To achieve fine-grained spatial-temporal dynamics, coverage, and localization measurement, the key information is to obtain the link PDR and signal fingerprint at a fine-grained geography scale. We take a  $6 \times 6 \text{ km}^2$  area as an example to demonstrate the difficulty of obtaining such information. If we split the whole area into  $100 \text{ m}^2$  (i.e., the geography scale) cells and deploy a LoRa end node in each cell, 3,600 LoRa end nodes are required. The number of LoRa end nodes increases as the geography scale becomes more fine-grained. The expensive cost makes a static deployment impossible to achieve the fine-grained link-level measurement.

In this paper, we deploy a mobile LoRaWAN system and propose novel methods to measure the LoRa link-level coverage area and localization accuracy in a wide urban area at a fine-grained geography scale. Our deployed mobile LoRaWAN system consists of two LoRa gateways and six mobile LoRa end nodes in a  $6 \times 6 \text{ km}^2$  urban area, which continuously transmits data packets with the location information while they are moving. Although benefiting from the mobility of the LoRa end nodes, thousands of LoRa links are recorded efficiently, covering a variety of different locations, we still encounter two challenges to achieve the fine-grained and whole-area measurement. On the one hand, since a LoRa end node keeps moving, it needs time to observe enough packets for PDR calculation, but it travels for a distance as well. Such mobility leads to a granularity tradeoff between the PDR calculation and the geography scale. On the other hand, the users carrying the mobile end nodes moved freely in their daily life, without any requirement for movement. Thus, the locations of the collected data are not uniformly distributed across the areas of interest. Although we have available data records over four months, some locations and roads are still uncovered. In such areas, it is not trivial to infer the coverage performance and establish a fingerprint map for localization.

To solve the first challenge, The PDR granularity indicates the PDR estimation precision we can achieve by observing different numbers of packet transmissions. The more packets we count, the higher the precision is. For example, the precision will be 0.1 if we only count 10 packets in total, but it will be 0.01 if 100 packets are counted. We estimate the speed of each LoRa end node (§3.3.3), then adaptively adjust the geography scale to ensure the PDR granularity is not higher than 0.1 (§3.4.1). Moreover, we adopt DeepLoRa [79] to generate the expected signal power (ESP) [10] for every location in the area. DeepLoRa [79] is a deep neural network (DNN) based ESP estimation model to predict accurate ESP values by taking a land-cover type sequence as input. For coverage, with the calculated PDRs in the covered locations, we establish an ESP based PDR prediction model to infer the PDRs in the uncovered locations (§3.4.5). For localization, we use the ESPs observed by different gateways as fingerprints to generate a fingerprint map for each location.

With the ESP, PDR, and fingerprint map, our link-level measurement includes the following aspects. First, with the ESPs and PDRs in the covered locations, we analyze the overall, spatial

and temporal link dynamics for link property analysis (§3.4.3 and §3.4.4). Second, we estimate the coverage area of each gateway with/without link ESP gains (§3.5). Third, we study the localization accuracy with the fingerprint map under various settings (§3.6). Our measurement study presents three key observations, and the conclusions are as follows:

- The distance cannot reflect the link quality anymore, and the temporal link behavior is much more dynamic due to the micro-environment changes.
- Although the maximum communication range of a gateway observed by us is over 3 km, its actual coverage area is irregular and only about  $11.3 \text{ km}^2$ , which is much less than expected.
- The fingerprint-based LoRa localization accuracy is quite limited in sparse gateway deployment. More gateways, site-survey, and dynamic calibration are needed.

We summarize our contributions as follows:

- We deploy a real mobile LoRaWAN system in an urban area and measure massive LoRa links over four months.
- We propose several methods to measure spatial/temporal link dynamics and enable coverage area calculation using sparsely received LoRa packets.
- We report the localization accuracy in such a wide-area deployment providing more insights for future localization method design in LoRaWAN.

## 3.2 Related Work

**LoRa Link Dynamic Study.** To estimate the coverage of LoRa gateways without the deployment and on-site measurements, Demetri et al. [10], SateLoc [11], and DeepLoRa [79] develop different models to accurately estimate the signal path loss by understanding the impact of land-cover types in an urban environment. And a variety of remote sensing techniques are adopted to recognize land covers through the LoRa link. For example, Demetri et al. [10] first design an automated processing toolchain with the multi-spectral images from remote sensing and then apply the Okumura-Hata formula [8] for path loss prediction. Similarly, SateLoc [11] proposes a segmented Bor model [1] to capture the different path loss exponents with corresponding land covers. DeepLoRa [79] further incorporates the deep learning techniques for LoRa link estimation. It de-

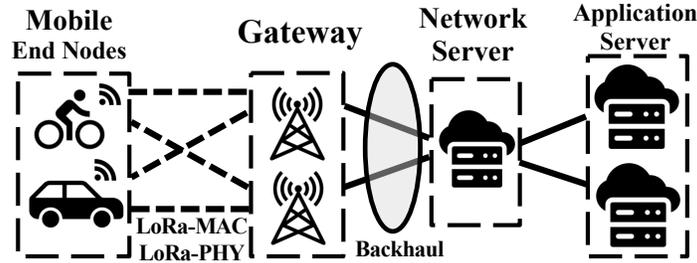


Figure 3.1 Illustration of our LoRaWAN architecture.

velops a land-cover aware path loss model based on the Bi-LSTM (Bidirectional Long Short Term Memory) and reduces the estimation error to less than 4 dB, which is  $2\times$  smaller than state-of-the-art models [10]. In contrast, we study the relationship between a path loss and the resulting PDR, which is crucial in bridging the gap between link behavior and network coverage.

**LoRa Coverage Measurement.** Recent years have witnessed several measurement works [5, 7, 47, 52, 80, 86, 87] to reveal the LoRaWAN performance in real environments. Liando et al. [52] deploy three gateways and more than 50 static end nodes in a  $3\times 3\text{ km}^2$  campus to study the LoRaWAN performance for measurement, including the communication range, network throughput, and energy efficiency. Results show that the LoS and NLoS communication ranges are  $9.08\text{ km}$  and  $2\text{ km}$ , respectively. Similarly, Centenaro et al. [5] observe a communication range of  $2\text{ km}$  in an area of high buildings. And the communication range they reported varies from  $1\text{ km}$  to  $20\text{ km}$  in the central business district [7]. Besides, LoSee [47] adopts a mobile end node mounted on a bike to study the LoRaWAN coverage ability on the campus scale (e.g.,  $4.5\text{ km}^2$ ). For reliable PDR calculation, the mobile end node must transmit 50 to 100 packets on the spot. Focusing on the indoor environments (e.g., office buildings, residential buildings, car parks, warehouses), Xu et al. [86] study the LoRa link behavior and energy profile by deploying ten static and two mobile LoRa end nodes. Compared to these measurement studies only focusing on the spatial link behavior, we analyze the temporal characteristics of LoRa links and provide a more fine-grained coverage area study than existing works in a  $6\times 6\text{ km}^2$  urban area.

**LoRa Localization Method.** Studies mainly adopt two kinds of technologies for LoRa localization: 1) TDoA-based localization; 2) RSSI-based localization. TDoA-based approaches uti-

lize the time differences of the same signal arriving at different gateways. TDoA has been implemented in the LoRaWAN network to perform localization both for stationary [88] and mobile scenarios [89–91]. However, due to the limited bandwidth of commercial LoRa end nodes, TDoA-based localization error can reach hundreds of meters since only  $\mu s$ -level time resolution [92,93] can be achieved. Researchers have improved TDoA-based localization to meter-level by customizing dedicated LoRa devices. Nandakumar et al. [94] proposed a multi-band LoRa backscatter device based on CSS modulation. Bansal et al. [95] present a distributed software-radio-based station network that spans a wide bandwidth encompassing the TV whitespaces and offers a high aperture. Those approaches, however, cannot be applied directly in existing LoRaWAN systems. Besides, TDoA-based systems require at least three gateways that are strictly time-synchronized or phase-synchronized which is not applicable in scenarios with sparse gateway deployment.

We can utilize received signal strength indicator (RSSI) measurements for localization [48] according to the path loss models mentioned above [1, 9]. However, the performance is highly affected by channel dynamics in complicated environments [10, 11, 79]. Fingerprint-based approaches [49, 50, 96, 97] also use RSSI values as a fingerprint to locate an end node by matching its fingerprint with known reference locations in the database. Machine learning approaches have been adopted for fingerprint matching, such as k-Nearest-Neighbor (KNN) [49], SVM [96], Bayesian inference [50, 97]. SateLoc [11] proposes a weighted combination strategy for multi-gateway likelihood maps based on fingerprint matching and selects the point with the highest likelihood as the predicted location. SateLoc achieves a 43.5 m median localization error in a 227,500 m<sup>2</sup> urban area. Based on our LoRaWAN setting, we adopt link RSSI localization which is similar to SateLoc and provide a detailed localization comparison with the data collected from our mobile LoRa system.

### 3.3 System and Dataset Overview

In this section, we first briefly review the LoRaWAN technical specification and define the LoRa coverage problem. An overview is then given on the system architecture, configuration, and deployment. Finally, we show the measurements and analysis results from our deployed mobile LoRa system.

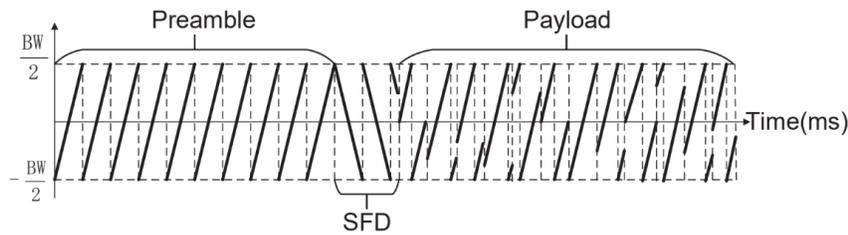


Figure 3.2 The structure of a LoRa packet.

### 3.3.1 LoRaWAN Primer

We illustrate the architecture of LoRaWAN in Figure 3.1, which operates in the infrastructure mode. Multiple LoRa end nodes run the LoRa-MAC (media access control) and LoRa-PHY protocols and connect to the gateways in their communication range. Transport protocols like TCP, 6lowpan, and COAP is not involved in the LoRaWAN networking stack yet. Hence, we mainly focus on the link layer performance. Upon receiving the LoRa packets, gateways forward them to LoRa network servers for further processing. Note that there is no energy constraint on the gateway in most scenarios [77, 98]. Since the connection between gateways and network servers are usually cellular networks or wired networks. As the packet forwarder, gateways also forward the control messages (e.g., PHY configurations, MAC settings) from network servers to end nodes. Finally, network servers filter duplicated LoRa packets and disseminate the valid ones into application servers for different applications.

As for LoRa networking, LoRa-PHY uses CSS to modulate data symbols. Figure 3.2 shows the structure of a LoRa packet, which consists of the preamble, start frame delimiter (SFD), and payload. Specifically, the preamble consists of multiple base up-chirps, followed by the SFD with 2.25 base down-chirps for packet detection and alignment. The payload contains multiple modulated chirps with different shifted initial frequencies for encoded data bits. In LoRa-PHY, three parameters (i.e., bandwidth (BW), spreading factor (SF), and coding rate (CR)) can be configured to adapt the communication range. For example, BW determines the frequency range of a chirp symbol, such as 125, 250, and 500  $kHz$ , in which a small BW corresponds to an extensive communication range [77]. SF denotes the data bits a chirp symbol represents, ranging from 7 to 12.

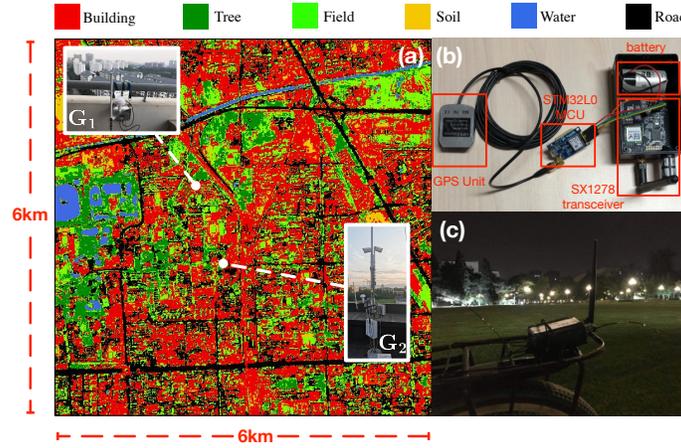


Figure 3.3 We deploy two gateways and six mobile nodes in the urban areas, covering various land cover types.

The communication range gets larger as the SF increases and enhances the noise resilience of LoRa signals. Besides, CR introduces data redundancy in the coding process for extra noise tolerance, which can be assigned as 4/5, 4/6, 4/7, and 4/8.

Sitting upon LoRa-PHY, LoRa-MAC adopts an ALOHA-based protocol that allows end nodes to transmit as soon as they wake up, and exponential back-off is involved in case of collisions. However, ISM bands regulation imposes a maximum 1% transmission duty cycle to end nodes and gateways when using an ALOHA MAC. As a result, it puts a significant limitation on the downlink capacity of the gateways as they need to serve all the surrounding end nodes with relatively scarce transmission opportunities.

### 3.3.2 Our System Overview

We first introduce the hardware and deployment of our mobile LoRaWAN system. Illustrated in Figure 3.3, two gateways  $G_1$  and  $G_2$  and six mobile end nodes (e.g., bicycle, car) are deployed in the  $6 \times 6 \text{ km}^2$  urban area. Both gateways are equipped with an MCU, an SX1276 transceiver [76], and a Raspberry Pi 3 for programming remotely. We further indicate the location of our two gateways  $G_1$  and  $G_2$  in the campus as white points in Figure 3.3(a), which are located at the rooftop of two different buildings at the height of 84 m and 68 m, respectively. Note that the ground altitude of the campus area is about 52 m, and the distance between  $G_1$  and  $G_2$  is 1332.14 m. The gateways are

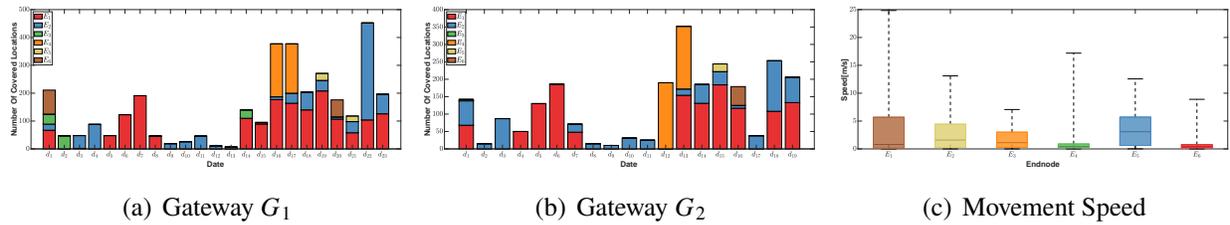


Figure 3.4 The accumulated number of different locations observed by (a)  $G_1$  and (b)  $G_2$  across different users on different days. (c) Movement speed distribution of six mobile end nodes.

powered by PoE (Power over Ethernet) and provided with Internet access. Thus it can forward the LoRa packets to our network and application servers running on the cloud (e.g., Digital Ocean).

On the transmitter side, the LoRa end nodes are implemented with an MCU, an SX1278 transceiver, and a GPS unit, as shown in Figure 3.3(b). Figure 3.3(c) illustrates the 5 LoRa end nodes mounted on different bicycles, and the remaining end node is put inside a BYD car under the front windshield glass. These end nodes move freely with the bicycles/car in the users' daily life without any constraints, they send a packet to the gateways every five seconds only when they are moving for power efficiency.

By default, our experiment uses the spreading factor  $SF = 12$ , bandwidth  $BW = 125 \text{ kHz}$ , and coding rate  $CR = 4/5$ . We enable a regulation-compatible power amplifier controlled by the register PA\_HP [76] and connected to the pin PA\_BOOST [76] on the SX1278 transceiver. The total transmission power reaches  $19 \text{ dB}$ , which complies with LoRa regulations. The operating channel is set as  $486.3 \text{ kHz}$ ,  $486.5 \text{ kHz}$ ,  $486.7 \text{ kHz}$ ,  $486.9 \text{ kHz}$ ,  $487.1 \text{ kHz}$  and  $487.3 \text{ kHz}$ , respectively. Thus we can avoid potential packet loss due to collisions between different end nodes. The experiment spans four months, during which the end node owners traveled as usual (e.g., eating, office, home). Thus the collected data records can only cover several parts of the whole area. To obtain the land-cover types in this area for the LoRa based localization, we apply the satellite remote sensing imaging on the whole area of interest by following the instructions in existing works [10, 11, 79], including buildings, roads, parking lots, lakes, a river, grassland, trees, and playground.

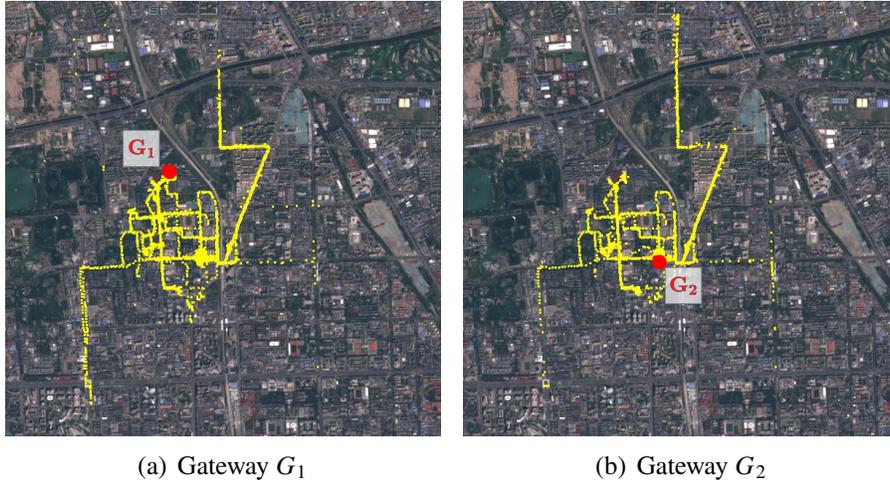


Figure 3.5 The spatial distribution of data records in the view of  $G_1$  and  $G_2$ .

### 3.3.3 Collected Dataset Overview

This section provides detailed instruction on our collected LoRa dataset, spanning from Dec 22 to Mar 15. Considering the fast movement of an end node, the transmission interval between two adjacent packets is set as 5s. We encode the GPS coordinates, timestamps, and sequence numbers into the payload of LoRa packets. And the corresponding SNR and RSSI are logged at the gateway side. Upon receiving the packets, the logged data records can be extracted from the network server to keep the duplicate packets at both gateways, delivering over 30,000 records in total. Besides, we can calculate the link distance and the height difference between the end node and gateway pair by decoding the GPS data in the payloads.

We further illustrate the measuring locations on the main roads of the  $6 \times 6 \text{ km}^2$  urban area, shown in Figure 3.5. The yellow points and red ones indicate the moving end nodes and the gateway  $G_1$  and  $G_2$  as the receivers for successful packet transmissions. And the maximum communication range of  $G_1$  and  $G_2$  can be larger than 3 km. Additionally, we observe similar trajectories of end nodes for  $G_1$  and  $G_2$ , but the PDR of an identical road is quite diverse. For example, both  $G_1$  and  $G_2$  have poor performance on the right-center roads in common. However,  $G_1$  has better coverage for the left-bottom road while  $G_2$  performs better on the middle-top road, especially the part in the north of a river on the top. The observation shows that the maximum communication range is too coarse-grained to understand the coverage of LoRaWAN in an urban area, and a finer-grained

report on the measurement of LoRaWAN is required.

To demonstrate the coverage of both gateways statistically in our mobile LoRa system, we show the total number of covered locations with successful transmissions in terms of end nodes and days, shown in Figure 3.4. Specifically, we use a  $10 \times 10 \text{ m}^2$  square block to define our “locations”. Thus the whole area can be divided into  $600 \times 600$  locations. For each end node, we calculate the total number of various trajectories with corresponding transmitting locations. For example, if more than one packet is received in a new location, we count once for the current end node and derive the total covered locations. Figure 3.4(a) and 3.4(b) show that the covered locations by  $G_1$  on 23 different days, while  $G_2$  observes data for 19 days. Regarding the successful transmissions for each day, the maximum and minimum locations observed by  $G_1$  are 452 and 7. In contrast,  $G_2$ 's maximum and minimum observed locations are 352 and 10, respectively. From the view of mobile end nodes, end nodes 1 (e.g., red) and 2 (e.g., deep blue) contribute the most data records in different locations on most days. And other nodes demonstrate a varied covered location. For example, end node 4 (e.g., orange) only delivers the most covered locations in two days.

To measure the mobility of our end nodes, we further calculated the speed of each end node by using the timestamps between two adjacent locations in a trajectory. The speed distributions (i.e., min, 25%, median, 75%, and max) of different end nodes are shown in Figure 3.4(c). The maximum observed speed is about  $25 \text{ m/s}$  ( $90 \text{ km/h}$ ) from the end node 1 (i.e., the BYD car). The median speed is less than  $5 \text{ m/s}$  ( $18 \text{ km/h}$ ) for most nodes, which is reasonable for a bicycle. Note that data records related to end node 1 are taken during the morning and afternoon traffic peak hours. Since LoRa-PHY is resilient to the Doppler effect [52] in the range of our observed speed, we can use these data records to estimate an equivalent PDR for different locations for transmitting nodes.

### 3.4 Link Behavior Study

Given our collected dataset with mobile LoRa nodes, we study the LoRa link behavior in the urban area. Two metrics, ESP and PDR, are adopted to indicate the signal path loss over a physical channel and reliable coverage in an area. By carefully analyzing their spatial and temporal distri-

butions, we establish a PDR prediction model that associates a position’s computed ESP value to the estimated PDR.

### 3.4.1 Estimation Methodology on Metrics

**ESP Estimation.** We use ESP to depict the LoRa signal attenuation over a long-distance transmission. Although RSSI is a widely adopted indicator to measure the signal attenuation of a physical link in WSNs [99–101] and Wi-Fi [102], it can be more error-prone below the noise floor in LoRaWAN. Thus, we choose ESP which combines RSSI and SNR to calibrate the expected signal path loss in our measurement study, which be calculated as follows [49, 79]:

$$\text{ESP} = \text{RSSI} + \text{SNR} - 10 \log_{10}(1 + 10^{0.1\text{SNR}}) \quad (3.1)$$

where RSSI is the received signal strength indicator, and SNR is the signal-to-noise ratio. Given a received data packet, its RSSI and SNR will be automatically calculated by gateways forwarded to the network server.

**PDR Estimation.** Given a PDR threshold, the PDR of nodes with each position can be used to determine the coverage of our mobile LoRa system. Due to the mobility of the end nodes, the data packets are scattered along various trajectories. Our basic idea is to utilize all trajectories that pass the position based on their coordinates to calculate the PDR of a specific position.

Given this trajectory-based PDR estimation method, a trade-off should be considered between the position granularity and the estimation accuracy. On the one hand, a fine-grained position granularity is desirable so that the micro-differences can be reflected across the observed “positions” by our mobile end nodes. On the other hand, the number of available trajectories can be reduced for observed locations if we split the urban area at a highly finer-grained scale to represent a position. Consequently, the PDR accuracy of mobile end nodes can suffer from the estimation bias from limited trajectories. For example, assuming the true PDR of a position is 90%, the calculated PDR is only 80% due to one packet loss of five observed packets. More than ten packet records are required for each position to mitigate the scarce trajectory distribution.

In practice, we first divide the  $6 \times 6 \text{ km}^2$  area into 1,600  $150 \times 150 \text{ m}^2$  square blocks. Each block

represents a position denoted as  $p(i, j)$  to balance the estimation granularity and the estimation error, where  $i$  and  $j$  are the coordinates of the corresponding block. Assuming the average speed of an end node is  $3 \text{ m/s}$  from Figure 3.4(c), the packet interval between two adjacent transmissions is set as  $5 \text{ s}$ . Thus the end node can travel through  $150 \text{ m}$  for ten continuous packet records.

Upon receiving the LoRa packets at the gateway side, we first extract all trajectories for each end node. Then, we estimate all  $n$  positions that a trajectory  $t$  covers. For the  $k^{\text{th}}$  position  $p_t(k)$  of trajectory  $t$ , we use the sequence numbers of the data records to count the total number of transmitted LoRa packets passing through the current position, denoted as  $c_t(p_t(k))$ . And the number of successfully received LoRa packets is denoted as  $c_r(p_t(k))$ . The trajectory  $t$  only contributes a valid PDR estimation as  $c_r(p_t(k))/c_t(p_t(k))$  for the position  $p_t(k)$  when  $c_t(p_t(k))$  is larger than 10. When we traverse all trajectories to compute their PDR estimations for the covered positions, we calculate the average value with all PDR estimations for each position.

Furthermore, we adaptively enlarge the splitting area of a position where the observed packet is less than five but not 0. Specifically, if the total number of packet transmissions is less than 5 for position  $p_t(k)$ , we keep increasing the area of the position by adding its adjacent blocks until more than 5 data records are reported. For  $G_1$ , the blocks of packets less than 5 take 12.16% of all the blocks number. For  $G_2$ , the blocks of packets less than 5 take 12.97%. In this way, we deliver a reliable PDR estimation for those covered positions with one or two lossy trajectories (e.g., the right-middle roads for  $G_1$  and  $G_2$  in Figure 3.5).

### 3.4.2 Overall PDR and ESP Distribution

We further demonstrate the estimated PDR and ESP across different positions for  $G_1$  and  $G_2$ . Illustrated in Figure 3.6(a), the CDFs of PDR are distributed similarly for  $G_1$  (e.g., blue dashed curve) and  $G_2$  (e.g., solid red curve). In comparison,  $G_1$  provides a little better PDR for the covered positions than  $G_2$  does. And 60% of links are reliable with a PDR higher than 90% for  $G_1$ . And the remaining 40% LoRa links are intermediate links with dynamic link behaviors.

Figure 3.6(b) further shows the CDFs of ESP in all recorded data packets. We can observe that the minimum ESP is  $-142.3 \text{ dBm}$  for all packets, which is consistent with the reported  $-148 \text{ dBm}$

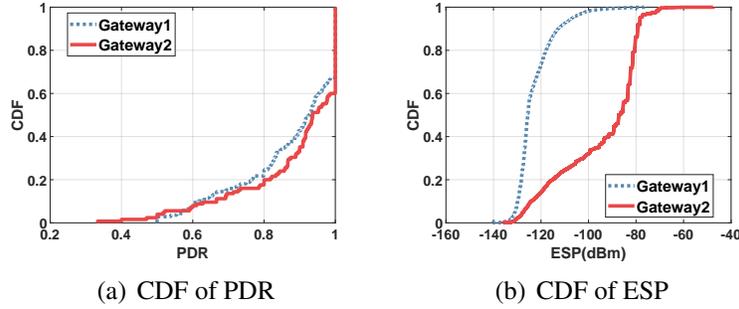


Figure 3.6 CDFs of PDR and ESP observed at  $G_1$  and  $G_2$ .

for the sensitivity of SX1276 [76]. Notice that LoRa gateways with different transceiver types definitely receive signals at different sensitivity levels, resulting in a varied link budget. Compared with  $G_1$ , the ESP observed at  $G_2$  is much higher. For example,  $G_1$  has 20% ESP higher than  $-120$  dBm and the maximum ESP is  $-80$  dBm. However, 80% ESP of  $G_2$  is higher than  $-120$  dBm, and the maximum ESP is approaching  $-47.34$  dBm. As shown in Figure 3.3(a), we attribute the ESP difference to the deployment environment.  $G_1$ 's antenna is partially hidden by the wall and railing while there is no obstacle for  $G_2$ .

**Remark.** Figure 3.6 reflects the distribution inconsistency between PDR and ESP. Due to the strong noise tolerance ability of LoRa, low ESP (e.g., median value  $-127$  dB) can achieve similar PDR distribution as high ESP (e.g., median value  $-87$  dB) does.

### 3.4.3 Spatial PDR and ESP Distribution

We study the spatial distribution of PDR and ESP regarding the link distance. For each position (e.g.,  $150 \times 150$   $m^2$  block), the distance between the center of the block to a gateway location is first calculated as its distance. And we use the GPS coordinates to compute the distance between the end node and a gateway for each data packet. The spatial PDR distribution is shown in Figure 3.7. A similar spatial distribution can be observed at  $G_1$  and  $G_2$ , where the intermediate links with low PDR are scattered at all distance levels. We further illustrate the spatial ESP distribution in Figure 3.8. As the distance increase, the ESP values are reduced for both  $G_1$  and  $G_2$  and scattered in a relatively wide range at different distance levels. Specifically, the maximum range of ESP values is from  $-140$  dBm to  $-95$  dBm at  $G_1$  in Figure 3.8(a), when the distance is about 1,000 m.

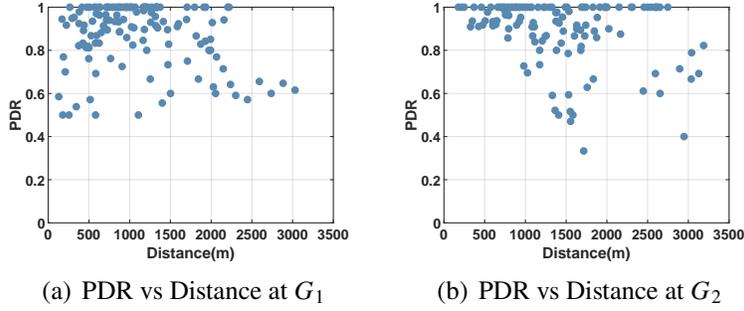


Figure 3.7 The spatial distribution for PDR and distance.

In contrast, it is from  $-139 \text{ dBm}$  to  $-100 \text{ dBm}$  for  $G_2$  in Figure 3.8(b). Additionally, the longest distance observed by ESP is about  $3.5 \text{ km}$ , which is longer than  $3.2 \text{ km}$  observed by PDR. The main reason is that the data records reported at those long-distance positions are from the end node 1 (i.e., the car). And it becomes hard to observe enough data records in our defined position area due to the high mobility, resulting in a failed estimation of PDR in long-distance areas.

**Remark.** The distance of a LoRa link is weakly associated with its PDR and ESP. A rough estimation of ESP can be given with the link distance (Figure 3.7), but the link distance cannot be used for fine-grained PDR prediction (Figure 3.8).

### 3.4.4 Temporal PDR and ESP Distribution

The temporal distribution of PDR and ESP is evaluated for transmission days. We first associate the trajectories per day to each position and then compute the standard deviation of per-day PDR values to depict the temporal PDR changes for each position. As for ESP, we first divide the whole area into  $360,000 \text{ } 10 \times 10 \text{ m}^2$  blocks and then calculate the average ESP of the associated data records to represent the ESP of the block. The standard deviation of ESP values can be further derived for each block.

We show the CDFs of PDR and ESP deviation in Figure 3.9(a) and 3.9(b), respectively. On the one hand,  $G_1$  and  $G_2$  exhibit analogous temporal deviation on PDR and ESP. For example, 30% of positions have more than 5 dB variance for ESP. And the maximum ESP deviation is about  $-15 \text{ dB}$ . Besides, more than 10% variances of PDR are reported for 40% of positions. And the maximum PDR deviation is larger than 30%. On the other hand, the only difference in temporal distribution

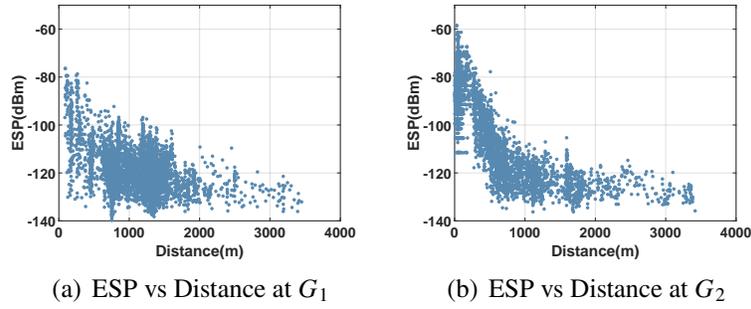


Figure 3.8 The spatial distribution for ESP and distance.

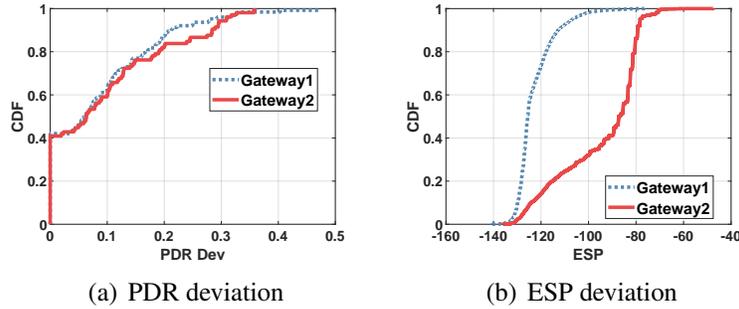


Figure 3.9 The standard deviation of ESP and PDR observed on different days.

over time is from the micro-environment (e.g., surrounding obstacles like other bicycles and cars), demonstrating the significant impact of the micro-environment patterns on the link performance for different end nodes.

**Remark.** LoRa links are highly dynamic over time in an urban environment, shown in Figure 3.9, which can be attributed to the frequently varying micro-environment [80].

### 3.4.5 ESP based PDR Prediction

Based on the above observations on PDR and ESP distributions, we build a PDR prediction model by feeding ESP as input. First, we calculate the average ESP of all observed data records for each associated position in the urban area. Given the measured PDR for covered areas, we obtain a variety of pairs of PDR and ESP. Then, the Gaussian process regression (GPR) [103] is adopted to predict the numerical PDR from ESP for those uncovered areas.

To achieve a more accurate regression learner, we choose the exponential function as the kernel function and complete the fitting processing, shown in Figure 3.10. Statistically, the root-mean-

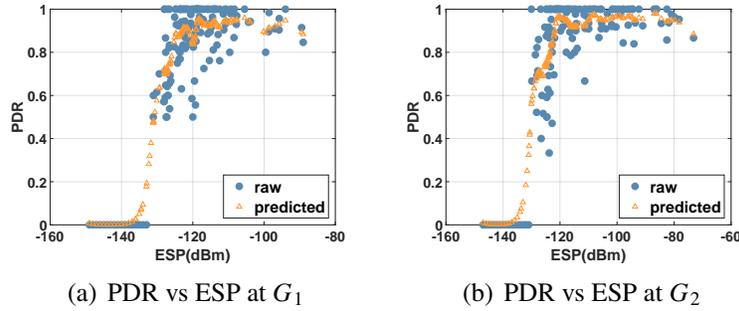


Figure 3.10 Gaussian process regression analysis between PDR and ESP at  $G_1$  and  $G_2$ .

square error is 0.12448 and 0.13678 for  $G_1$  and  $G_2$ , and the coefficient of determination is 0.84 and 0.82, respectively. From the raw data pairs (e.g., blue dot), when ESP is lower than  $-133$  dBm and  $-131$  dBm for both gateways, the measured PDR nears 0 based on our measurement study. Additionally, a 11 dB wide transition zone (i.e.,  $[-131$  dBm,  $-120$  dBm]) can be observed in both  $G_1$  and  $G_2$ , which is larger than a 3 dB transition zone in WSNs [101]. The reason is that in LoRa long-distance communication, LoRa links are affected by more complicated factors and are less predictable with only ESP, thus introducing more ambiguity. Even when the ESP is larger than  $-120$  dBm, the PDR achieves a high performance but is not always 100%. And it can decrease below 70% due to a large temporal variance of PDR and ESP observed in §3.4.4. As for the uncovered areas with the given ESP, the predicted data points (e.g., yellow triangle) exhibit a good match with the ground truth. However, it cannot reflect the dynamic PDR accurately in our mobile LoRa system.

**Remark.** ESP is a relatively good indicator to predict the PDR of a position. A 13 dB transition zone and the PDR dynamic under large ESP indicate LoRa links are less predictable than other wireless techniques like Wi-Fi and Zigbee.

### 3.5 Coverage Area Study

#### 3.5.1 LoRa Coverage Problem

The coverage area indicates where a gateway can reliably communicate with any end node and is determined by LoRa-PHY and LoRa-MAC. The influence of LoRa-PHY on coverage is explicit. LoRa-PHY determines a signal-to-noise ratio (SNR) threshold, under which LoRa chirp

symbols cannot be decoded correctly. The SNR thresholds are determined by different LoRa-PHY configurations [77]. The observed ESP of various LoRa links is related to their distance. Thus, LoRa-PHY determines the link reliability for LoRa transmissions.

Besides, LoRa-MAC may influence the coverage, too. For example, LoRa-MAC determines collision probability when multiple LoRa end-nodes are deployed in the same area and share an identical gateway. WiChronos [104] reported that when an end node transmits a 1-byte message every ten minutes, the collision probability is about 1.4% for 100 nodes, increasing to 12.75% for 1000 nodes. However, the influence of collision on the coverage is implicit since the collision is not determined by link distance but by the transmission schedule. If the schedule is not well adjusted, the end nodes far from the gateway may not have a higher collision probability than the end nodes near the gateway even if the transmission of the far end nodes is using a longer signal on-air time (e.g., larger spreading factor). Therefore, if the transmission schedules of all end-nodes are uniformly random, the collision will uniformly degrade the transmission reliability for long and short links, making it stained for part of the LoRa-PHY covered area. Many works [98, 105] focus on solving the collided LoRa signals to enhance the LoRa transmission reliability.

In our measurement work, we focus on the LoRa-PHY coverage to determine the maximum area a LoRa gateway can cover. By adjusting the channel of each mobile end node to a different frequency 3.3.2, there is no signal collision in our collected datasets.

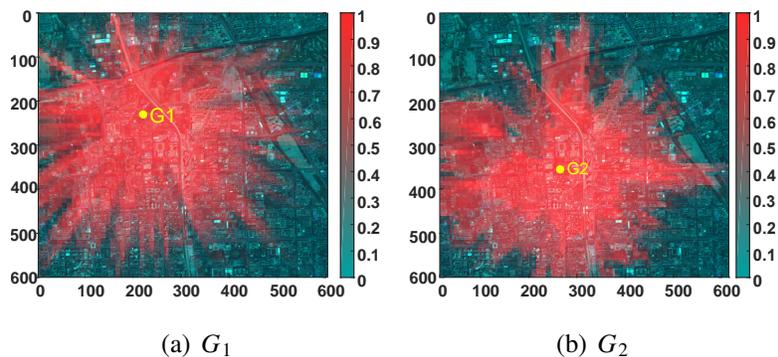


Figure 3.11 The heatmap of PDR values for different positions in the urban area.

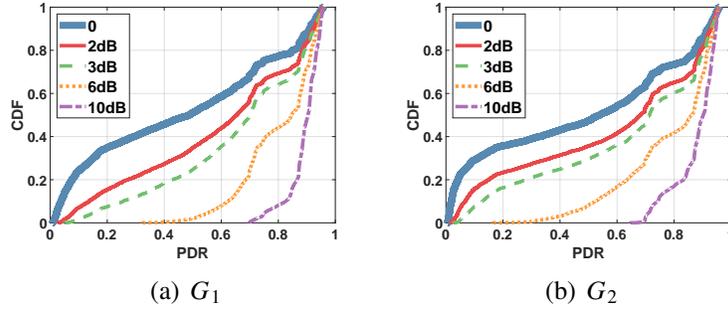


Figure 3.12 CDF of predicted PDR with different ESP gains.

### 3.5.2 Methodology and Implementation

In this section, we study the coverage of each gateway in our deployed mobile LoRa system. And the coverage area is defined as the covered area whose sum of the positions with a PDR value larger than 70%. Specifically, by dividing the urban area into “positions” ( $150 \times 150 m^2$ ), we first compute the corresponding PDR with our data records for those covered areas. We first adopt DeepLoRa [79] to estimate an average ESP for each position for those uncovered ones. Then, we can predict the associated PDR based on the PDR-ESP regression model in § 3.4.5. Figure 3.11 illustrates the distribution of calculated and predicted PDR values for all positions in the urban areas. We can observe an irregular PDR distribution in different directions for both  $G_1$  and  $G_2$ . And the covered positions for  $G_1$  and  $G_2$  are distributed non-uniformly. Statistically, the coverage area of  $G_1$  and  $G_2$  is  $11.4 km^2$  and  $11.6 km^2$ , respectively, far from covering all  $6 \times 6 km^2$  reliably.

### 3.5.3 Coverage Improvement ESP Gain

To enhance the coverage area of each gateway in the wild, several systems [50, 106, 107] have been proposed to cooperate with multiple gateways for extra SNR gains of received LoRa signals. For example, an SNR gain of  $2 \sim 3 dB$  can be achieved through the coherent combining across three or more gateways [50, 106]. Equation 3.1 shows the SNR gain is equivalent to the ESP gain. To quantitative the relationship between the ESP gains and the coverage area in our deployed system, we manually add an ESP gain for each position and then recalculate the corresponding PDR under the enhanced ESP. For fairness, different ESP gains from  $2 dB$  to  $10 dB$  are selected randomly, resulting in the CDF of predicted PDR in Figure 3.12. As the extra ESP gains go up, the PDR

increases as well. For example, the median PDR improvement can reach 48.6% to 62.8% at  $G_1$  with a 3 dB ESP gain, shown in Figure 3.12. And it gets larger from 50.3% to 62.7% when the ESP gain is 10 dB, delivering a covered area with all PDR values larger than 70%. The observations in Figure 3.12 verify the effectiveness of the SNR enhancement method.

Table 3.1 Coverage area under different ESP gains.

ESP Gains (dB)	0	2	3	6	10
$G_1$ Coverage Area ( $km^2$ )	11.4	15.2	17.7	27.1	35.9
$G_2$ Coverage Area ( $km^2$ )	11.6	15.3	17.3	23.7	33.0

Illustrated in Table 3.1, we further adopt the enhanced PDR to calculate the coverage area. And a steady improvement of the coverage area can be observed at  $G_1$  and  $G_2$  as the ESP gains increase. Given the 2 dB ESP gains, the coverage area can be increased by 32.6%. And we can approximately cover the whole urban area of  $6 \times 6 km^2$  via only one gateway, with a given ESP gain of 10 dB.

**Remark.** Due to the observed link dynamics, the coverage area of a gateway is usually irregular. Beyond deploying new gateways, it can be more effective to enlarge the coverage area of a gateway by capturing extra SNR gains of LoRa signals.

### 3.6 Localization Accuracy Study

#### 3.6.1 Methodology and Implementation

Recent years have witnessed a variety of localization systems [11, 47, 79, 108–111] built on the knowledge of LoRa link behaviors with path loss. Among them, SateLoc [11] is the SOTA method. The basic method is illustrated as follows:

Suppose we have several gateways to cover a certain area for localization. Each gateway will generate an ESP map as a part of the fingerprint map. The whole area is split into many geography cells, which indicate the location unit in the localization process. Given the  $m^{th}$  gateway's ESP map, the likelihood of  $L_{m,i}$  for the  $i^{th}$  cell that an end node  $e$  is located can be formulated as follows:

$$L_{m,i} = 1 - \frac{|\bar{E}_{m,e} - E_{m,i}|}{\max(|\bar{E}_{m,e} - E_m|) - \min(|\bar{E}_{m,e} - E_m|)} \quad (3.2)$$

where  $\bar{E}_{m,e}$  is the average ESP value of each packet, which is transmitted by the end node  $e$  and received at the  $m^{th}$  gateway.  $E_{m,i}$  is the ESP value predicted by path loss models at the  $i^{th}$  cell in the  $m^{th}$  ESP map. The likelihood is then scaled and normalized according to the value range of differences between received and ESP values in the  $m^{th}$  ESP map. Given the likelihood map for each gateway, the fingerprint-based localization leverages the joint likelihood of multiple gateways, in which the cell with the highest likelihood is selected as the predicted location:

$$Location = \arg \max_i \sum_{m=1}^M L_{m,i} \quad (3.3)$$

To evaluate the performance of LoRa link-based localization systems in our deployment, we implement SateLoc based on four different path loss models for ESP map generation, including Bor model [1], PATH/INTERSECTION [10], SateLoc [11] and DeepLoRa [79]. To obtain the remote sensing images for the environmental analysis for PATH/INTERSECTION, SateLoc, and DeepLoRa, we first use the Sentinel-2 open-access API to get multi-spectral images of 10  $m$  resolution for all four path loss models. The models are then trained with the collected dataset in our deployed system, delivering 2 ESP maps for both gateways. Each pixel in our ESP map corresponds to a  $10 \times 10 m^2$  cell in a real map. Note that the evaluated data points are filtered from the whole dataset, in which each packet record contains the ESP values of the same frame from the end node received at two gateways. Finally, we collected available data records covering 1,495 different  $10 \times 10 m^2$  locations.

### 3.6.2 Overall Comparison of Localization Accuracy

Illustrated in 3.14(a), the CDF of localization error is given for the comparison study of localization accuracy. On our dataset, with the most accurate DeepLoRa [79], the median localization error reaches up to 400  $m$  while adopting the approach in SateLoc [11], we got a median localization error of about 500  $m$ . The worst localization error of those state-of-the-art models can even reach 2,000  $m$ .

This localization accuracy is much worse than that reported by SateLoc [11]. The best accuracy achieved by SateLoc shows that 100% localization error is within 100  $m$  and the median localization

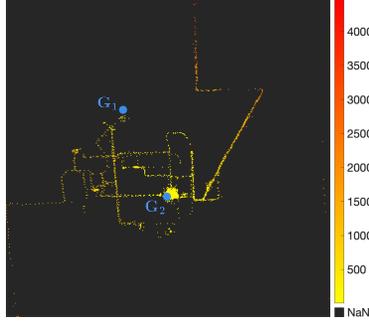


Figure 3.13 Spatial distribution of the localization errors.

error is  $43.5\text{ m}$  given the multi-spectral images of  $50\text{ cm}$  resolution for three gateways.

This is reasonable due to the property difference of the datasets used. On the one hand, only two gateways are deployed in our system, resulting in serious fingerprint ambiguity compared to three or more gateways. On the other hand, the localization accuracy is bounded by the resolution of cell splitting. Since the fine-grained link estimation is based on cell splitting and a cell is the smallest unit of distance comparison in our system. Therefore, a less fine-grained cell splitting can induce much higher localization errors in urban areas. For example, we compute the median localization error of 40 cells (which equals  $400\text{ m}$  since each cell is a  $10\text{ m} \times 10\text{ m}$  area). With a similar cell-wise error, we can get a median error of  $20\text{ m}$  if we have access to remote sensing of  $50\text{ cm}$  resolution which can outperform the localization error reported by SateLoc. Thus, improving the resolution of multi-spectral images can improve localization accuracy. Compared with other models, DeepLoRa achieves the best performance, consistent with the reported results [79]. Since DeepLoRa can provide more accurate ESP estimation than others, which can mitigate lots of fingerprint ambiguities. Besides, PATH/INTERSECTION has the worst performance among all approaches.

Given the two generate ESP maps for  $G_1$  and  $G_2$ , we further show the spatial distribution of the localization errors of DeepLoRa [79] in Figure 3.13. The lighter the color is, the smaller the localization error is. And the PDR reaches 0 for the black areas. An interesting observation is that the evaluated data records near  $G_2$  have the best accuracy while it suffers from the estimated data records near  $G_1$ . The evaluated data records far from both  $G_1$  and  $G_2$  have the worst localization performance. The reason has two folds. First, the ESP dynamic increases at distant locations. The ESP dynamic makes DeepLoRa hard to predict the ESP fingerprint accurately. Second, the ESP

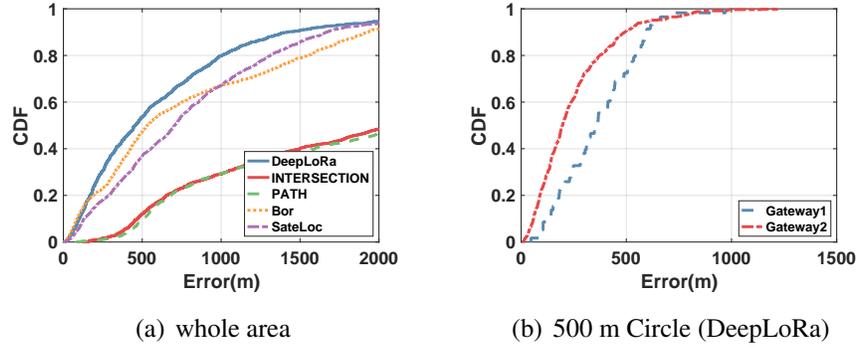


Figure 3.14 The CDF of localization errors under different ESP estimation methods in different ranges.

value is close to the LoRa sensitivity at long distances. The fingerprint ambiguity is increasing at many borderline areas in the whole area.

We further reduce the number of evaluated data records to see whether we can achieve a better localization accuracy when the evaluated data records are close to either  $G_1$  or  $G_2$ . We only select the cells whose distance from the gateway is smaller than 500 m. We use DeepLoRa to generate the two ESP map of  $G_1$  and  $G_2$ . Figure 3.14(b) shows that the data records around  $G_2$  have more accurate localization results than those around  $G_1$ . The reason is that the ESP observed by  $G_1$  is much more dynamic than  $G_2$  (Section 3.4.3). For  $G_2$ , the median localization error is about 220 m. Regarding the 500 m range, it is still hard to support fine-grained localization. As shown in Figure 3.15, we can detect different traffic trends under current median localization of 500 m by drawing part of the trajectories of a single end node. The trends of the predicted locations almost follow the actual movement of the end node when it stays around a gateway(a), moves across the blocks, or moves towards(b)/away(c) a gateway. It possible to apply the localization model to traffic trend prediction.

**Remark.** The ESP fingerprint-based localization highly depends on the granularity of the position unit, the number of gateways, and the distance to gateways. Given two gateways at  $100\text{ m}^2$  granularity, a sparse site survey can only achieve road-level localization for traffic trend tracking. Additionally, the dynamic nature of link ESP in urban areas degrades the localization accuracy.

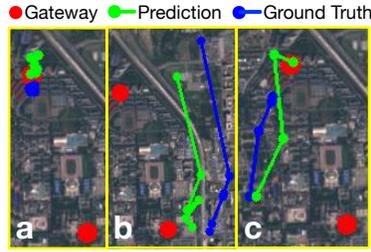


Figure 3.15 Tracking different traffic trends under median localization error of 500m.

### 3.7 Observations, Insight, and Discussion

**Observations.** We deploy a LoRaWAN with two gateways and six mobile LoRa end nodes. By taking advantage of mobility, we accumulate data records that last more than 20 days to cover a large area. Moreover, we develop a mobility adaptive method to achieve the PDR estimation and coverage area calculation. Based on our link behavior study, we further verify the feasibility of fingerprint-based LoRa localization in practice. We have three key observations: 1) The temporal link behavior is much more dynamic. The main reason is the micro-environment change; 2) To obtain SNR gains of LoRa signals is an efficient way to enlarge the network coverage; 3) The localization accuracy by taking LoRa signals as the fingerprint is far from needed. It highly depends on system deployment and the granularity of site-survey.

**Our Insights.** We present a few key insights for the LoRa communication stack and localization method design in the future as follows:

- To deal with the link dynamics, SF 12 may not be resilient enough. We need a flexible way to extend SF to 13 or more, which is not supported on commercial-of-the-shelf LoRa radios, to avoid temporal disconnection.
- To deal with the link dynamics, the ad-hoc multiple-hop relay may be an alternative way to forward the data reliably. How to reduce the energy consumption for forwarders searching at a very low duty cycle and extra cost to maintain the network status is a critical problem.
- To obtain SNR gains in a LoRa gateway is an efficient way to enhance the coverage ability. Hence, how to detect and recover weak signals with less overhead is another important problem.
- The fingerprint-based LoRa localization suffers from the hard taming link behavior. More sophisticated techniques are needed to achieve accurate localization with narrow bandwidth and

low-cost end nodes.

**Measurement Universality and Deployment Diversity.** With the similar settings of the LoRa transceivers, our observations may be applicable to other typical urban areas with high-density obstacles and frequent micro-environment changes as shown in Figure 3. For new areas with great disparities (e.g., rural areas, forest areas, mountain areas) from our current urban environment, the results may vary since the link behavior is highly related to the types of different land covers along the link path. The gateway siting and deployment also affect final results. A higher antenna and fewer obstacles would result in higher PDR, higher ESP, and better coverage with more LOS links.

## CHAPTER 4

### **FACETOUGH: PRACTICAL FACE TOUCH DETECTION WITH A MULTIMODAL WEARABLE SYSTEM FOR EPIDEMIOLOGICAL SURVEILLANCE**

#### **4.1 Introduction**

Face touch is an unconscious and automatic behavior that most of us have. In general, the face-touch frequency is at a rate of 15-54 times per hour, and almost half of the face touches came in contact with mucous membranes [12–15]. The frequency of face-touch behaviors is uneven between the two hands for the human and primates, and most self-touches are with the non-dominant hand [13, 112–114]. During the COVID-19 pandemic, face-touch behaviors significantly expose humans to an epidemiological risk. Clinical studies have shown that the main route of transmission of infectious diseases (e.g., COVID-19, Ebola, influenza) is through droplets sneezed or coughed out by infected people and are then inhaled by someone else [16–18]. Those droplets can land on surfaces like tables or chairs and live for more than 72 hours. When we touch the surfaces, the epidemic viruses from polluted hands can be easily transferred [14, 19, 115] to your eyes, mouth, and nose, and then cause respiratory diseases.

Face touch can not be diminished by physical protection. Also, due to its unconscious nature, it is a hard-to-break habit that requires much conscious effort. To avoid the potentially serious consequences of face-touch behaviors and track people’s face-touch patterns, prior work analyzed the occurrence, and the frequency of face-touch to evaluate the progress of face-touch behavioral change during the pandemic [19, 116]. For example, a monitoring system can send haptic feedback to users when they touch the face. Such systems can alert people who work in sensitive domains like airports and hospitals not to touch their faces, which help stop the spread of epidemic virus [116]. Besides infectious diseases, many studies showed that face touch is associated with the user’s emotional status [117], mental health conditions [118], and workload evaluation [119]. However, the existing study still relies on clinicians or professionals to manually collect the occurrence and frequency of face touches, which introduces high labor costs and significant overhead for data analytics.

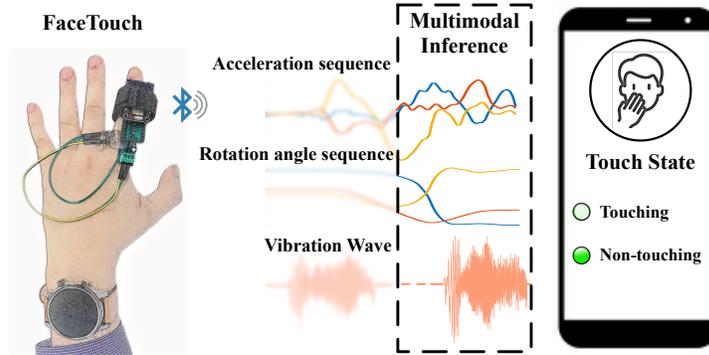


Figure 4.1 Illustration of our multimodal wearable system for practical face touch detection.

Wearable Awareness Enhancement Devices (AEDs) such as smartwatches, smart rings that support automatic behavior logging and facilitate behavioral changes (e.g., sleep trackers [120] and smoking detector [121]) can be used to address such problem. Mainstream automatic face-touch monitoring is currently performed by recognizing face-touching gestures. Prior work has investigated a variety of emerging sensing techniques and wireless signals, including acoustic [20–23, 35–38], radio frequency signals [24–26, 39], and magnetic signals [27, 28], to measure the distance between the hand and the face and recognize potential hand-to-face gestures. On-body sensors like inertial sensors also have been investigated to extract features from the movement of hands to classify face-touch gestures [30–34]. However, many similar gestures (e.g., picking up the phone, wearing a hat, or adjusting eyeglasses) can significantly degrade their system performance and generate lots of false alarms, causing unnecessary panic and/or bringing medical resources to a place where they are not needed. To filter out these false-positive gestures, a recent work leverages sensors in the ear to accurately detect touch events [29]. However, since it relies on always-on sensing and signal processing to guarantee high recall, the battery life is extremely limited (e.g., the system requires charging multiple times per day), increasing the user burdens and degrading the user experience. Therefore, there is a significant need for an accurate and low-power face-touch monitoring system. To fill the gaps, we propose FaceTouch (Figure 4.1), a novel wearable system consisting of a ring and a wristwatch using three sensing modalities: acceleration, rotation, and vibration. The wristwatch is equipped with an inertial sensor, and the ring contains a pair of small vibration trans-

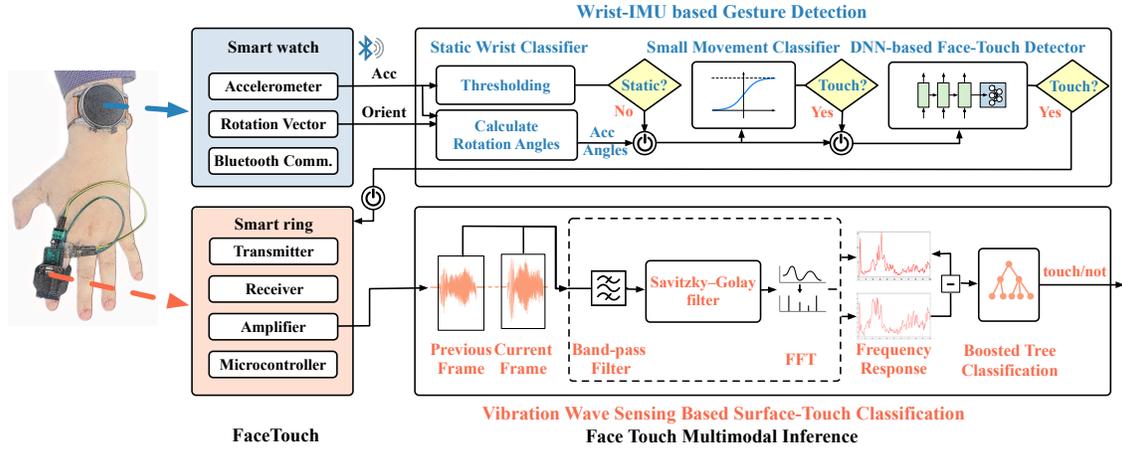


Figure 4.2 Overview of our multimodal wearable system for face-touch gesture detection.

ducers<sup>1</sup>. As shown in Figure 4.2, we divide the face touch detection into two sequential tasks. The first task is face-touch gesture detection which detects the movements when the hand moves toward the face, and the other is the surface-touch classification to determine whether the fingers touch the skin. We first leverage the inertial sensor on the wrist to detect whether an arm gesture is towards the user’s face (called *face-touch gesture*) through the pattern of wrist movement and rotation. The energy consumption of the MEMS-based inertial sensor is negligible, so we can run it in an always-on manner. Then, we design and implement a novel wearable ring to extract touch-related vibration features due to the changes in propagation paths when the finger touches different surfaces. By combining these sensing modalities together, FaceTouch can precisely distinguish face-touch behaviors from ambiguous arm gestures while minimizing energy consumption. By accurately capturing the evolution of users’ face-touch patterns over time in different activities, FaceTouch can provide users with records of when/where they touch their face to increase their awareness, trigger alerts when user touch their face at really high frequency or involve high-risk behaviors such as long-time face scratching. In the long-term point of view, FaceTouch can help with reducing frequent face-touch behaviors of the users.

To realize FaceTouch, we face three technical challenges: 1) Face touch detection requires always-on sensing and real-time signal processing that can drain the battery quickly. Since both vibration sensing and gesture recognition contain power-hungry components, it is challenging to

<sup>1</sup>The vibration transducers can send haptic feedback to users when they touch faces

design a face touch detection algorithm that achieves high recall and precision while minimizing power consumption. 2) Existing vibration sensing requires two separate devices (i.e., one on the moving body part and the other on the body part being touched) to sense vibration waves from the transmitter and the receiver, which is impractical and leads to a high bar for many users. However, simply placing the transmitter and the receiver close together on a single device results in saturation of the receiver device and poor sensitivity to the touch event. 3) Considering the limited computation resource of a wristwatch and the importance of user experience, our face-touch gesture detection methods must be computing-efficient to optimize energy consumption while keeping high precision and recall for diverse users without exhausting user training or cooperation.

We design three key components in FaceTouch to address the challenges. First, we design FaceTouch as an end-to-end system that adopts a cascading classification model to ensure high detection precision/recall while minimizing power consumption. The cascading model not only effectively fuses results from complementary sensor modalities, including IMU and vibration sensing, but also only triggers the energy-consuming component when it is essential, such that it balances the trade-off between performance and computation/energy overhead to achieve a practical system. Specifically, FaceTouch leverages two lightweight classifiers to lower the duty cycle of DNN-based face-touch gesture detection and the vibration-based surface-touch classification. The first one is an acceleration-threshold-based classifier, which filters out the static wrist gestures when the wrist is at rest. The second classifier is a logistic regression model [122, 123] for detecting small movements like typing keyboard and page-turning. Second, we propose a novel vibration sensor that requires only a single point of instrumentation. To detect touch events and overcome the signal saturation problem, we inject a chirp vibration signal in the finger, extract unique features caused by materials' properties, and design a lightweight boosted tree classification model to precisely detect face-touch events across different users and surfaces. Third, we adopt computing-efficient GRU unit [124] instead of LSTM [30] in DNN-based IMU arm gesture recognition for sequence processing. For training/fine-tuning a DNN model for a specific group of people (e.g., medical staff, farmer, etc.) with only a little new user-conscious input, we adopt the consistent-regularization-based semi-

supervised learning [125] to exploit deep model training with a small portion of labeled data and a large amount of unlabeled data to achieve compatible performance with supervised learning. This enables a quick start of our system in the initial phase of real-life usage. The system can be continuously boosted when we collect user-specific data. The subsequently collected data can be used to train via semi-supervised learning without requiring the users to provide labels.

We implement FaceTouch prototype using off-the-shelf hardware components and commercial wearable devices. The ring prototype is low-cost (< \$80) and low-power ( $60.89 \mu\text{W}$ ). FaceTouch utilizes the sensing data from the always-on inertial sensor on the wristwatch and vibration features from the ring to precisely monitor face-touch behaviors. We conduct a user study with 10 participants. The participants touch their faces and various surfaces (e.g., glass, cloth, rubber, wood) with daily activities (sitting, standing, walking) and false-positive behaviors (drinking, calling, adjusting glasses, etc.). Experimental results show that FaceTouch achieves 93.5% F1-score for face-touch detection with leave-one-user-out cross-validation, which is 9% higher than the state-of-the-art method [29]. The F1-score is close to that of the personalized model (93.9%), which demonstrates the generality of FaceTouch across diverse users.

The contributions of this work are:

- We propose FaceTouch, a multi-modal wearable system capable of detecting face touch in practical scenarios. To the best of our knowledge, FaceTouch is the first system that can achieve long-term (e.g., 79-day to 273-day usage without charging) and accurate face touch monitoring. The system can be used to prevent the potentially serious consequences of face-touch behaviors.
- We design an effective cascading model using always-on inertial sensors to lower the duty cycle of power-hungry components while maintaining high recall. Unlike prior work that only detects face-touch gestures, FaceTouch can distinguish face-touch from various false-positive behaviors while the user performs daily activities.
- We design a novel vibration-based sensing unit to extract unique vibration features while the user touches the face. It is the first vibration sensor that requires only a single point of instrumentation, containing both the transmitter and the receiver. As a result, it can sense the reflected vibration

signal from the fingertips and classify surface materials.

- We implement the FaceTouch prototype using off-the-shelf hardware components and commercial wearable devices. We evaluated the prototype and validated its performance with 10 participants. Overall, FaceTouch achieves an average of 93.5% F-1 score. The average power is only 60.89  $\mu\text{W}$  in normal daily usage and 209.15  $\mu\text{W}$  in extremely heavy usage.

## 4.2 System Overview

In this paper, we design a novel wearable system on the wrist and the finger using three sensing modalities: *accelerations, rotations, and vibration waves*. The key idea to minimize energy consumption is to use low-power sensors and lightweight signal processing to filter out irrelevant hand movements. Face-touch gestures and their similar gestures (e.g., drinking) involve large angular variations at the elbow and small angular variations at the shoulder [30]. In the long term, the frequency of such gestures is extremely low (e.g., less than 30 times per hour) for regular users in working environments (e.g., college students attending a class) [126]. Therefore, we design a cascading classifier to turn each component on step by step so that the energy-consuming components remain asleep for most of the time without missing any key movements. FaceTouch leverages low-power and ubiquitous IMU sensors in existing wearable devices to detect gestures that approach the face. We then design a novel vibration sensor on a smart ring to distinguish the face touch from false-positive behaviors that involve different touching surfaces. Specifically, we inject vibration chirps (1–10 kHz) into the finger and monitor the vibration that propagates through the finger to classify surface materials. Figure 4.2 illustrates the overview of FaceTouch, containing four individual classifiers:

- **Static wrist classifier:** The first classifier leverages an always-on accelerometer and a simple threshold-based approach to classify whether the wrist is static.

- **Small movement classifier:** If the system detects a hand movement, it turns on the gyroscope. Then, both accelerometer and gyroscope data will be fed into a logistic regression classifier to determine if hand movement is small or significant.

- **DNN-based face-touch detector:** The temporal sequences of accelerometer and gyro data will

Table 4.1 Example of scenarios that trigger the cascading classifiers at different steps. ✓ indicates a possible face touch event and will trigger the next classifier for further validation. ✗ indicates the next classifier will not be triggered due to irrelevance between the current event and a face-touch event. FaceTouch detects a face touch only if the results of all classifiers are ✓.

Scenario		Cascading Steps			
		Static Wrist Classifier	Small Movement Classifier	DNN-based Face-touch Detector	Vibration-based Surface Classifier
Static		✗	✗	✗	✗
Small Movements(typing, reading, ...)		✓	✗	✗	✗
Large Movements	Confounding Gestures (drinking water, smoking,...)	✓	✓	✓	✗
	Face Touch	✓	✓	✓	✓
	Other Large Movements	✓	✓	✗	✗

be fed into a recurrent neural network (RNN) based deep neural network to classify whether the hand is approaching the face.

- **Surface-touch classifier:** If the system detects a face-touch gesture, it turns on the vibration sensor. The signal captured by the sensor is then processed with a band-pass filter (1–10 kHz), a Savitzky–Golay filter, and a Fast Fourier Transformation (FFT) to extract frequency responses, which are fed into a series of boosted trees to classify the type of touching the surface. The system detects a face touch only if the movement involves both a face-touch gesture and a skin-touch event.

Table 4.1 demonstrates how FaceTouch filters various non-face-touch events out while guarantee high precision and recall. For example, all four classifiers will be triggered for confounding gestures that are similar to face-touch gestures. However, these confounding gestures do not involve a skin-touch event and will be filtered out by the vibration-based surface classifier. For other large movements that do not involve face-touch gestures, FaceTouch reduces energy consumption by remaining the vibration-based classifier asleep. It detects a face touch only if all classifiers are triggered, and the vibration-based classifier returns true.

### 4.3 Vibration-based Surface Touching Classification

In this section, we first present how touching various surfaces affects vibration waves, followed by how we measure the wave changes using a novel vibration sensor and extract unique features associated with surface-touch events. The vibration sensing serves as a skin-touch detector. FaceTouch leverages it to filter out false-positive gestures (e.g., drinking, calling, etc.) because the user does not touch a skin surface while performing these gestures.

**Vibration Through Human Skin** The propagation of vibration appears differently through media like gases, liquids, and solids. By analyzing the propagation of vibration along the touch surfaces we can estimate various material properties, including stiffness, damping ratio, elastic constants, and viscoelasticity. And these material properties can be further used to analyze the material characterization of composite materials like the human skin [127]. Prior work has shown the finger is a good conductor for vibration propagation [128], and demonstrated the capability of leveraging vibration through the finger to transmit data to other surfaces [128], recognize hand gestures [129], localize finger taps on unmodified surfaces [130], and user authentication [131]. However, all efforts distribute the transmitter and the receiver on different surfaces or parts of the body. Therefore, they all require two separate devices on the body (e.g., one on the finger and the other on the face), which is not user-friendly and practical. In this paper, we aim to design a novel vibration sensor using only a single point of instrumentation on the finger for surface touch detection.

#### **4.3.1 Vibration-based Surface-touch Detection**

We exploit the spectral properties of reflected vibration changes to classify touch surfaces. Compared with no-touch events, touching different materials has a significant impact on how the touch surface reflects the vibrations. As shown in Figure 4.3, we place both the transmitter and the receiver on the index proximal. The vibration can propagate from the transmitter to the receiver via two major paths. One is the direct path from the transmitter to the receiver (marked as the green line), and the vibration amplitude remains the same for touch and no-touch events. The other is the indirect path from the fingertip to the receiver (marked as blue, purple, and black lines). The amplitude of vibration via indirect path depends on the material's properties (e.g., mass, density, and spring constant) and vibration frequency [128, 131]. For example, when the vibrations propagate from the finger to a less dense medium (e.g., water), the phase of the reflected wave will be inverted, and the amplitude of the reflected waves will be smaller than that of passing from the finger to a dense medium (e.g., metal). Therefore, by monitoring the amplitude of reflected vibration, our system can detect whether the finger touches a surface and classify the material.

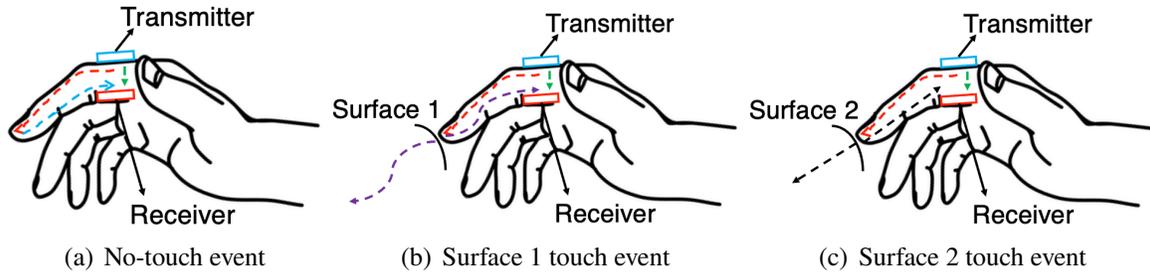


Figure 4.3 The rationale of proposed surface touch detection using vibration sensing.

We validate the sensing rationale using two Piezo-based transducers [132]. In this experiment, we place one sensor on the top of the index proximal as the transmitter and the other at the bottom of the index proximal as the receiver. The transmitter leverages an LM386 amplifier to inject the vibration at 2.5 kHz. The captured signal is amplified using an LMV358 amplifier and sampled by a 12-bit ADC at 100 ksps. After applying a bandpass filter, we perform 2048-points FFT with a sliding window to continuously extract the frequency response at 2.5 kHz. With this setup, we asked a user to perform a pointing gesture and touch five different surfaces (skin, glass, wood, cloth, and rubber).

The red error bars in Figure 4.4 show the mean frequency response at 2.5 kHz with 90% confidence intervals. We have two observations. First, since the frequency response of surface-touch events is significantly higher than that of no-touch events, the frequency response change can be used to detect surface-touch events. Second, we observe a negative correlation between the frequency response and material hardness. Soft materials absorb more energy from mechanical waves, which decreases the corresponding frequency response [133]. Thus, we can leverage the observation to classify the surface material. We also repeated the experiment on ten users and observed similar results, which validated the feasibility of using the wearable ring to capture the reflected vibration change caused by various materials.

**Identification of Sensing Sites** We investigated various skin sites on the hand for surface-touch detection. As shown in Figure 4.5, we asked a user to wear our sensor on 14 phalanges of the five fingers and then touched the face ten times using corresponding fingertip (e.g., using the index finger to touch the face when the user wears the sensor on the three index phalanges.). Figure 4.6 shows

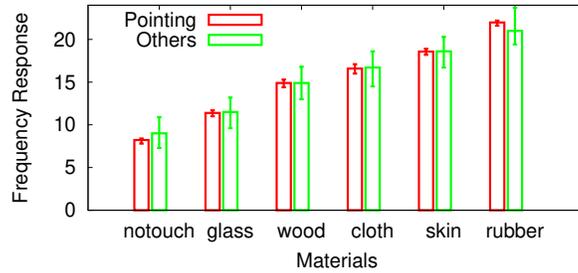


Figure 4.4 Frequency response at 2.5 kHz: the user touches five surfaces using either a fixed gesture or other gestures.

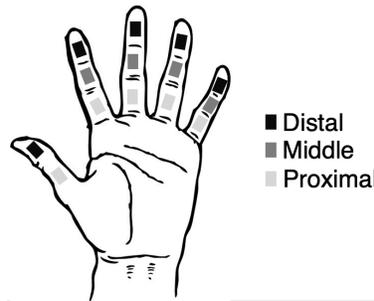


Figure 4.5 The 14 different measurement locations on the hand.

the mean frequency response at 2.5 kHz (with 90% confidence intervals as error bars) when the user wears the sensor on the 14 phalanges of the five fingers (x-axis). Overall, the frequency response increases when the sensor is closer to the fingertip. We observe the frequency response on the thumb is relatively low because the thumb has more dedicated muscle and a bigger nail, which can either block or absorb some vibration and decrease the frequency response [134]. Except for the thumb, the sensing granularity on other fingers is similar to each other. Since the ring is normally worn on the proximal and most face touch events involve index finger touch [135], we select the index proximal as the optimal sensing site. In this paper, we focus on optimizing the sensing sites on the finger due to the signal attenuation on the skin surface. To support long-distance sensing capability, the vibration generator may leak hearable sound to the environment, degrading the user experience. We carefully design a vibration sensor and fine-tune the duty cycle to minimize the hearable sound to the user. And we will discuss how to block the hearable sound to the environment in Sec. 4.7.

We also explore the capability of detecting surface touch events using other fingers while the sensor is placed on the index proximal. Figure 4.7 shows the comparison of the frequency response

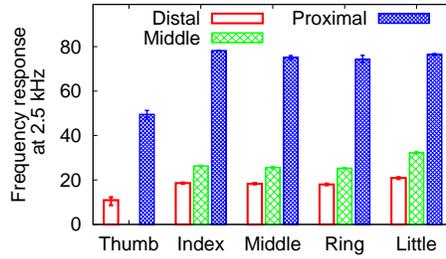


Figure 4.6 Comparison of the frequency response at 2.5 kHz on 14 skin sites.

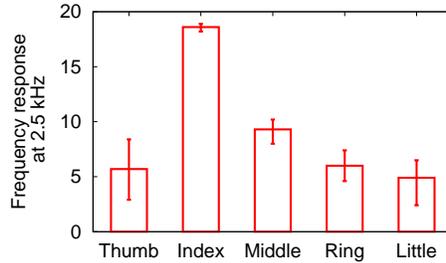


Figure 4.7 Comparison of the frequency response at 2.5 kHz at five fingers.

at 2.5 kHz when different fingers touch the face. We observe the frequency response drops when other fingers touch the face because the wave propagation distance increases. However, it is still possible to reliably detect the touch events when other fingers touch the face. We leave it as future work.

### 4.3.2 Practical surface-touch detection

Although we have shown the capability of detecting surface-touch events using a novel vibration sensor on the finger, we conducted all experiments when the user performs a fixed pointing gesture. However, hand gestures affect skin tension, which can generate similar features to face touch signals. The green bars in Figure 4.4 show the mean frequency response with 90% confidence intervals as error bars when the user performs various hand gestures and touches surfaces. We observed the error bars overlap among surfaces, which degraded the sensing granularity and robustness. Therefore, the frequency response on a narrow band (e.g., 2.5 kHz) can be noisy across diverse users.

To overcome the challenge, we inject vibration chirps (1–10 kHz<sup>2</sup>) on the finger to extract unique

<sup>2</sup>The minimum frequency of the chirp signal is 1 kHz, which is high enough that users can hardly feel the vibration [136].

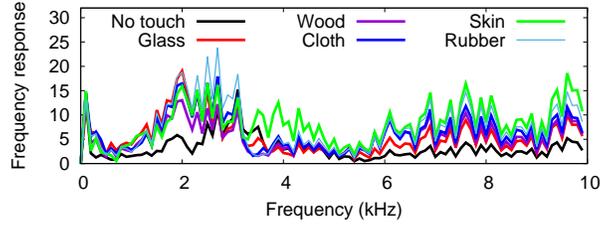


Figure 4.8 Frequency responses of the chirp signal (1–10 kHz) when touching on five materials.

features related to face touch events. Vibration chirps have been investigated to recognize finger gestures, and their frequency responses vary based on how vibration wave travels through the finger, namely tissue, muscle, blood, and bone [129]. Therefore, by scanning a wide variety of vibration frequencies, FaceTouch leverages the properties of different waves propagating through the finger to extract unique features for surface-touch classification. We conducted a user study with ten participants to validate the design, and participants touched five surfaces with various gestures. Figure 4.8 shows the mean frequency responses within 1–10 kHz. Overall, the frequency responses of no-touch events are significantly lower than those of touch events for most of the frequencies. The observation implies fewer vibration waves can be reflected from the fingertip during no-touch events. Besides, we observe that surface materials affect the power of frequency responses at different vibration frequencies. For example, the frequency responses when touching the skin are the highest among all touch events after 3.5 kHz. The frequency responses of the rubber touch is the highest within 2.5–3.5 kHz.

### 4.3.3 Touch event detection and touch surface classification

According to the observations above, our feature extraction is based on a comparison to a no-touch signal, which can be recorded when the user wears the sensor for the first time. Then, we compute the mean Euclidean distance of 192 frequency responses between the current chirp signal and the no-touch signal. To classify the surface materials, we extend the feature vector from 192 to 235 by adding the Mel-frequency cepstral coefficients (MFCC), zero-crossing rate, mean energy, and energy variation. The features are then fed into a supervised learning model to capture the relationship between the feature vector and surface materials. Specifically, we choose boosted tree classification [137] that optimizes a sequence of classification trees with weights associated with

decisions. As the number of trees grows, it can correct errors made by the previously trained tree. We also implement our touch event classification model using other classification backends (e.g., SVM and deep neural networks). However, these models do not outperform boosted trees classification because the feature dimensionality is relatively small in our system, and running a large number of iterations degrades the performance due to the overfitting problem. Another benefit of boosted tree classification is its low computing complexity because it only involves comparison and addition operations during real-time inference. When we deploy the model on embedded systems (e.g., STM32 or MSP432), the inference latency and energy consumption are significantly lower than that of SVM and deep neural networks. Specifically, the time complexity of boosted tree classification is less than 5% and 11% of that using a four-layer feed-forward neural network and SVMs, respectively. As a result, those models consume 98% and 93% more energy to classify surface materials. Overall, boosted tree classification achieves a better trade-off between efficiency and accuracy.

The vibration-based module can only detect skin touches and cannot distinguish face touch from other skin touches (e.g., hand touch, clenching fist). However, our cascading mechanism mentioned in Sec. 4.2 can effectively filter these confounding skin-touch events because these movements do not involve face-touch gestures.

#### **4.4 Wrist-IMU based Face-Touch Gesture Detection**

Based on the acceleration and rotation data obtained from a wrist-IMU, we design a sequential face-touch gesture detector with three classifiers to optimize energy efficiency and detection precision for long-term continuous usage. First, we design a threshold-based static wrist classifier that filters out the static periods in which no gesture is involved. For the rest of the wrist movements, we use a small-movement classifier to further filter out those movements which could not be a face-touch gesture with explicit feature definition. At last, we develop a DNN-based face-touch gesture classifier to accurately recognize face-touch gestures from other confounding gestures. In comparison with the first classifier, the computational overhead of the later classifiers is high but they will be triggered infrequently. Thus, we optimize the energy efficiency for continuous monitoring while

keeping a high precision and recall.

#### 4.4.1 IMU Data Input

We use the accelerometer and gyroscope of a 9-axis IMU equipped on a commercial off-the-shelf smartwatch to record the acceleration and rotation data of the wrist. Specifically, at any timestamp  $t$  under 50Hz sampling rate, we adopt Android API 28 to obtain the 3-axis linear acceleration  $a(t)$  and the orientation  $o(t)$  of the device in a Global Reference Frame (GRF) in the form of a quaternion which is a mathematical entity that provides convenience for representing three-dimension orientations and rotations of objects. The linear acceleration reflects the linear movements of the wrist while from the change of device orientation in GRF, we can calculate the rotation angles of the wrist in GRF [138]. As face-touch gestures involve a series of wrist movements and rotations, to determine whether a user touches the face at time  $t_i$ ,  $i \in \mathbb{N}$ , we look ahead for a short time interval  $\Delta t$  and take two sequences of the acceleration and rotation data samples  $S_a^i = \{a(t_i - \Delta t), \dots, a(t_i)\}$  and  $S_o^i = \{o(t_i - \Delta t), \dots, o(t_i)\}$  directly as the input of our face-touch gesture detector. Since we do not leverage accelerations and orientations for location tracking, the drifts of IMU sensors will not be accumulated over time, resulting in acceptable bounded errors.

#### 4.4.2 Always-on Static Wrist Classifier

According to the common daily life pattern, people’s wrists are almost static most of the time in a day due to sleeping, resting, etc. Face touch rarely happens during static periods. In addition, it would be a waste of energy to run all three classifiers to detect face-touch gestures during the static periods. Therefore, as the basic step, we filter out all static periods to improve the energy efficiency without losing detection accuracy by designing a computation-efficient static-wrist classifier.

**Method Design** Specifically, when a user’s wrist remains static, both the average and variance of linear acceleration values within the static period should be close to 0. With two non-negative pre-configured threshold vectors  $T_1 = [T_1^0, T_1^1, T_1^2]$  and  $T_2 = [T_2^0, T_2^1, T_2^2]$ , we compare the absolute average and variance of the 3-axis acceleration sequence  $\overline{a(t_i)} = \sum_{(a_j) \in S_a^i} (a_j) = [\overline{a(t_i)^0}, \overline{a(t_i)^1}, \overline{a(t_i)^2}]$  and  $\sigma(a(t_i)) = [\sigma(a(t_i)^0), \sigma(a(t_i)^1), \sigma(a(t_i)^2)]$  with  $T_1$  and  $T_2$ , separately. If both  $\overline{a(t_i)}$  and  $\sigma(a(t_i))$  are lower than  $T_1$  and  $T_2$  on all 3-axis, the status of the user’s wrist is recognized as static.

As such, the next-tier classifier would not be triggered and continuously check the next acceleration sequence.

#### 4.4.3 Small Movement Classifier

Among all possible arm gestures that human beings can perform, face-touch gestures are only a small part of them. Common face-touch gestures involve relatively large movements of the whole forearm and even the elbow. In this case, the wrist moves faster and we can expect that the wrist acceleration changes rapidly, and the wrist twists obviously. We further utilize the gap between small movements and face-touch gestures to improve computational efficiency by filtering out those gestures that only involve small wrist movements.

**Method Design** First, we calculate the rotation matrix  $R(t_i)$  from orientation quaternion  $o(t_i)$ , and then obtain the related 3-axis euler angles  $\theta(t_i) = [\theta(t_i)^0, \theta(t_i)^1, \theta(t_i)^2]$ . Then given a rotation sequence  $S_o^i$ , we can convert it to an rotation-angle sequence  $S_\theta^i = \{\theta(t_i - \Delta t), \dots, \theta(t_i)\}$ . We use  $\overline{\theta(t_i)}$  and  $\sigma(\theta(t_i))$  to indicate the average and variance of the rotation-angle sequence. Moreover, we mainly consider five statistical values of wrist movements:  $\overline{a(t_i)}$ ,  $\sigma(a(t_i))$ ,  $\overline{\theta(t_i)}$ ,  $\sigma(\theta(t_i))$ , and  $\theta(t_i) - \theta(t_i - \Delta t)$  that can reflect the displacement, speed, and rotation changing features of wrist movements. With these features, we train a logistic regression model [122, 123, 139] to recognize whether a wrist movement is small or not. If a small wrist movement is detected, we will switch to running the static-wrist classifier for continuous static wrist checking. Finally, the logistic regression model training makes sure that the face-touch gestures would not be classified into the category of small wrist movements.

#### 4.4.4 DNN-based Face-touch Detector

After the processing of the first two classifiers, most gestures that are not related to face-touch events are filtered out with low computation cost. To accurately detect face-touch gestures out of other wrist movements with the same magnitude, we need to extract more gesture-specific features from the raw data sequence with a more powerful model. As the gesture data consists of sequential accelerations and rotations, it contains temporal relation between data at different time steps where such relationship is unique across different gesture types. Therefore, we need a sequence prediction

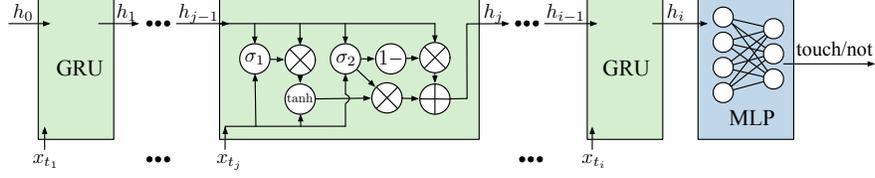


Figure 4.9 The DNN model for detecting face-touch gestures. The DNN consists of a single-layer GRU and a 2-layer MLP. The GRUs take the raw data sequences as input and the MLP outputs whether a face-touch gesture is detected.

Table 4.2 DNN-based Face-touch Detector configurations.

Input( $6 \times t_i$ sequence)
GRU-32
FC-16
ReLU
FC-2
soft-max
Output( $2 \times 1$ )

model to capture such temporal relations.

**Method Design** LSTM [30] is a widely used DNN unit to handle sequential data. Considering the computation efficiency, however, we adopt a GRU unit [124] that requires less computational cost while achieving a similar performance of LSTM for short sequence data. GRU-based DNN can learn the inherent relationship between sequential IMU data and face-touch/non-face-touch behaviors as long as the DNN can be trained with a dataset that covers the feature space as much as possible. To determine whether a user touches face at time  $t_i$ , we generate the network input data sequences with the acceleration sequence  $S_a^i$  and angle sequence  $S_\theta^i$  as follows:

$$\begin{aligned}
 X &= x_{t_1}, \dots, x_{t_j}, \dots, x_{t_i} \\
 x_{t_j} &= [a(t_j)^0, a(t_j)^1, a(t_j)^2, \theta(t_j)^0, \theta(t_j)^1, \theta(t_j)^2],
 \end{aligned} \tag{4.1}$$

where  $t_1$  is  $t_i - \Delta t$ ,  $t_j$  is a timestamp within range  $[t_i - \Delta t, t_i]$  when acceleration and orientation are sampled at 50Hz.

As in Figure. 4.9, our network architecture consists of GRU network noted as  $F_{\text{gru}}$  and Multiple Layer Perceptron (MLP) network noted as  $F_{\text{mlp}}$ . The GRU network accepts sequence input  $X$  and outputs a hidden feature  $h_i$  at the last time step  $t_i$  which is then taken by the MLP network as input. The MLP network outputs the prediction results. For training with labeled data, we adopt cross-

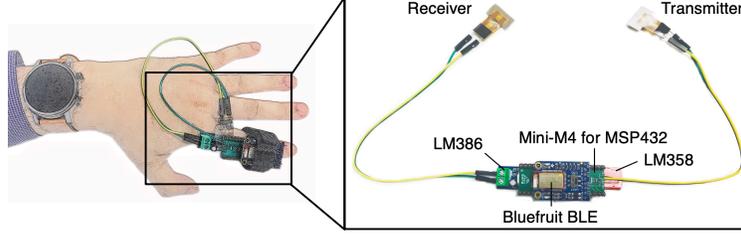


Figure 4.10 FaceTouch ring prototype.

entropy loss:

$$\begin{aligned} \mathcal{L}_l = & \mathbb{E}_{(X,y) \sim \mathbb{X}_l} (1 - y) \log (1 - F_{\text{mlp}}(F_{\text{gru}}(X))) \\ & + y \log F_{\text{mlp}}(F_{\text{gru}}(X)) \end{aligned} \quad (4.2)$$

where  $\mathbb{X}_l$  is the set of labeled training data. We adopt consistency regularization for the DNN model training with unlabeled data to make the prediction robust to small perturbations which can improve the performance of the face-touch detection model when labeled data is rare. In such case, the loss is defined as:

$$\mathcal{L}_u = \mathbb{E}_{X \sim \mathbb{X}_u, \delta \sim \mathcal{N}(0, \sigma_n)} |F_{\text{mlp}}(F_{\text{gru}}(X + \delta)) - F_{\text{mlp}}(F_{\text{gru}}(X))| \quad (4.3)$$

where  $\mathbb{X}_u$  is the set of unlabeled training data and  $\delta$  is a Gaussian noise  $\mathcal{N}(0, \sigma_n)$  to perturb the input, where  $\sigma_n$  is much smaller than the scale of  $X$ .

## 4.5 Implementation

The prototype consists of two main components, namely a novel vibration sensing unit and an IMU sensing unit. For vibration-based sensing, we designed and implemented a compact and low-cost (e.g., < \$80) wearable ring (Figure 4.10) using off-the-shelf hardware components. The sensing unit is powered by a small 400 mWh battery. For IMU sensing, we adopt a commercial off-the-shelf Moto 360 smartwatch.

**Vibration Sensing Unit** To minimize the form factor of the sensing unit, we utilized two Piezo transducers (PHUA2010), each of which is 20 mm by 10 mm in size. We built a ring-like device to attach both transducers. One is placed on top of the index proximal to inject vibration chirp (1–10 kHz) into the finger and the other is placed at the bottom to capture the surface waves from the finger. The transmitter sends one 50-ms chirp of 1.65 Vpp per second. We utilize an LM386 amplifier to

amplify the chirp signal. The frequency range of the chirp signal was optimized in Sec. 4.3.2 to retain maximum vibration information propagating in the finger. The receiver is connected to an LMV358 amplifier (amplification gain is 32) to amplify the vibration signals. The amplified signals are then sampled by a 12-bit ADC at 100 ksp/s. Both amplifiers have large quiescent current draws (e.g., 0.2–4 mA). To minimize the power consumption of the sensing unit, the positive supply voltages (VCC) of amplifiers are connected to an analog switch, and we switch the amplifiers off when the sensing unit is not triggered by the cascading model as mentioned in Sec. 4.2.

We use a MINI-M4 for MSP432 micro-controller<sup>3</sup> to digitize vibration signals from amplifiers, extract touch-related features, and the boosted tree is deployed on the micro-controller to detect touch events and classify the surface materials. When the microcontroller is not active, it only consumes 40  $\mu$ W power in idle mode. The microcontroller has a low-power MCU with 80  $\mu$ A/MHz in active mode and a 12-bit ADC with 450  $\mu$ A at 100 ksp/s. During signal digitization, we leverage the Direct memory access (DMA) in the microcontroller so that the main processor will be in the low-power mode during data sampling. We implement the touch event detection algorithm in Sec. 4.3.3 for binary skin-touch classification on the microcontroller. The number of boosted trees is 50, and the maximum tree depth is 4. The prediction results will be transmitted to the nearby devices via a Bluetooth transceiver [140]. We integrate a 400 mWh Lithium-ion battery underneath the MINI-M4 board to power the sensing unit.

**Wrist-IMU Sensing Unit** We implement the face-touch gesture detection models including Static Wrist Classifier, Small Movement Classifier, and DNN-based Face-touch Detector, and deploy these models on a smartwatch (Moto 360 3rd Gen). The IMU data collected by the smartwatch is sampled at 50Hz. Based on our real-world experiments, the average length of face-touch gestures is around 1.5 seconds. Thus, to capture all face-touch events while ensuring the energy efficiency of the system, we trigger the static wrist classifier every 500ms and take 1.5s data back from the triggered timestamp. The system will process the raw sensor data based on our wrist-based gesture detection in Sec. 4.4.

---

<sup>3</sup>To minimize the power consumption of the board, we removed all irrelevant components on the MINI-M4 FOR MSP432 board, including the USB bridge and LED indicators.

We empirically selected the thresholds of the static classifier by calculating the average and variance of linear acceleration values of multiple collected static wrist data records and data with wrist movements. The thresholds are:  $T_1 = [0.1, 0.1, 0.1]$ ,  $T_2 = [0.5, 0.5, 0.5]$ . Since the differences between static/non-static data are significant, our static classifier achieves 100% precision and recall in filtering out static data. The small movement classifier is fitted from historical data. We conduct model selection on the small movement classifier by tuning the precision of the model to be as higher as possible while keeping 100% recall for better final task performance. Table 4.2 shows the configuration of DNN-based Face-touch Detector in Sec. 4.4.4. We adopt a single-layer GRU with a hidden dimension of 32 and a 2-layer MLP with two fully connected layers of size 16 and size 2. The DNN model is trained on data described in section 4.6.

## 4.6 Evaluation

In this section, we first discuss the experimental setup and the process of data collection, followed by FaceTouch’s overall performance and practical considerations.

### 4.6.1 Experiment Setup and Data Collection

In our study, we recruited 10 participants (8 males and 2 females with ages ranging from 20 to 30+). For each participant, we collected both face-touch and non-touch data throughout various daily activities (sitting, standing, and walking), touching surfaces (e.g., glass, wood, rubber), confounding gestures involving the hand moving toward the head direction but not touching the face (e.g., drinking, picking up the phone), and face-touch areas (e.g., jaw, forehead). By covering as many scenarios as possible, we obtained a comprehensive dataset that includes various combinations of movement replacement/speed/angle sequences when users touch face/not touch face. During the data collection process, the participants wore FaceTouch on their wrists and index fingers. They were asked to perform gestures naturally as in daily life during the data collection. Though they were wearing the prototype, their arm movements were not restricted, and the influence of the prototype on their behaviors was minimized.

The two devices were clock-synchronized by network time. The inertial data consists of 3-axis accelerometer data, orientation in quaternion form rotation vector acquired at 50Hz from Android

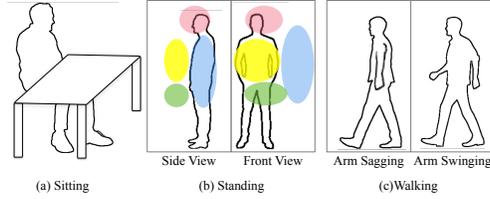


Figure 4.11 The illustration of the three scenarios in our dataset: (a) Sitting, (b) Standing: the four colored spaces represent regions from which face-touch movements may start, and (c) Walking.

Table 4.3 Data composition of participants in our dataset.

#Gestures Type	Activity	Sitting	Standing	Walking	
				Swing	No Swing
Face-touch		$70 \times 10$	$70 \times 10$	$70 \times 10$	$20 \times 10$
Non-touch		$70 \times 10$	$70 \times 10$	$20 \times 10$	$20 \times 10$
Confounding Gestures		$100 \times 10$			

API 28. The vibration data consists of 235 features described in Sec. 4.3.3. For data annotation, participants were accompanied by an instructor who timely logged the start and end time of touching face events while recorded by an RGB camera which was also clock synchronized. During the data segmentation and labeling process, the annotator integrated the manually logged timestamps of events with camera videos to get accurate labels. To collect comprehensive face-touch and non-touch gestures, participants performed various arm and hand gestures in three typical daily activities as in Figure 4.11. We collected about three-hour sensing data from each participant. Table 4.3 shows the data composition of each participant.

For face-touch behaviors, participants moved their hands from various starting points, which are randomly and evenly sampled, and then touched one of the five areas on the face shown in figure 4.12. For non-touch behaviors, participants performed various daily activities, including static, typing, reading books, fetching objects, lifting, putting things down, exercising, and walking with swing arms or non-swinging arms. We also collected many confounding gestures, including drinking water, eating, brushing teeth, adjusting the glasses/masks/hair, picking up the phone, and raising hands. These gestures inevitably confuse many existing systems using near-field sensing.

## 4.6.2 Overall Performance

**Precision, Recall and F-1 Score** We evaluate FaceTouch using leave-one-user-out cross-validation

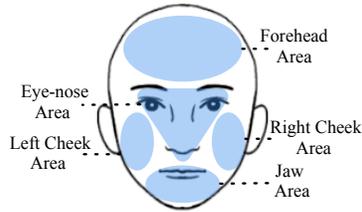


Figure 4.12 Five face areas covered in our experiments.

Table 4.4 Comparison of experimental settings.

	#Participants	Session Duration (per user)	#Activity Scenarios	Confounding Gestures	Cross-user model
FaceTouch	10	3 hrs	3	various types	✓
FaceSense	14	15 mins	1	Only facial	✓

which takes data of 9 participants as training data and data of the remaining participant as testing data. We compare our results with three baselines. The first baseline is FaceTouch without the vibration sensing module. The second method is the personalized model, in which the data of the same participant is split into the training set and testing set by 8:2. And we also compare FaceTouch’s performance with FaceSense (generic model) [29], a state-of-the-art face touch detection system using a dedicated earbud. Table 4.4 shows the comparison of experimental settings. Both studies have a similar user group. we evaluate FaceTouch via a long-term session (3 hours), while FaceSense was evaluated for only 15 minutes per user. FaceTouch’s dataset contains three activity scenarios (sitting, standing, and walking) and various types of confounding gestures (introduced in section 4.6.1) that are similar to face touch. For the FaceSense study, all data are collected when the participants are sitting, and only facial movements that are confounded with face touch are considered. Both studies leverage the cross-user model to evaluate their systems.

Figure 4.13 shows the mean precision, recall, and F-1 score of ’s generic model, as well as the three baselines. The main metric we adopt for comparison is the F-1 score. Overall, FaceTouch achieves 93.5% F-1 score. Without our vibration sensing module, the F-1 score drops 4%. The improvement mainly comes from reducing false positives since the vibration sensor can effectively filter out non-skin touch events like drinking, calling phones, etc. Also, the F-1 score of FaceTouch is only 1% lower than that of the personalized model, which demonstrates the generality of the

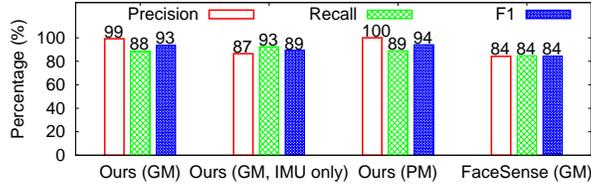


Figure 4.13 FaceTouch’s overall performance.

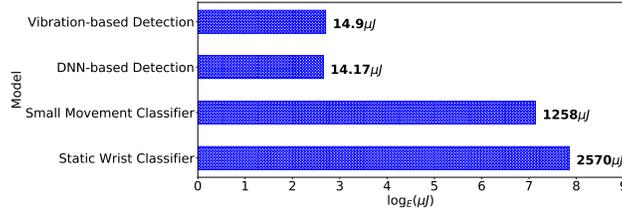


Figure 4.14 Energy consumption of each step in FaceTouch’s cascading classification.

system. Finally, FaceTouch’s generic model outperforms the FaceSense by 9% on the F-1 score.

**Energy Consumption and Inference latency** We measure the energy consumption of FaceTouch using a Monsoon High Voltage Power Monitor [141]. The energy consumption of each step in FaceTouch’s cascading classification is shown in Figure 4.14. When the wrist is static, FaceTouch only consumes  $14.9 \mu\text{J}$  every 500 ms to monitor the accelerometer data. Then, depending on the wrist movement, different IMU-based classifiers will be triggered step by step, and the total energy consumption to produce a face-touch inference is  $1287.07 \mu\text{J}$ . We also measure the energy consumption of the vibration sensing unit, which is shown in Table 4.5. Overall, each face-touch inference consumes about  $2570 \mu\text{J}$  with signal digitization ( $750 \mu\text{J}$ ), injecting vibration chirp signals ( $330 \mu\text{J}$ ), and Bluetooth communication ( $1050 \mu\text{J}$ ) as the main contributors. When the cascading IMU-based classifier does not trigger the vibration-sensing module, the main processor will remain in low-power mode, and the vibration-sensing unit will be turned off. The power consumption in this mode is around  $40 \mu\text{W}$ .

Table 4.5 Energy consumption for producing one face-touch inference using our vibration sensing unit.

	Vibration-sensing Unit		Microcontroller		Bluetooth
	Transmitter	Receiver	ADC	Surface-touch classification	
<b>Energy</b>	$330\mu\text{J} (\pm 2)$	$200\mu\text{J} (\pm 2)$	$750\mu\text{J} (\pm 5)$	$<240\mu\text{J} (\pm 1)$	$1050\mu\text{J} (\pm 20)$

The inference latency of FaceTouch depends on the type of gestures. Overall, the inference latency for static wrist filtering, small movement filtering, face-touch gesture detection, and surface classification is 0.049 ms, 0.046 ms, 3.01 ms, and 25 ms, respectively. In the worst case, when all the classifiers are triggered, the total inference latency is 28.105 ms, which is still significantly less than the face touch detection interval (500 ms).

**Practicality** To evaluate the practicality of using FaceTouch in the wild, we conducted a two-hour user study in four typical scenarios. In scenario  $S_1$ , the participant frequently touched the face or performed confounding gestures and remained in the static state (e.g., the wrist is static) for the rest of the time. In scenario  $S_2$ , the participant was in the active state (e.g., the wrist moves frequently) and frequently touched the face or performed confounding gestures. In scenario  $S_3$ , the participant conducted various activities (e.g., drinking, eating, etc.) involving many confounding gestures. In scenario  $S_4$ , the participant was in the active state but touched his/her face or performed confounding gestures at normal frequency. Table 4.6 summarizes the frequency of face touch and confounding gestures in the study, as well as the corresponding power consumption. We have two observations. First, for normal scenarios (e.g.,  $S_4$  where the participant performs various non-face-touch gestures and occasionally touches the face), the total power consumption is below  $61 \mu\text{W}$ , which supports a run time of 273 days using a 400 mWh Lithium-ion battery. In this case, the vibration sensing unit was rarely triggered since most irrelevant gestures were filtered by the IMU sensing module. Second,  $S_1$ ,  $S_2$ , and  $S_3$  involve many confounding gestures which trigger all four classifiers frequently. However, in these extreme scenarios, the total power consumption is still below  $210 \mu\text{W}$ , which supports a run time of 79 days using a 400 mWh Lithium-ion battery. Scenarios  $S_1$  and  $S_2$  involve touch events at least once per minute and confounding gestures at least three times per minute, and  $S_3$  assumes an even higher frequency of confounding gestures (e.g., 25+ per min). These scenarios either hardly happen in real life or won't last for hours. In general, the real-life running time of FaceTouch using a 400 mWh Lithium-ion battery should be between 79 and 273 days depending on the frequency of face touch behaviors and confounding gestures. Therefore, the long-term study demonstrates the practicality of FaceTouch in real use cases. And

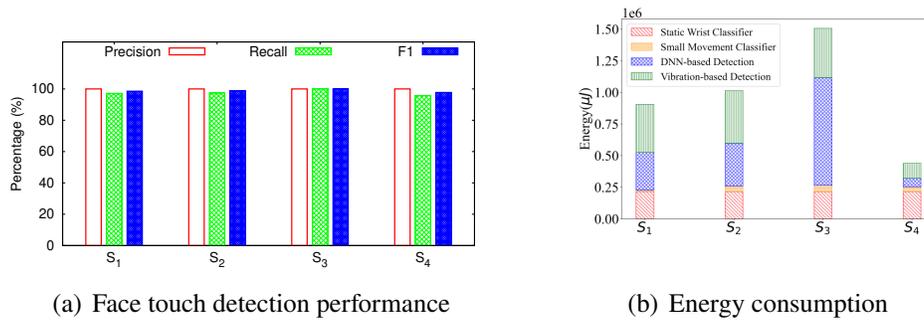


Figure 4.15 The overall performance and the energy consumption of FaceTouch in four daily scenarios.

Table 4.6 Details of long-term experiments in four different scenarios and resulting average power.

Scenario	Frequency of face Touch (/h)	Frequency of Confounding Gestures (/h)	Average Power( $\mu\text{W}$ )	
			IMU Sensing	Vibration Sensing
1	71	279	72.84	52.83
2	76	297	82.91	57.83
3	33	1645	154.89	54.26
4	24	15	44.47	16.42

the battery life is several magnitudes longer than existing solutions [28, 29, 142]. Figure 4.15(a) shows the overall performance of the long-term experiments using FaceTouch in four daily scenarios, FaceTouch achieves at least 97% F-1 score, demonstrating the robustness of FaceTouch in realistic scenarios. Figure 4.15(b) shows the corresponding composition of energy consumption, the energy consumption varies among the four scenarios. The always-on Static Wrist Classifier consumes the same amount of energy in all the scenarios and most of the energy is consumed by the IMU sensor. The energy consumption of the Small movement Classifier is negligible due to its lightweight computation overhead. However, DNN and vibration sensing consume the most energy in FaceTouch. Depending on the occurrence of confounding gestures and touch events, the DNN-based Face-touch Gesture Detector consumes 20–60% of the total energy consumption. Interestingly, since the IMU-based sensing filtered out many irrelevant gestures, the vibration sensing unit, which is the most power-hungry component in FaceTouch, is less triggered and consumes less than 25% of total energy in all scenarios.

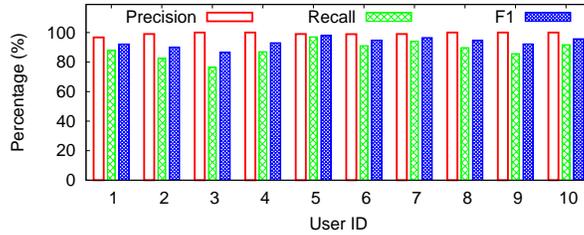


Figure 4.16 Impact of user diversity among 10 users.

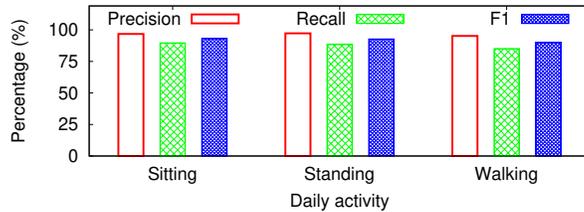


Figure 4.17 Impact of user activity.

### 4.6.3 Practical Considerations

We then examine the impact of user diversity, activities, confounding gestures, surface materials, and training the DNN-based face-touch detector with unlabeled data.

**User Diversity** We first look at the impact of user diversity. Figure 4.16 compares the precision, recall, and F-1 score across 10 participants using leave-one-user-out cross-validation. We observe the F-1 score remains high (87–98%) among the 10 participants, which validates the multi-modal sensing system significantly reduces false positives.

**Activity** Figure 4.17 shows the precision, recall, and F-1 score across three types of daily activities. The performance in the standing scenario is the best, followed by that of the sitting and walking scenarios. For standing scenarios, users often touch the face from the body side. Thus, the movement of the face-touch gesture tends to be more significant than that when the user is sitting, resulting in high detection accuracy. For walking scenarios, when the upper arm moves fast, the movement of the lower arm can be ambiguous, resulting in erroneous features.

**Confounding Gestures** Confounding gestures like drinking and calling can degrade the system’s accuracy and robustness. Figure 4.18 shows the impact of five false positive behaviors. The recall of all these behaviors is almost 100%. Since none of the false positive behaviors involves skin touch events, these behaviors can be effectively filtered out by our vibration-sensing module.

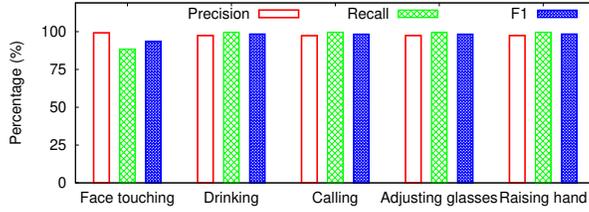


Figure 4.18 Impact of confounding gestures.

Labels	Cloth	Glass	Notouch	Rubber	Skin	Wood
Cloth	564	6		6	6	18
Glass	6	582				12
Notouch			600			
Rubber				594	6	
Skin	18				582	
Wood	12	12			6	570
	Predictions					

Figure 4.19 Confusion matrix: surface material classification.

**Surface Materials Classification** In this experiment, each participant touched five surfaces (cloth, glass, rubber, skin, and wood) ten times in random order. Figure 4.19 shows the confusion matrix across the five surface materials using leave-one-user-out cross-validation. We observe Face-Touch can precisely classify surface materials, which further distinguishes confounding gestures that may be mistaken as face-touch events by the IMU-sensing module. Note that both face-touch gesture detection and surface material detection are indispensable. The former relies on the latter to reduce false positives, the latter relies on the former to filter out irrelevant gestures that touch the skin of other body parts (e.g., leg, arm) but don't approach the face. For example, in the cases of drinking water or calling, FaceTouch may first recognize the touch event and the approaching face gesture. However, since the user does not touch the skin, the system can successfully filter out the false positive behavior by classifying the touching material.

**Training DNN-based detector with Unlabeled Data** In Section 4.4.4, we use unlabeled data for face-touch detection DNN model training through the loss in Equation (4.3). We conduct experiments with the  $\sigma_n = 0.01$  for the Gaussian noise. For the generic model, we use 80% of all the data as training data with a random split of 10% of the training data as labeled, and the rest are unlabeled data. We compare the DNN performance of training with/without semi-supervised

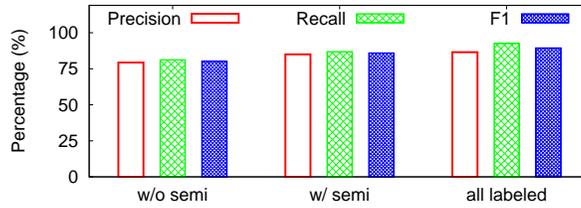


Figure 4.20 Training DNN-based detector with/without semi-supervised learning. Only 10% of the training data are labeled while the rest 90% are unlabeled.

learning in Figure 4.20. The DNN model trained with semi-supervised learning achieves 85% precision, 86.8% recall, and 85.9% F-1 score, which outperforms the DNN model trained without semi-supervised learning (79.3% precision, 81.2% recall, and 80.2% F-1 score) 5% on F-1 score and is close to our best F-1 score (89.4%) achieved on 100% labeled data. The results demonstrate that semi-supervised learning enables utilizing large-scale unlabeled data collected from different users in real-world life to improve the deep face-touch gesture detection performance.

#### 4.7 Discussion and Future Work

**Face Touch using Different Fingers or Hands:** In our experiments, the participants wore the ring on the index finger and the face-touch events involved the index finger touching the face by default which is compatible with most cases in daily life. We explore the capability of detecting surface touch events using other fingers with the sensor placed on the index proximal and observe the frequency response drops when other fingers touch the face because the wave propagation distance increases. Thus, in our current implementation, if the users touch their faces without using the index finger, the precision of face touch detection decreases as the features from the vibration sensing module can be erroneous. To solve this problem, one solution is moving the vibration sensor from finger to wrist. However, the attenuation of the reflected vibration signal increases as the propagation path increases. Once the SNR is below a certain threshold, the system cannot reliably extract features related to face-touch events. We can leverage a powerful vibration sensor to increase the sensing range, but such a sensor may degrade the user experience. We leave it as future work.

**Miniaturization:** In our current implementation, the IMU and vibration sensors are placed

separately on a smartwatch and a ring prototype. The ring prototype is built with off-the-shelf components that can be easily purchased from the market. There are more miniature components that can be used to reduce the size of the prototype. Besides, there are already many commodity smart rings in the market (i.e., Oura Ring). It's practically feasible to integrate vibration units into such ring systems so that the prototype can be further miniaturized. The utilization of smartwatches [143, 144] and smart rings in improving people's life quality has been well studied (e.g., sleep tracking [145], heart rate monitoring [146], menstrual cycle tracking [147]), so wearing such wrist-ring system only introduces acceptable overhead in our experiments and users' daily life.

It is possible to implement a single-piece sensing system (e.g., a wearable ring) containing both IMU-based and vibration-based sensing modules. However, the main challenge is to deploy our DNN model on a resource-constrained microcontroller, which has limited computation resources and memory space. We can also offload the sensing data to a remote server (e.g., a smartphone), but the data communication can be extremely energy-consuming. Prior work has investigated various ways to compress the DNN model that reduces both storage and computation requirements [148]. Therefore, we will investigate these approaches to compress our DNN model without sacrificing performance and then combine the wrist-ring system into a single device. We leave it as future work.

**Noticeable Vibration Sound:** The skin vibration generator operates on human audible frequency. In our experiments, once the transceiver is triggered, it will keep on for a while. Thus, the vibration sound may be a little bit annoying. We have conducted a small-scale user study among our 10 experiment subjects about the acceptance of the vibration sound and the average acceptance score is 4.2 out of 5, the score variance is 0.75, and the lowest score is 3. Large-scale user study remains a future work. We intend to address the vibration sound issue in two folds to make it negligible to the users. First, we will design a soft cover to absorb the sound propagating to the air while maintaining the vibration signals on the skin. If the soft material has similar acoustic properties to the skin, it will maximize the signal strength to the sensor [130]. For this purpose, we consider using inert silicone material to build the soft cover in our future work. Second, we can further reduce

the duty cycle of the vibration signal to turn the vibration sound down. However, a low duty cycle can cause miss-detection for short-time face-touch behaviors. Thus, we will explore an adaptive duty cycle scheme to optimize the user experience and sensing performance.

**ML Classifier Generalization:** Due to the user diversity, especially the length of arms, when we adopt our pre-trained IMU-based classifiers to new users, the performance may be subject to the size of the training dataset. If the training dataset covers the face-touch arm gesture data performed by similar body-condition users, the performance will be promising. To avoid performance degradation, we will also involve a bootstrapping process to calibrate the parameters of the IMU-based classifiers. Specifically, before a new user uses our system, we will ask the user to perform several pre-configured arm and hand gestures to fine-tune our classifiers. When it comes to larger user groups in which the user diversity is higher, and there are less data (maybe only the bootstrapping data) for unseen users, the semi-supervised learning that we explored in the experiment can provide us with a quick start for training a preliminary model of fair performance. Moreover, we will pursue a data-argumentation method to generate some synthesis training data covering more diverse users to enhance the generality of our IMU-based classifier.

#### **4.8 Related Work**

**Near-field Sensing Methods** Near-field sensing systems leverage wireless signals, including acoustic, Wi-Fi, Bluetooth, or magnetic signals, to extract unique features from the movement of hand gestures. Acoustic-based systems utilize microphones to capture reflected ultrasound signals and recognize hand/arm gestures [21–23, 35–38]. Recent work has turned existing smartphones and earsets into a sonar sensing system to detect face touching in the presence of user mobility [20]. Radio frequency-based methods measure the reflection of electromagnetic waves from the human body to recognize hand gestures [24–26, 39]. However, the main issue of these near-field sensing systems is that they cannot distinguish many false-positive gestures (e.g., calling, drinking) from face touch since these gestures have similar hand trajectories with face-touching gestures. Our system leverages a novel vibration sensor on a ring to filter out false-positive gestures. The performance of our system which is reported with confounding gestures present in the testing dataset

outperforms the most recent work [20].

**On-body Sensing Methods** On-body sensing systems utilize off-the-shelf or customized wearable devices (e.g., wristbands, earbuds) and various sensing techniques (e.g., ECG, EOG) to detect face touching. No Face-touch [30], FaceOff [31], Nudge [32], Immutouch [33], and Face Touch Aware [34] developed mobile apps on off-the-shelf smartwatches or smartbands to detect face touching using accelerometers. FaceSense designed a customized earbud with impedance sensing and thermal sensing [29]. ElectroRing [142], SkinTrack [149], and ActiTouch [150] built customized rings that couple a high-frequency AC signal (10–80 Mhz) to the finger to detect the skin-touch events. However, some of these on-body systems require wearing a separate transceiver on the body part being touched which is impractical in face-touch scenarios. Our ring prototype requires only a single point of instrumentation of both the transmitter and receiver and overcomes the signal saturation problem, achieving precise face-touch detection across different users and surfaces. Besides, all of these on-body sensing systems drain the battery in a few hours because they require always-on sensing and continuous signal processing to monitor and alert the users. Our system leverages three sensing modalities and a cascading classifier to filter out irrelevant hand gestures and minimize the active time of the energy-consuming components. As a result, the power consumption is several magnitudes lower than existing solutions.

#### 4.9 Conclusion

To conclude, we propose FaceTouch, a low-power, practical, and user-friendly face-touch detection for epidemiological surveillance. FaceTouch consists of a wrist-based IMU sensor and a novel ring-based vibration sensor. To simultaneously achieve high detection precision and low energy consumption, our magic relies on four hierarchical classifiers. We first apply two energy-efficient IMU-based classifiers to filter out irrelevant gestures. For face-touch like gestures, we use an IMU-based DNN classifier and a vibration-wave based classifier to guarantee high precision and recall. We implement FaceTouch using off-the-shelf hardware components and evaluate its performance under various complex scenarios across ten participants. Experimental results show that the F-1 score is 93.5%. The power consumption is 60.89  $\mu\text{W}$  in normal usage and 209.15  $\mu\text{W}$  in extremely

heavy usage.

## CHAPTER 5

### CONCLUSION

This dissertation shows innovative research and development in lower-power AIoT systems, focusing on optimizing energy efficiency, data processing, network reliability, and so on.

In Chapter 2, we present DeepLoRa. DeepLoRa adopts a deep learning-based approach that utilizes fine-grained landcover information extracted from remote sensing images to accurately estimate the path loss of long-distance LoRa links in complex environments. Compared with previous environment-aware models, DeepLoRa enables per-link estimation that takes both the type and order of landcovers into account. DeepLoRa also shows fair transferability. The evaluation shows that DeepLoRa reduces the estimation error to less than 4 dB, which is  $2\times$  smaller than state-of-the-art models. In Chapter 3, we further present LoSee, which shows a fine-grained LoRa link-level measurement in a  $6\times 6$   $km^2$  urban area. By such measurement, LoSee studies three fundamental research issues and draws the following conclusions: 1) The spatial and temporal behavior of LoRa links is quite dynamic due to environmental factors; 2) The coverage of LoRa gateways is anisotropic; 3) The median error of RSSI-fingerprint-based localization in a given setting is about 400 m. Without densely deployed LoRa gateways, the SOTA LoRa localization can support road-level localization.

In Chapter 4, we propose FaceTouch, a low-power multimodal wearable system that enables AI algorithms to monitor face-touch events to protect people from virus infection during the COVID-19 pandemic. We implement FaceTouch using commercial off-the-shelf hardware components and evaluate its performance with various user activities and false-positive behaviors. FaceTouch achieves a 93.5% F-1 score of face-touch detection and 60.89-209.15  $\mu W$  power consumption depending on usage, which is several magnitudes lower than the state-of-the-art systems.

There are still many research topics along with this dissertation, and many possible future works still remain to be addressed. Regarding improving large-scale LoRa deployment with AI techniques, weak signal detection, optimum transmission configuration prediction, and gateway deployment planning are critically important; Also, there's still much room for improvement in localization ac-

curacy in LoRaWAN systems with advanced models. To enable AI algorithms with IoT infrastructures in complicated real-world applications, challenges like selecting and combining appropriate sensing modalities as information sources in IoT systems, designing lightweight AI models that can achieve efficient on-device computing, and further miniaturizing system components still require exploration and investigation.

## BIBLIOGRAPHY

- [1] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, “Do lora low-power wide-area networks scale?” in *Proceedings of ACM MSWiM*, 2016.
- [2] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, “On the coverage of lpwans: range evaluation and channel attenuation model for lora technology,” in *Proceedings of IEEE ITST*, 2015.
- [3] O. Iova, A. Murphy, G. P. Picco, L. Ghio, D. Molteni, F. Ossi, and F. Cagnacci, “Lora from the city to the mountains: Exploration of hardware and environmental factors,” in *Proceedings of EWSN*, 2017.
- [4] S. Kartakis, B. D. Choudhary, A. D. Gluhak, L. Lambrinos, and J. A. McCann, “Demystifying low-power wide-area communications for city iot applications,” in *Proceedings of ACM WiNTECH*, 2016.
- [5] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, “Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios,” *IEEE Wireless Communications*, 2016.
- [6] B. Moyer, “Low power wide area: A survey of longer-range iot wireless protocols (2015),” retrieved sept. 7, 2015.”
- [7] A. J. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadinia, and N. Strachan, “Evaluation of lora and lorawan for wireless sensor networks,” in *Proceedings of IEEE SENSORS*, 2016.
- [8] Y. Okumura, “Field strength and its variability in vhf and uhf land-mobile radio service,” *Rev. Electr. Commun. Lab.*, 1968.
- [9] M. Hata, “Empirical formula for propagation loss in land mobile radio services,” *IEEE Transactions on Vehicular Technology*, 1980.
- [10] S. Demetri, M. Zúñiga, G. P. Picco, F. Kuipers, L. Bruzzone, and T. Telkamp, “Automated estimation of link quality for lora: a remote sensing approach,” in *Proceedings of ACM/IEEE IPSN*, 2019.
- [11] Y. Lin, W. Dong, Y. Gao, and T. Gu, “Sateloc: A virtual fingerprinting approach to outdoor lora localization using satellite images,” in *Proceedings of ACM/IEEE IPSN*, 2020.
- [12] T. Hatta and S. J. Dimond, “Differences in face touching by japanese and british people,” *Neuropsychologia*, vol. 22, no. 4, pp. 531–534, 1984.
- [13] S. Dimond and R. Harries, “Face touching in monkeys, apes and man: Evolutionary origins and cerebral asymmetry,” *Neuropsychologia*, vol. 22, no. 2, pp. 227–233, 1984.
- [14] Y. L. A. Kwok, J. Gralton, and M.-L. McLaws, “Face touching: a frequent habit that has implications for hand hygiene,” *American journal of infection control*, vol. 43, no. 2, pp. 112–114, 2015.

- [15] K. Morita, K. Hashimoto, M. Ogata, H. Tsutsumi, S.-i. Tanabe, and S. Hori, “Measurement of face-touching frequency in a simulated train,” in *E3S Web of Conferences*, vol. 111. EDP Sciences, 2019, p. 02027.
- [16] T. Singhal, “A review of coronavirus disease-2019 (covid-19),” *The Indian Journal of Pediatrics*, pp. 1–6, 2020.
- [17] M.-L. McLaws, A. A. Chughtai, S. Salmon, and C. R. MacIntyre, “A highly precautionary doffing sequence for health care workers after caring for wet ebola patients to further reduce occupational acquisition of ebola,” *American journal of infection control*, vol. 44, no. 7, pp. 740–744, 2016.
- [18] M. Liu, J. Ou, L. Zhang, X. Shen, R. Hong, H. Ma, B.-P. Zhu, and R. E. Fontaine, “Protective effect of hand-washing and good hygienic habits against seasonal influenza: a case-control study,” *Medicine*, vol. 95, no. 11, 2016.
- [19] N. Zhang, P. Wang, T. Miao, P.-T. Chan, W. Jia, P. Zhao, B. Su, X. Chen, and Y. Li, “Real human surface touch behavior based quantitative analysis on infection spread via fomite route in an office,” *Building and Environment*, vol. 191, p. 107578, 2021.
- [20] C. Rojas, N. Poulsen, M. Van Tuyl, D. Vargas, Z. Cohen, J. Paradiso, P. Maes, K. Esvelt, and F. Adib, “A scalable solution for signaling face touches to reduce the spread of surface-based pathogens,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 1, 2021.
- [21] S. Yun, Y.-C. Chen, H. Zheng, L. Qiu, and W. Mao, “Strata: Fine-grained acoustic-based device-free tracking,” in *Proceedings of the 15th annual international conference on mobile systems, applications, and services*, 2017, pp. 15–28.
- [22] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, “Fingerio: Using active sonar for fine-grained finger tracking,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 1515–1525.
- [23] W. Mao, J. He, and L. Qiu, “Cat: high-precision acoustic motion tracking,” in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 2016, pp. 69–81.
- [24] B. Kellogg, V. Talla, and S. Gollakota, “Bringing gesture recognition to all devices,” in *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*, 2014, pp. 303–316.
- [25] C. Zhao, K.-Y. Chen, M. T. I. Aumi, S. Patel, and M. S. Reynolds, “Sideswipe: detecting in-air gestures around mobile devices using actual gsm signal,” in *Proceedings of the 27th annual ACM symposium on User interface software and technology*, 2014, pp. 527–534.
- [26] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, “3d tracking via body radio reflections,” in *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*, 2014, pp. 317–329.

- [27] N. D’Aurizio, T. L. Baldi, S. Marullo, G. Paolucci, and D. Prattichizzo, “Reducing face-touches to limit covid-19 outbreak: an overview of solutions,” in *2021 29th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2021, pp. 645–650.
- [28] D. Chen, M. Wang, C. He, Q. Luo, Y. Irvantchi, A. Sample, K. G. Shin, and X. Wang, “Magx: wearable, untethered hands tracking with passive magnets,” in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 269–282.
- [29] V. Kakaraparathi, Q. Shao, C. J. Carver, T. Pham, N. Bui, P. Nguyen, X. Zhou, and T. Vu, “Facesense: Sensing face touch with an ear-worn system,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 3, pp. 1–27, 2021.
- [30] S. Marnilo, T. L. Baldi, G. Paolucci, N. D’Aurizio, and D. Prattichizzo, “No face-touch: Exploiting wearable devices and machine learning for gesture detection,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4187–4193.
- [31] X. Anthony’Chen, “Faceoff: Detecting face touching with a wrist-worn accelerometer,” *arXiv e-prints*, pp. arXiv–2008, 2020.
- [32] “Nudge,” <https://www.nudgeband.co.uk/>, 2021, Accessed: 2021-11-03.
- [33] “Immutouch: Stay healthy and hygienic with immutouch,” <https://immutouch.com/>, 2021, accessed: 2021-11-03.
- [34] “Face Touch Aware : the apple watch app,” <https://facetouch.app>, 2021, Accessed: 2021-11-03.
- [35] S. Gupta, D. Morris, S. Patel, and D. Tan, “Soundwave: using the doppler effect to sense gestures,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 1911–1914.
- [36] Y. Qifan, T. Hao, Z. Xuebing, L. Yin, and Z. Sanfeng, “Dolphin: Ultrasonic-based gesture recognition on smartphone platform,” in *2014 IEEE 17th International Conference on Computational Science and Engineering*. IEEE, 2014, pp. 1461–1468.
- [37] W. Ruan, Q. Z. Sheng, L. Yang, T. Gu, P. Xu, and L. Shangguan, “Audiogest: enabling fine-grained hand gesture detection by decoding echo signal,” in *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*, 2016, pp. 474–485.
- [38] X. Li, H. Dai, L. Cui, and Y. Wang, “Sonicoperator: Ultrasonic gesture recognition with deep neural network on mobiles,” in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*. IEEE, 2017, pp. 1–7.
- [39] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, “Whole-home gesture recognition using wireless signals,” in *Proceedings of the 19th annual international conference on Mobile computing & networking*, 2013, pp. 27–38.

- [40] L. Mo, Y. He, Y. Liu, J. Zhao, S.-J. Tang, X.-Y. Li, and G. Dai, "Canopy closure estimates with greenorbs: sustainable sensing in the forest," in *Proceedings of ACM SenSys*, 2009.
- [41] X. Mao, X. Miao, Y. He, X.-Y. Li, and Y. Liu, "Citysee: Urban CO<sub>2</sub> monitoring with sensors," in *Proceedings of IEEE INFOCOM*, 2012.
- [42] M. Ceriotti, M. Corrà, L. D'Orazio, R. Doriguzzi, D. Facchin, S. Guna, G. P. Jesi, R. L. Cigno, L. Mottola, A. L. Murphy *et al.*, "Is there light at the ends of the tunnel? wireless sensor networks for adaptive lighting in road tunnels," in *Processing of IEEE/ACM IPSN*, 2011.
- [43] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of lorawan for iot: From technology to application," *Sensors*, 2018.
- [44] L. Li, J. Ren, and Q. Zhu, "On the application of lora lpwan technology in sailing monitoring system," in *Proceedings of IEEE WONS*. IEEE, 2017.
- [45] B. Reynders and S. Pollin, "Chirp spread spectrum as a modulation technique for long range communication," in *Proceedings of IEEE SCVT*, 2016.
- [46] L. Vangelista, A. Zanella, and M. Zorzi, "Long-range iot technologies: The dawn of lora™," in *Future access enablers of ubiquitous and intelligent infrastructures*. Springer, 2015.
- [47] Y. Yao, Z. Ma, and Z. Cao, "Losee: Long-range shared bike communication system based on lorawan protocol," in *Proceedings of EWSN*, 2019.
- [48] K.-H. Lam, C.-C. Cheung, and W.-C. Lee, "Lora-based localization systems for noisy outdoor environment," in *Proceedings of IEEE WiMob*, 2017.
- [49] M. Aernouts, R. Berkvens, K. Van Vlaenderen, and M. Weyn, "Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas," *Data*, 2018.
- [50] W. Choi, Y.-S. Chang, Y. Jung, and J. Song, "Low-power lora signal-based outdoor positioning using fingerprint algorithm," *ISPRS International Journal of Geo-Information*, 2018.
- [51] H. T. Friis, "A note on a simple transmission formula," *Proceedings of IEEE IRE*, 1946.
- [52] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of lora: Experiences from a large-scale measurement study," *ACM Transactions on Sensor Networks*, 2019.
- [53] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. Prentice Hall PTR New Jersey, 1996, vol. 2.
- [54] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, 2015.
- [55] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, 2015.
- [56] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, 2014.

- [57] C. A. Oroza, Z. Zhang, T. Watteyne, and S. D. Glaser, "A machine-learning-based connectivity model for complex terrain large-scale low-power wireless deployments," *IEEE Transactions on Cognitive Communications and Networking*, 2017.
- [58] Y. Zhang, J. Wen, G. Yang, Z. He, and X. Luo, "Air-to-air path loss prediction based on machine learning methods in urban environments," *Wireless Communications and Mobile Computing*, 2018.
- [59] S. I. Popoola, S. Misra, and A. A. Atayero, "Outdoor path loss predictions based on extreme learning machine," *Wireless Personal Communications*, 2018.
- [60] H. Cheng, H. Lee, and S. Ma, "Cnn-based indoor path loss modeling with reconstruction of input images," in *Proceedings of IEEE ICTC*, 2018.
- [61] E. Ostlin, H.-J. Zepernick, and H. Suzuki, "Macrocell path-loss prediction using artificial neural networks," *IEEE Transactions on Vehicular Technology*, 2010.
- [62] A. Novelli, M. A. Aguilar, A. Nemmaoui, F. J. Aguilar, and E. Tarantino, "Performance evaluation of object based greenhouse detection from sentinel-2 msi and landsat 8 oli data: A case study from almería (spain)," *International journal of applied earth observation and geoinformation*, 2016.
- [63] M. Pesaresi, C. Corbane, A. Julea, A. J. Florczyk, V. Syrris, and P. Soille, "Assessment of the added-value of sentinel-2 for detecting built-up areas," *Remote Sensing*, 2016.
- [64] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2005.
- [65] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE transactions on Neural Networks*, 2002.
- [66] C. Huang, L. Davis, and J. Townshend, "An assessment of support vector machines for land cover classification," *International Journal of remote sensing*, 2002.
- [67] O. Vinyals, S. V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust asr," in *Proceedings of IEEE ICASSP*, 2012.
- [68] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *Proceedings of ICML*, 2011.
- [69] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 1997.
- [70] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, 1997.
- [71] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the international conference on artificial intelligence and statistics*, 2011.
- [72] D. Mandrioli, "Semantic segmentation editor," <https://github.com/Hitachi-Automotive-And-Industry-Lab/semantic-segmentation-editor>.

- [73] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- [74] L. Alliance, “A technical overview of lora and lorawan,” in <https://lora-alliance.org/resource-hub/what-lorawanr>, Retrieved by July 19th 2021.
- [75] A. Research, “Nb-iot and lte-m issues to boost lora and sigfox near and long-term lead in lpwa network connections,” in <https://tinyurl.com/2026-cellular-iot>, Retrieved by July 19th 2021.
- [76] L. SX1276, “77/78/79 datasheet, rev. 4,” *Semtech, March*, 2015.
- [77] C. Li, H. Guo, S. Tong, X. Zeng, Z. Cao, M. Zhang, Q. Yan, L. Xiao, J. Wang, and Y. Liu, “Nelora: Towards ultra-low snr lora communication with neural-enhanced demodulation,” in *Proceedings of ACM SenSys*, 2021.
- [78] C. Li and Z. Cao, “Lora networking techniques for large-scale and long-term iot: A down-to-top survey,” *ACM Computing Surveys*, 2022.
- [79] L. Li, Y. Yuguang, C. Zhichao, and Z. Mi, “Deeplora: Learning accurate path loss model for long distance links in lpwan,” in *Proceedings of IEEE INFOCOM*, 2021.
- [80] J. Yang, Z. Xu, and J. Wang, “Ferrylink: Combating link degradation for practical lpwan deployments,” in *Proceedings of IEEE ICPADS*. IEEE, 2021.
- [81] J. Navarro-Ortiz, S. Sendra, P. Ameigeiras, and J. M. Lopez-Soler, “Integration of lorawan and 4g/5g for the industrial internet of things,” *IEEE Communications Magazine*, 2018.
- [82] J. Haxhibeqiri, A. Karaagac, F. Van den Abeele, W. Joseph, I. Moerman, and J. Hoebeke, “Lora indoor coverage and performance in an industrial environment: Case study,” in *Proceedings of IEEE international conference on emerging technologies and factory automation (ETFA)*, 2017.
- [83] J. Petäjajarvi, K. Mikhaylov, M. Hämäläinen, and J. Iinatti, “Evaluation of lora lpwan technology for remote health and wellbeing monitoring,” in *Proceeding of the International Symposium on Medical Information and Communication Technology*, 2016.
- [84] D. Magrin, M. Centenaro, and L. Vangelista, “Performance evaluation of lora networks in a smart city scenario,” in *Proceeding of IEEE ICC*, 2017.
- [85] Y. Wang, X. Zheng, L. Liu, and H. Ma, “Polartracker: Attitude-aware channel access for floating low power wide area networks,” in *Proceedings of IEEE INFOCOM*, 2021.

- [86] W. Xu, J. Y. Kim, W. Huang, S. S. Kanhere, S. K. Jha, and W. Hu, "Measurement, characterization, and modeling of lora technology in multifloor buildings," *IEEE Internet of Things Journal*, 2019.
- [87] C. Li, X. Guo, L. Shuangguan, Z. Cao, and K. Jamieson, "Curvinglora to boost lora network throughput via concurrent transmission," in *Proceedings of USENIX NSDI*, 2022.
- [88] *GPS-free geolocation using LoRa in low-power WANs*. IEEE, 2017.
- [89] N. Podevijn, D. Plets, J. Trogh, L. Martens, P. Suanet, K. Hendrikse, and W. Joseph, "Tdoa-based outdoor positioning with tracking algorithm in a public lora network," *Wireless Communications and Mobile Computing*, 2018.
- [90] N. Podevijn, D. Plets, M. Aernouts, R. Berkvens, L. Martens, M. Weyn, and W. Joseph, "Experimental tdoa localisation in real public lora networks," in *Proceedings of CEUR Workshop Proceedings*, 2019.
- [91] D. F. Carvalho, A. Depari, P. Ferrari, A. Flammini, S. Rinaldi, and E. Sisinni, "On the feasibility of mobile sensing and tracking applications based on lpwan," in *Proceedings of IEEE SAS*. IEEE, 2018.
- [92] A. Dongare, C. Hesling, K. Bhatia, A. Balanuta, R. L. Pereira, B. Iannucci, and A. Rowe, "Openchirp: A low-power wide-area networking architecture," in *Proceedings of IEEE PerCom Workshops*, 2017.
- [93] C. Gu, L. Jiang, and R. Tan, "Lora-based localization: Opportunities and challenges," *arXiv preprint arXiv:1812.11481*, 2018.
- [94] R. Nandakumar, V. Iyer, and S. Gollakota, "3d localization for sub-centimeter sized devices," in *Proceedings of ACM SenSys*, 2018.
- [95] A. Bansal, A. Gadre, V. Singh, A. Rowe, B. Iannucci, and S. Kumar, "Owl: Accurate lora localization using the tv whitespaces," in *Proceedings of the 20th International Conference on Information Processing in Sensor Networks*, 2021.
- [96] H. Sallouha, A. Chiumento, and S. Pollin, "Localization in long-range ultra narrow band iot networks using rssi," in *Proceedings of IEEE ICC*, 2017.
- [97] Y. Li, Z. He, Y. Li, H. Xu, L. Pei, and Y. Zhang, "Towards location enhanced iot: Characterization of lora signal for wide area localization," in *Proceedings of IEEE UPINLBS*. IEEE, 2018.
- [98] S. Tong, J. Wang, and Y. Liu, "Combating packet collisions using non-stationary signal scaling in LPWANs," in *Proceedings of ACM MobiSys*, 2020.
- [99] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, and X. Li, "Does wireless sensor network scale? a measurement study on greenorbs," *IEEE Transactions on Parallel and Distributed Systems*, 2012.

- [100] W. Dong, Y. Liu, Y. He, T. Zhu, and C. Chen, “Measurement and analysis on the packet delivery performance in a large-scale sensor network,” *IEEE/ACM Transactions on Networking*, 2013.
- [101] J. Wang and W. Dong, “Understanding the link-level behaviors of a large scale urban sensor network,” in *Proceedings of IEEE MSN*, 2016.
- [102] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, “Link-level measurements from an 802.11 b mesh network,” in *Proceedings of ACM SIGCOMM*, 2004.
- [103] C. K. Williams and C. E. Rasmussen, “Gaussian processes for regression,” 1996.
- [104] Y. Sangar and B. Krishnaswamy, “Wichronos: Energy-efficient modulation for long-range, large-scale wireless networks,” in *Proceedings of ACM MobiCom*, 2020.
- [105] C. Li, Z. Cao, and L. Xiao, “Curvealoha: Non-linear chirps enabled high throughput random channel access for lora,” in *Proceedings of IEEE INFOCOM*, 2022.
- [106] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe, “Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks,” in *Proceedings of ACM/IEEE IPSN*, 2018.
- [107] A. Gadre, R. Narayanan, A. Luong, A. Rowe, B. Iannucci, and S. Kumar, “Frequency Configuration for Low-Power Wide-Area Networks in a Heartbeat,” in *Proceedings of USENIX NSDI*, 2020.
- [108] L. Chen, J. Xiong, X. Chen, S. I. Lee, K. Chen, D. Han, D. Fang, Z. Tang, and Z. Wang, “WideSee: towards wide-area contactless wireless sensing,” in *Proceedings of ACM SenSys*, 2019.
- [109] R. Nandakumar, V. Iyer, and S. Gollakota, “3D Localization for Sub-Centimeter Sized Devices,” in *Proceedings of ACM SenSys*, 2018.
- [110] F. Zhang, Z. Chang, K. Niu, J. Xiong, B. Jin, Q. Lv, and D. Zhang, “Exploring LoRa for Long-range Through-wall Sensing,” *Proceedings of ACM IMWUT*, 2020.
- [111] S. Zhang, W. Wang, N. Zhang, and T. Jiang, “RF Backscatter-based State Estimation for Micro Aerial Vehicles,” in *Proceedings of IEEE INFOCOM*, 2020.
- [112] W. C. McGrew and L. F. Marchant, “On the other hand: current issues in and meta-analysis of the behavioral laterality of hand function in nonhuman primates,” *American Journal of Physical Anthropology: The Official Publication of the American Association of Physical Anthropologists*, vol. 104, no. S25, pp. 201–232, 1997.
- [113] L. J. Rogers, G. Vallortigara, and R. J. Andrew, *Divided brains: the biology and behaviour of brain asymmetries*. Cambridge University Press, 2013.
- [114] N. Zhang, W. Jia, P. Wang, M.-F. King, P.-T. Chan, and Y. Li, “Most self-touches are with the nondominant hand,” *Scientific reports*, vol. 10, no. 1, pp. 1–13, 2020.

- [115] S. L. Warnes, Z. R. Little, and C. W. Keevil, “Human coronavirus 229e remains infectious on common touch surface materials,” *MBio*, vol. 6, no. 6, pp. e01697–15, 2015.
- [116] A. Guellich, E. Tella, M. Ariane, C. Grodner, H.-N. Nguyen-Chi, and E. Mahé, “The face mask-touching behavior during the covid-19 pandemic: Observational study of public transportation users in the greater paris region: The french-mask-touch study,” *Journal of transport & health*, vol. 21, p. 101078, 2021.
- [117] Z. Witkower and J. L. Tracy, “Bodily communication of emotion: Evidence for extrafacial behavioral expressions and available coding systems,” *Emotion Review*, vol. 11, no. 2, pp. 184–193, 2019.
- [118] R. Haratian, “Assistive wearable technology for mental wellbeing: Sensors and signal processing approaches,” in *2019 5th International Conference on Frontiers of Signal Processing (ICFSP)*. IEEE, 2019, pp. 7–11.
- [119] S. M. Mueller, S. Martin, and M. Grunwald, “Self-touch: contact durations and point of touch of spontaneous facial self-touches differ depending on cognitive and emotional load,” *PloS one*, vol. 14, no. 3, p. e0213677, 2019.
- [120] M. De Zambotti, N. Cellini, A. Goldstone, I. M. Colrain, and F. C. Baker, “Wearable sleep technology in clinical and research settings,” *Medicine and science in sports and exercise*, vol. 51, no. 7, p. 1538, 2019.
- [121] A. Parate, M.-C. Chiu, C. Chadowitz, D. Ganesan, and E. Kalogerakis, “Risq: Recognizing smoking gestures with inertial sensors on a wristband,” in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, 2014, pp. 149–161.
- [122] R. E. Wright, “Logistic regression.” 1995.
- [123] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.
- [124] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [125] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, “Virtual adversarial training: a regularization method for supervised and semi-supervised learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.
- [126] R. O. Nelson, R. A. Boykin, and S. C. Hayes, “Long-term effects of self-monitoring on reactivity and on accuracy,” *Behaviour research and therapy*, vol. 20, no. 4, pp. 357–363, 1982.
- [127] S. Foti, C. G. Lai, G. J. Rix, and C. Strobbia, *Surface wave methods for near-surface site characterization*. CRC press, 2014.

- [128] N. Roy, M. Gowda, and R. R. Choudhury, “Ripple: Communicating through physical vibration,” in *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*, 2015, pp. 265–278.
- [129] C. Zhang, Q. Xue, A. Waghmare, R. Meng, S. Jain, Y. Han, X. Li, K. Cunefare, T. Ploetz, T. Starner *et al.*, “Fingerping: Recognizing fine-grained hand poses using active acoustic on-body sensing,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–10.
- [130] J. Gong, A. Gupta, and H. Benko, “Acustico: Surface tap detection and localization using wrist-based acoustic tdoa sensing,” in *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 2020, pp. 406–419.
- [131] W. Wang, L. Yang, and Q. Zhang, “Touch-and-guard: secure pairing through hand resonance,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 670–681.
- [132] “TDK piezolisten™ piezo transducers,” [https://product.tdk.com/en/search/sw\\_piezo/speaker/piezolisten/info?part\\_no=PHUA2010-049B-00-000&utm\\_source=piezolisten\\_commercial\\_phu\\_en.pdf&utm\\_medium=catalog](https://product.tdk.com/en/search/sw_piezo/speaker/piezolisten/info?part_no=PHUA2010-049B-00-000&utm_source=piezolisten_commercial_phu_en.pdf&utm_medium=catalog), 2010, accessed: 2010-09-30.
- [133] R. A. Serway and C. Vuille, *College physics*. Cengage Learning, 2014.
- [134] D. Rachaveti, N. Chakrabhavi, V. Shankar, and S. Varadhan, “Thumbs up: movements made by the thumb are smoother and larger than fingers in finger-thumb opposition tasks,” *PeerJ*, vol. 6, p. e5763, 2018.
- [135] F. W. Jones, *Finger-ring Lore: Historical, Legendary, Anecdotal*. Good Press, 2019.
- [136] N. J. Mansfield, *Human response to vibration*. CRC press, 2004.
- [137] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Annals of statistics*, pp. 1189–1232, 2001.
- [138] J. Diebel, “Representing attitude: Euler angles, unit quaternions, and rotation vectors,” *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.
- [139] S. Menard, *Applied logistic regression analysis*. Sage, 2002, vol. 106.
- [140] “Adafruit bluefruit le uart friend - bluetooth low energy (ble),” 2022, accessed: 2022-01-30.
- [141] “Monsoon high voltage power monitor,” <https://www.msoon.com/high-voltage-power-monitor>, 2021, accessed: 2021-11-03.
- [142] W. Kienzle, E. Whitmire, C. Rittaler, and H. Benko, “Electroring: Subtle pinch and touch detection with a ring,” in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–12.
- [143] Z. Song, Z. Cao, Z. Li, and J. Wang, “Magic wand: Towards plug-and-play gesture recognition on smartwatch,” in *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2020, pp. 275–282.

- [144] Z. Song, Z. Cao, Z. Li, J. Wang, and Y. Liu, “Inertial motion tracking on mobile and wearable devices: Recent advancements and challenges,” *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 692–705, 2021.
- [145] M. Altini and H. Kinnunen, “The promise of sleep: A multi-sensor approach for accurate sleep stage detection using the oura ring,” *Sensors*, vol. 21, no. 13, p. 4302, 2021.
- [146] D. Phan, L. Y. Siong, P. N. Pathirana, and A. Seneviratne, “Smartwatch: Performance evaluation for long-term heart rate monitoring,” in *2015 International symposium on bioelectronics and bioinformatics (ISBB)*. IEEE, 2015, pp. 144–147.
- [147] A. Maijala, H. Kinnunen, H. Koskimäki, T. Jämsä, and M. Kangas, “Nocturnal finger skin temperature in menstrual cycle tracking: ambulatory pilot study using a wearable oura ring,” *BMC Women’s Health*, vol. 19, no. 1, pp. 1–10, 2019.
- [148] R. Mishra, H. P. Gupta, and T. Dutta, “A survey on deep neural network compression: Challenges, overview, and solutions,” *arXiv preprint arXiv:2010.03954*, 2020.
- [149] Y. Zhang, J. Zhou, G. Laput, and C. Harrison, “Skintrack: Using the body as an electrical waveguide for continuous finger tracking on the skin,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 1491–1503.
- [150] Y. Zhang, W. Kienzle, Y. Ma, S. S. Ng, H. Benko, and C. Harrison, “Actitouch: Robust touch detection for on-skin ar/vr interfaces,” in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 2019, pp. 1151–1159.