

A TEMPERATURE AND ACTIVITY BASED ADAPTIVE SMARTPHONE SENSING
FRAMEWORK IN THE CONTEXT OF URBAN HEAT ISLANDS

by

HIMANSHU PENDYALA

(Under the Direction of Lakshmish Ramaswamy)

ABSTRACT

Modern smartphones are equipped with powerful sensors and communication technologies which can provide useful information about the user including location, altitude, speed etc. and can communicate this information with external devices. This makes them ideal devices for research purposes as they can either be used as sensors themselves or can be used to communicate with sensors. Sensing at a high static sampling frequency leads to rapid depletion of energy sources. On the other hand, sensing at a lower frequency may not capture important events in the surroundings. Therefore, there is a need for adaptive sensing mechanism which can regulate the sampling frequency without compromising the accuracy of data collected. As part of this research, activity and temperature based adaptive sensing mechanisms were developed and tested. These adaptive sampling approaches are applied in the context of Urban Heat Islands to collect spatial-temporal weather data efficiently.

INDEX WORDS: Adaptive sampling, Crowdsensing, Sensor, Smartphones, Urban Heat
Islands

A TEMPERATURE AND ACTIVITY BASED ADAPTIVE SMARTPHONE SENSING
FRAMEWORK IN THE CONTEXT OF URBAN HEAT ISLANDS

by

HIMANSHU PENDYALA

BE, BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCES, INDIA, 2015

MS, BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCES, INDIA, 2015

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment
of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2018

© 2018

HIMANSHU PENDYALA

All Rights Reserved

A TEMPERATURE AND ACTIVITY BASED ADAPTIVE SMARTPHONE SENSING
FRAMEWORK IN THE CONTEXT OF URBAN HEAT ISLANDS

by

HIMANSHU PENDYALA

Major Professor: Lakshmish Ramaswamy
Committee: Deepak Mishra
John. A. Miller

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
December 2018

DEDICATION

To my family for their endless love, support and encouragement.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude for the continuous support and encouragement provided by Dr. Lakshmish Ramaswamy during my graduate academic career. He provided me with inputs needed to conduct this research from time to time and guided me by assigning projects which made a better computer science engineer. I would like to thank him for all the patience and time he took out to guide me.

I would also like to thank my committee members Dr. Deepak Mishra and Dr. John Miller for their time and guidance. Lastly, I would like to thank all my professors, friends, Department of Computer Science and University of Georgia for making my stay at UGA a memorable one.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER	
1 INTRODUCTION.....	1
1.1 INTRODUCTION.....	1
1.2 CROWDSENSING.....	2
1.3 ADAPTIVE SAMPLING.....	3
1.4 CHALLENGES IN CROWDSENSING.....	5
1.5 THESIS CONTRIBUTION.....	5
2 BACKGROUND.....	7
2.1 URBAN HEAT ISLANDS.....	7
2.2 MOTIVATION.....	10
2.3 KESTREL WEATHER DATA LOGGER.....	12
2.4 RESEARCH ON UHI AT THE UNIVERSITY OF GEORGIA.....	13
2.5 NEED FOR ADAPTIVE SENSING FRAMEWORK	14
3 TEMPERATURE AND ACTIVITY BASED ADAPTIVE SAMPLING STRATEGY.....	16
3.1 TEMPERATURE BASED ADAPTIVE SAMPLING.....	17

3.2	ACTIVITY BASED ADAPTIVE SAMPLING.....	19
4	OVERVIEW OF ANDROID COMPONENTS.....	21
4.1	ANDROID ECOSYSTEM.....	21
4.2	BLUETOOTH ON ANDROID.....	22
4.3	ANDROID COMPONENTS.....	24
5	APPLICATION ARCHITECTURE AND DESIGN.....	28
5.1	APPLICATION OVERVIEW.....	28
5.2	APPLICATION ARCHITECTURE.....	29
6	EXPERIMENTS AND RESULTS.....	33
6.1	EXPERIMENTAL SETUP.....	33
6.2	EXPERIMENTAL DATASET.....	35
6.3	EXPERIMENTAL RESULTS.....	37
6.4	COMBINING ACTIVITY BASED ADAPTIVE SAMPLING AND TEMPERATURE BASED ADAPTIVE SAMPLING.....	47
7	CONCLUSIONS.....	48
	REFERENCES	50

LIST OF TABLES

	Page
Table 1: Predefined sampling frequencies for temperature based adaptive sampling.....	36
Table 2: Predefined sampling frequencies for activity based adaptive sampling.....	36
Table 3: Comparison between data collected by activity based adaptive sampling and static sampling for Experiment 1.	38
Table 4: Comparison between data collected by activity based adaptive sampling and static sampling for Experiment 2.	39
Table 5: Comparison between data collected by temperature based adaptive sampling and static sampling for Experiment 1.....	40
Table 6: Comparison between data collected by temperature based adaptive sampling and static sampling for Experiment 2.....	41
Table 7: Sampling frequencies used to evaluate temperature based adaptive sampling.....	41
Table 8: Sampling frequencies used to evaluate activity based adaptive sampling.....	41

LIST OF FIGURES

	Page
Figure 1: Urban Heat Islands.....	8
Figure 2: Kestrel Drop 2 data logger.....	13
Figure 3: Overview of the architecture of ‘UHI SCOUTS’ application.....	29
Figure 4: Screenshot showing the view where the participant can enter sensor details.....	31
Figure 5 Screenshot showing the view where data extracted from sensor is displayed	31
Figure 6: Activity based adaptive sampling Experiment 1	37
Figure 7: Activity based adaptive sampling Experiment 2.....	38
Figure 8: Temperature based adaptive sampling Experiment 1.....	39
Figure 9: Temperature based adaptive sampling Experiment 2.....	40
Figure 10: Standard RMSE Comparison for Activity Based Adaptive Sampling for Experiment Series 1.....	43
Figure 11: Nearest RMSE Comparison for Activity Based Adaptive Sampling for Experiment Series 1.....	44
Figure 12: Standard RMSE Comparison for Activity Based Adaptive Sampling for Experiment Series 2.....	44
Figure 13: Nearest RMSE Comparison for Activity Based Adaptive Sampling for Experiment Series 2.....	45
Figure 14: Nearest RMSE comparison for Temperature Based Adaptive Sampling	45
Figure 15: Standard RMSE comparison for Temperature Based Adaptive Sampling	46

CHAPTER 1

INTRODUCTION

This chapter discusses about the Urban Heat Island problem (UHI), and the adverse effects it has on community. We also discuss the existing sources of data to analyze UHIs and their need for additional data sources to better address this problem. In the last section, we succinctly discuss about the contribution of this research and the approach we proposed as one of the solutions to this problem.

1.1 Introduction

UHI is a widely prevalent phenomenon in urban areas. It is a phenomenon where the temperatures in the urban areas are significantly higher compared to the temperatures in the surrounding rural areas [1]. Replacement of natural vegetation cover with man-made structures like high rise buildings, pavements, roads and concrete structures increase the heat storage capacities and affect the surface albedo. Large amount of heat remains trapped in these surfaces and result in increased surface and air temperatures in urban areas.

UHIs have been studied extensively and standard methods and sources of data have been identified [2, 3, 4]. There are several sources of data available to study, analyze and map UHIs. Some of the traditional sources of spatial-temporal weather data are satellites, aircraft-borne sensors, stationary weather stations. Although these methods help us in analyzing UHIs, they provide relatively less granular spatial-temporal weather data. For instance, the thermal band resolution of Landsat8 satellite imagery that has been broadly used in remote sensing to produce

heat maps is 100 meters and for a specific location, it is only possible to obtain one image every sixteen days. Also, these maps are based on the surface reflectance data and do not accurately represent the actual heat experienced by the individuals. On the other hand, weather stations which are another popular source of temperature data, represent the temperatures of some static locations where they are installed. They are usually installed at the airports or on top of buildings and therefore could not be a reliable representation of the heat exposure to individuals.

1.2 Crowdsensing

Crowdsensing can be defined as a technique where groups of people equipped with mobile devices which are capable of sensing and computing assist in gathering information pertaining to events of interest [6]. Increased availability of inexpensive sensor devices and the rapid increase in the use of mobile phones enables crowdsourcing as an alternative source of collecting data.

The possibility of crowdsourcing temperature and weather observations has been identified in previous studies [7]. There are several research programs that have benefitted by utilizing crowdsourcing to obtain data. NoiseTube project [8] explored a crowdsourcing approach to estimate and map urban noise pollution using smartphones. ‘Cyanotracker’, a citizen science application helped individuals in reporting potential algal blooms to researchers by submitting photographs and text-based reports [9]. Crowdsourcing using vehicle and human borne sensors are being explored as a paradigm to improve spatial-temporal granularity.

These studies have explored the potential of utilizing crowdsourcing to collect weather data and map urban heat islands using them. A study titled “Sense the Zagreb Air” performed in the city of Zagreb, Croatia demonstrated an urban crowdsensing application for monitoring air quality using specially-designed wearable sensors and mobile phones [10]. One of the research studies, which explored the possibility of crowdsourcing utilized the temperature sensors present in

smartphones to effectively collect urban air temperatures [11]. An android application was developed to extract the temperature data from the sensor. The temperature readings given by the sensor in the mobile phone is modelled by a heat transfer model to reflect the actual environment temperature. This method proved to be more cost efficient when compared to deploying fixed-point weather stations at multiple locations. As the users move through different areas of the city, crowdsourcing can help in providing high resolution spatial data at an economical rate. With the high availability and widespread use of smartphones, crowdsourcing shows enormous potential for enabling large scale data collection in the future.

1.3 Adaptive Sampling

The availability of low-cost compact sensors and their ability to function without much need for human supervision make them an attractive tool for gathering data by sensing phenomena from the physical environment. There are several applications that benefitted by utilizing sensor deployments to monitor real-time environmental phenomena [12] [13]. Some areas where wireless sensor networks are successfully deployed are agriculture [14], vehicular telematics [15], patient monitoring in healthcare [16], collaborative target detection in military applications [17], water quality monitoring [18] etc. A typical operation flow in sensor network deployment involves individual sensor node obtaining information from their surrounding environment, processing the data and transmitting them to collection points [19]. Limited energy resources are the primary factor limiting the lifetime of deployed sensor nodes. Batteries powering sensor nodes have limited capacity it is not feasible to replace or recharge them due to environmental and cost constraints(factors). Sensors monitoring the environment usually collect data at a static sampling frequency. Although static sampling at a high frequency is a solution to capture

changes in the environment accurately, it results in redundant data being collected and rapid depletion of sensor's energy sources.

Therefore, there is a need for methods to sample data efficiently and reduce the conserve sensor's energy sources. For further motivation, a study on monitoring Storm Petrels (a sea bird), sensors registering temperature, sound and light are deployed in their burrows. The burrows are expected to be empty during the day and thus sensors can sample data at a lower frequency. In the context of a temperature monitoring sensor deployment, if the conditions in the environment are stable and the temperature does not vary, the state of the environment can be represented by a few data points collected over longer time intervals. In this case, static sampling at a high frequency results in collecting redundant data which is not required in studying the conditions of the environment. There has been some significant work done on different aspects of adaptive sampling. A study [20] explored adaptive sampling in the context of environmental monitoring by introducing a feedback control mechanism within each individual node, which adjusts the sampling frequency according to external conditions. Another study explored data driven adaptive sampling algorithm in the context of automated water quality monitoring [21] by making use of data already collected by the sensor nodes. Another research proposed a sensing architecture, where the sensing architecture is more quality aware and regulates the flow of data all the levels of the architecture according to the application's requirements [22]. This helps in conserving the network bandwidth by only allowing the flow of relevant data.

This research focuses on making the sampling process more efficient and as part of this implements activity and temperature based adaptive sampling strategies.

1.4 Challenges in crowdsensing

Although crowdsourcing data collection has proven to be an effective technique, there are some challenges in adopting this form of data collection. The orientation of the sensor placement and participant's location influence the measurements to a certain extent. For instance, when the sensor is placed inside participant's bag or pocket, the data becomes less representative of the actual temperature in that location. Similarly, on a windy day or when the participant is inside a building, under the tree shade etc., data reported is influenced by external factors. Limited energy resources of sensor devices, availability of network bandwidth to transmit data and computational power are some of the other challenges encountered while utilizing this approach [23]. One of the solutions to tackle these challenges is to eliminate collecting redundant data by making the sensing process adaptive. This reduces the amount of data transferred and helps preserve the energy resources of the sensor device. This research focuses on making use of crowdsourcing to obtain highly granular spatial-temporal weather data. This research also focuses on making the sampling process more adaptive, thereby leading to efficient use of sensor's energy resources and network bandwidth.

1.5 Thesis Contribution

This research focuses on developing methods to collect spatial-temporal weather data which can augment the existing data sources and enable better understanding of UHI. In this research, crowdsensing approach is explored for data collection. A mobile phone-based sensing framework is developed which aids in data collection. Since, static aggressive sensing with a predefined sampling frequency leads to faster depletion of the sensor's energy sources as well as mobile phone's energy sources, adaptive sensing strategies which aim at reducing the energy consumption of the devices are developed. The objective behind developing these adaptive

sensing strategies is to regulate the sampling frequency to conserve energy and ensure that the accuracy of the data is not compromised. There are two adaptive sampling strategies that are developed as part of this research – Temperature based adaptive sensing and Activity based adaptive sensing. Temperature based adaptive sensing adjusts the sampling frequency based on the previous temperature experienced by the subject. Activity based adaptive sensing utilizes Google Activity Recognition API to predict the subject’s activity based on the data from mobile phone sensors. These adaptive sampling strategies are deployed and tested rigorously. It has been observed that both the adaptive sampling strategies are much more energy efficient than sampling at a fixed frequency. On an average, sensor employing adaptive sampling strategy collected 75% less data compared to ordinary sensor and maintained data with relatively same accuracy. Accuracy of the data collected using adaptive sampling strategy and static sampling strategy are compared using root mean square error as a measure.

CHAPTER 2

BACKGROUND

2.1 Urban Heat Islands

Urbanization is one of the most profound examples of human intervention on natural environment. The availability of access to education, social and economic services have made urban localities favorable for living and resulted in large scale urbanization in recent times. According to United Nations reports, around 54% of the world's population are currently residing in urban areas and this number is projected to increase to 66% by 2050 [24]. Due to the substantial increase in CO₂ emissions, growing urban population has a direct influence on the global climate change. With such a high proportion of population living in urban areas, it is very important to understand the phenomena that determines urban climate and hence analyze the impacts of present and future urbanization patterns at the local and global scale.

Urban environments continuously change the original physical characteristics of a natural landscape by introducing various structures which have adverse effects on the environment. Urbanization has led to the replacement of vegetation cover with skyscrapers, roads and concrete surfaces. These structures increase the impervious surface area and heat storage capacities [25]. This increase in heat storage capacities further impacts surface emissivity or the ratio of the incoming electromagnetic radiation to reflected radiation. As land surface temperature plays a crucial role in maintaining energy balance between the land surface and atmosphere, the presence of impervious surfaces results in higher temperatures in urban areas as compared to rural areas. This phenomenon is referred to as the Urban Heat Island (UHI).

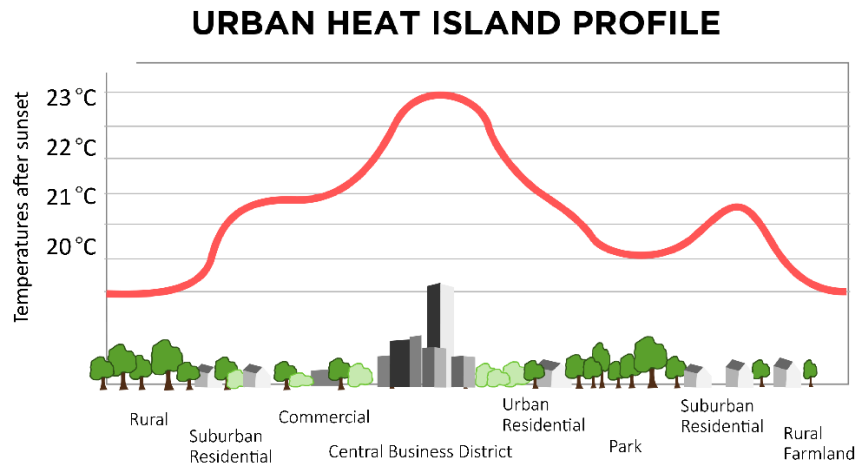


Figure 1: Urban Heat Islands [25]

UHIs results in a condition where the metropolitan areas are significantly warmer than the surrounding rural areas. For a given metropolitan area, difference in the temperature is directly proportional to the amount of urbanization [26]. However, based on the layer of the urban atmosphere being affected, UHIs can be more specifically defined into the following categories – Urban Canopy Layer (UCL) and Urban Boundary Layer (UBL). UCL is the layer of urban atmosphere present between the surface and the mean building height while UBL is the layer of atmosphere present above the UCL and extends up to the top of the buildings.

UHIs have several adverse effects on the population living in the urban areas. These effects include increased air pollution, heat-related illness and increased energy expenses. UHI are pointed as a potential cause for heat-related mortality. Sensitive population such as children and elderly people are susceptible to extreme heating conditions [27]. Previous research show that an average of 1000 people die every year in United States due to extreme heat [28]. Since UHIs are associated with increased temperatures in cities, they have the potential to extend the magnitude and duration of heat waves in urban areas. In a study on UHI conducted in 12 of the Asian megacities, it was found that the temperature difference in cities like Tokyo and

Bangkok was as high as 12°C and 8°C respectively during the day in summer [29]. This creates an increased demand for energy and electricity consumption especially during noon. Previous research indicates that energy demand for cooling increases from 1.5 to 2.0% for every 1°F (0.6°C) increase in air temperatures, starting from 68°F to 77°F (20°C to 25°C), suggesting that 5–10% of community-wide demand for electricity is used to compensate for the heat island effect [30]. Elevated temperatures due to UHI impacts the quality of air and environmental conditions. The combustion of fossil fuels produces volatile organic compounds which further increase the ozone production at ground level. This increase in the temperatures coupled with ground level ozone can cause respiratory and cardiovascular problems.

This research focuses on developing methods to collect spatial-temporal weather data which can augment the existing data sources and enable better understanding of UHI. In this research, crowdsensing approach is explored for data collection. A mobile phone-based sensing framework is developed which aids in data collection. Since, static aggressive sensing with a predefined sampling frequency leads to faster depletion of the sensor's energy sources as well as mobile phone's energy sources, adaptive sensing strategies which aim at reducing the energy consumption of the devices are developed. The objective behind developing these adaptive sensing strategies is to regulate the sampling frequency to conserve energy and ensure that the accuracy of the data is not compromised. There are two adaptive sampling strategies that are developed as part of this research – Temperature based adaptive sensing and Activity based adaptive sensing. Temperature based adaptive sensing adjusts the sampling frequency based on the previous temperature experienced by the subject. Activity based adaptive sensing utilizes Google Activity Recognition API to predict the subject's activity based on the data from mobile phone sensors. These adaptive sampling strategies are deployed and tested rigorously. It has been

observed that both the adaptive sampling strategies are much more energy efficient than sampling at a fixed frequency. It is also observed that activity based adaptive sampling performed relatively better than temperature based adaptive sampling. On an average, sensor employing adaptive sampling strategy collected 75% less data compared to ordinary sensor and maintained data with relatively same accuracy. Accuracy of the data collected using adaptive sampling strategy and static sampling strategy are compared using root mean square error as a measure.

2.2 Motivation

Sensors have proved to be an invaluable device in assisting scientific research by collecting environmental data. Their portable, energy efficient nature has enabled scientists to use them at various remote locations which prove to be challenging for humans to stay and collect data for prolonged period. Sensors automate the process of data collection and systems have been developed to transfer the data periodically to centralized data stores through wireless networks. Researchers and scientists are making extensive use of human-borne and vehicle-borne sensors to capture data and draw inferences from them.

The following traditional methods of obtaining surface weather (temperature) data were considered.

1. Satellite imagery: Satellites provide greater geographical coverage. However, they fail to provide detailed insights about the hot spots within the neighborhoods. The presence of tree-cover and high-rise buildings prevent the satellites from effectively capturing surface temperatures. Also, satellites can collect data pertaining to a location only when they pass over the specific location and the weather conditions are clear with no cloud cover. There are many available thermal infrared sensors to study land surface temperature. The

Geostationary Operational Environmental Satellite (GOES) has a 4-km resolution in the thermal infrared, while the NOAA-Advanced Very High-Resolution Radiometer (AVHRR), Terra and Aqua-Moderate Resolution Imaging Spectroradiometer (MODIS) have 1-km spatial resolution. High resolution data from the Terra-Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) has a 90-m resolution and Landsat-8 Enhanced Thematic Mapper (ETM+) has a 30-m resolution in thermal region [31].

2. Aircraft-borne instruments: Aircraft-borne or drone-borne instruments provide more granular data. However, this method of data collection is significantly more expensive than obtaining data using satellites or ground instruments.
3. Fixed point weather stations: Fixed point weather monitoring stations can be set up at various locations in the city to obtain weather data. This method provides higher resolution data but provide lesser geographical coverage due to their confined nature.

The above-mentioned methods for data collection have been used extensively by researchers and scientists to analyze UHIs [2, 3, 4, 5]. This research is focused on finding an efficient solution to collect spatial-temporal weather data. Firstly, relevance of spatial-temporal weather data collected from traditional sources was evaluated. Although, traditional sources of data help in mapping urban heat islands, there is a need for alternate sources of data to mitigate the limitations of this approach. Studies suggest that using alternate sources of data collection improves spatial resolution of the collected data and lead to more accurate analysis of urban heat islands [32]. Crowdsensing acts as an alternate data source which can be augmented to existing traditional sources of data to obtain highly granular weather data.

As part of this research, we utilized crowdsourcing as a tool to enable collecting high resolution weather data. Data obtained through traditional sources, when coupled with data obtained through crowdsourcing, provides higher resolution spatial-temporal weather data which helps map UHIs more accurately. External weather sensors and data loggers can be used to supplement the data collected from sources like satellite and fixed-point weather stations. Sensors are relatively inexpensive, compact and can be placed at remote locations without the need for constant human intervention. This make them an ideal choice for sourcing weather data. In this research, we focused on obtaining spatial-temporal weather data at several locations in a city. The method we adopted to do this involves extensive use of external sensors as well as sensors embedded in smartphones.

2.3 Kestrel Weather Data Logger

The availability of inexpensive, portable, compact and accurate data loggers has made them an ideal tool for the participants to use in the process of data collection. For our research, we considered the use of Kestrel Drop 2 data logger as the tool for collecting weather data. It can measure several weather parameters including temperature, humidity, heat index and dew point. It consists of a hermetically sealed, precision thermistor mounted externally and thermally isolated for rapid response. The thermistor is also capable of measuring temperature accurately when it is in contact with water or snow. The device can operate in the temperature ranges of -22°F and 140°F successfully with an accuracy of (\pm)0.9°F. The device is very compact and measures 1 X 1.8 X 2.4 inches and weighs just 34 grams. The Kestrel drop loggers meet international standards for water, dust and shock resistance (IP-67 and MIL-STD-810G) which makes it more user friendly for the participants. The device is powered using a button cell (CR2032) and can function for up to 6 months under ideal conditions. It is equipped with

Bluetooth Low Energy (BLE) which enables transfer of data to mobile phones and other devices compatible with BLE. Kestrel LiNK is a free mobile application that enables the data loggers to transfer data to mobile phones, view real time data and configure the device parameters like name, logging rate etc. The compatibility of the device with BLE and the battery life make sure the devices can function independently with minimal intervention of the participants. The compact size of the device makes it hassle free for the participants to carry them around with them.



Figure 2: Kestrel Drop 2 data logger

2.4 Research on UHI at the University of Georgia (UGA)

SCOUTS (Smart Community Urban Thermal Sensing Framework) is a research group at the UGA (University of Georgia) actively involved in developing methods to study and analyze urban heat islands. The goal of this research is to monitor, estimate and analyze the heat exposure to people, in urban areas, in order to mitigate human heat exposure hazards. The existing methods to study UHIs typically use data generated from satellite imagery, fixed point weather stations and aircraft borne sensors. As already discussed, there are certain limitations in using the traditional sources of data to map and analyze UHIs. SCOUTS explore the idea of

crowdsourcing the process of data collection as it helps in providing us temperatures which are more representative of the conditions experienced by the individuals and also improves the spatial granularity of the data. One of the aspects this research group focuses on, is developing a framework, which is based on dynamic sensors (human-borne, vehicle-borne, drone-borne) and measures the actual exposed temperature to the individuals. This framework also incorporates the readings from the weather stations. As part of this research, a mobile-phone based sensing framework is developed. This framework aims to assist crowdsourcing the process of data collection, thereby giving realistic insights into the heat exposure conditions individuals are exposed to.

2.5 Need for an Adaptive Sensing Framework

Although Kestrel Drop 2 data logger provides us with accurate weather data, there are some limitations in directly using this data to analyze Urban Heat Islands. Firstly, the device is not capable of providing the location data. Location data is obtained using GPS which demands significant battery power. The sensors are usually equipped with low power batteries and therefore their hardware does not support GPS. Location data is essential as we require information about the weather as well as the corresponding location coordinates to obtain spatial temporal data. Therefore, we need to find a way to include location component to the collected data to make it a spatial-temporal dataset.

The existing solution to extract data from the data logger makes use of Kestrel LiNK mobile application. The mobile application extracts the data from the sensor using BLE. This extracted data should be further transferred using methods like email or Universal Serial Bus (USB) to a central datastore where it can be analyzed. This process involves a lot of intervention by the participant which could potentially affect the quality of the dataset. Also, it involves a lot of

manual effort to merge the data collected by each participant into a single database. Therefore, we need a solution where the process of extracting the data from the sensor and transferring it to a datastore requires minimal human intervention.

Although sensors have many advantages, there are some limitations in using them. A major limitation of these devices is their limited energy resources. Due to their remote placement, it is difficult to replace the batteries from time to time. Additionally, as we are crowdsourcing the process of data collection, it is not an ideal solution to rely on participants to replace the batteries. Therefore, it is important to make sure that battery is not drained quickly and ensure longer sensor lifetimes. Since most of sensor's energy is consumed in data communication, it is important to develop a sensing methodology which reduces the number of data points being reported and simultaneously maintains the quality of dataset.

This research aims to provide solutions to these problems. We developed a robust cloud-based sensing framework, which aids in the process of data collection and requires minimum participant intervention. The framework is equipped with adaptive sensing capability which aims at reducing the data communication and thereby improving the energy efficiency of the sensors. Further chapters discuss the implementation details and the architectural overview of this solution.

CHAPTER 3

TEMPERATURE AND ACTIVITY BASED ADAPTIVE SAMPLING STRATEGY

As mentioned in previous sections, smartphones are ubiquitous mobile devices and are provided with several sensors including gyroscope, accelerometer, proximity, GPS, ambient light sensor etc. Affixing a sensor or data logger to a mobile phone gives us the opportunity to obtain and monitor dynamic information about the surrounding environment. This idea is leveraged in our research and an adaptive sensing framework is developed to efficiently monitor the environment. As part of this research, a smart and adaptive sensing framework is developed, deployed and tested extensively. Specifically, this section discusses about the need for an adaptive sampling strategy and also the implementation of adaptive sampling strategies explored as part of this research.

In our research, we develop a smartphone-based sensing framework which assists participants in crowdsourcing the process of data collection. Participants can carry the sensors and go about with their daily activities while the framework collects data at regular fixed intervals and transmits them to the cloud. Although sampling at fixed intervals provides us with valuable insights about the temperature experienced by the participants, there are two major reasons as to why this is not an efficient process. Firstly, consider the case in which the participant spends a significant period of time in a single location, usually a classroom, library or workplace (office). The variation in the temperature is not significant over time when the participant is within controlled surroundings or when the participant is static. In this case, sampling at a static

frequency will give us redundant data as the same set of temperatures can be represented fewer data points sampled at a lower frequency. Secondly, static sampling at higher frequency leads to rapid depletion of sensor's energy resources. Therefore, the sampling frequency of the sensor should be regulated in an efficient manner, thereby leading to reduced communication between the sensor and the connected central device and maintaining the quality of data points simultaneously.

In this research we implemented two adaptive sampling strategies. In the first method, we regulate the sampling frequency according to the temperature experienced by the participant. In the second method, the sensor's sampling frequency is adjusted according to the current activity of the participant.

3.1 Temperature based adaptive sampling

As the name suggests, this strategy leverages the information regarding the variation of the temperature in the subject's surrounding environment to effectively adjust the sampling frequency.

For instance, if the participant is in an environment where the temperature is changing continuously, it makes sense to sample the data at a higher frequency as it provides us with more relevant data points. On the contrary, if the participant is in a surrounding where the temperature conditions are controlled or relatively static, the conditions can be represented with fewer data points sampled over longer intervals of time. In cases like this, although sampling data at a higher frequency gives us an accurate representation of the conditions of the surrounding, it also results in accumulation of redundant data. Static sampling at a higher frequency also leads to rapid depletion in the sensor's energy resources. Therefore, sampling frequency can be adjusted

based on the participant's context in terms of the temperature experienced. This section further discusses about the implementation of this idea in our research.

The sampling frequency is broadly classified into 4 distinct categories namely base frequency, low frequency, medium frequency and high frequency. Base frequency is the default frequency at which the sensor starts transmitting data. Each of the other sampling frequencies are attributed to an appropriate range of temperature change in the surrounding environment.

Let $\{t_1, t_2, t_3, \dots, t_n\}$ be the last 'n' values of temperature observed by the sensor.

$$\{|t_2 - t_1|, |t_3 - t_2|, |t_4 - t_3|, \dots, |t_n - t_{n-1}|\}$$

This indicates the change in temperature over the last 'n' values observed by the sensor. To account for aberrant values and errors transmitted by sensor, we normalize the change in temperature by calculating the average change in temperature.

$$\{|t_2 - t_1|, |t_3 - t_2|, |t_4 - t_3|, \dots, |t_n - t_{n-1}|\}/n$$

The above value denotes the average change in the temperature observed where 'n' is the size of the window over which temperature change is considered. Application can be made more sensitive to the change in temperature by choosing a smaller value of 'n' and vice versa.

Based on the value of the average change in temperature observed over most recent readings, the system determines appropriate sampling frequency dynamically. The frequency values for different zones and the corresponding temperature difference thresholds are configurable according to the requirements on a given day or time. For instance, if a significant temperature difference is being observed or forecasted for a day, the sampling frequencies and threshold can be adjusted dynamically to obtain more quality data points.

3.2 Activity based adaptive sampling

The second strategy that we adopted is activity based adaptive sensing. The idea is to make use of the participant's current activity to determine a sampling frequency corresponding to the activity. This sampling strategy is relevant as the participant's current activity can provide inferences about the expected change in the location and mode of transport the participant is using. For instance, if the current activity of the participant is calculated to be still, it implies that the participant is stationary and an abrupt change in the spatial coordinates is not expected. In such a case, as the weather conditions of a particular location is not subjected to great variation, the temperature profile can be represented by data sampled over longer intervals of time. Similarly, when the participant is travelling in a vehicle, sampling can be performed at a higher frequency, as temperature data can be obtained over wide range of spatial coordinates. Thus, based on the activity of the participant, sampling frequency can be appropriately changed, thereby leading to more efficient use of the sensor's energy resources. Some of the activities of subject which are of interest are driving, running, walking and still. These activities provide inferences about the change in the participant's location and mode of transport based on which the sampling frequency is adjusted dynamically. For instance, if the current activity of the participant is still, a lower sampling frequency would suffice as the participant supposedly will not be exposed to highly variant weather conditions when stationary.

The on-board sensors like magnetometer, three-way accelerometer, GPS etc. in smartphones provide us with data related to the participant's location, speed, direction etc. Data gathered from these sensors can be used to predict participant's activity. This process would usually involve data collection followed by feature extraction and training of classifiers. This produces a model which can be used for detecting various meaningful patterns. Several techniques like hidden

Markov model, support vector machine are used to map these patterns to different activities. The accuracy of relying on the above method to perform activity recognition greatly depends on the developer's expertise in handling every step of the process. Not every developer often possesses these skills and therefore developing these from scratch lead to inaccurate models. Google released Activity Recognition (AR) service in 2013 to ease the development of mobile applications that require activity recognition feature. According to the documentation, [33] Google AR service is built on top of the on-board sensors available in a smartphone device. The activity recognition API detects activities by periodically reading sensor data and processing them using machine learning models.

Activity based adaptive sampling makes use of Google's AR service to efficiently predict the user's current activity. Google supports several types of activities including In Vehicle, On Foot, walking, running, still. For each activity, a specific and appropriate sampling frequency is assigned. Based on the predicted activity, the sampling frequency is updated automatically. For instance, if the user is currently sitting in an office or classroom, the activity will be predicted as still. Since the spatial coordinates of the user are constant, temperature can be collected at a reduced sampling frequency. This adaptive sampling method shows impressive results when performed over extended periods.

CHAPTER 4

OVERVIEW OF IMPORTANT COMPONENTS IN ANDROID

This section discusses about the architecture and design of the application in detail. Firstly, several important components in Android stack that play a crucial role in Android application development are discussed. Secondly, application architecture and the role played by each of these components in the functioning of our ‘UHI SCOUTS’ application is discussed.

4.1 Android Ecosystem

Android is a mobile operating system developed by Android Inc which is based on a modified version of Linux kernel and other open source software primarily designed for touchscreen mobile devices. Google acquired Android Inc in 2005. Google made the source code for Android public under Apache License 2.0 which gives developers freedom to modify and distribute it.

Android is a software stack made up of several layers of components. The Linux kernel forms the core for the operating system. There are several application programming interfaces and libraries built on top of kernel. The top level consists of applications and application framework which includes Java compatible libraries. Java is the most widely used language to develop android applications.

Android Studio is one of the popular IDE available for developing android applications. It was announced in 2013 at Google I/O conference as the official IDE for developing android applications. Due to strong online developer community support and versatility of the tool, Android Studio is used for developing this application.

In this research, more importance is given to Bluetooth Low Energy (BLE) platform, since it is the most widely used platform for communication with sensors. BLE platform is being supported since Android version 4.3 which sets the minimum requirements for the device on which this application can run.

4.2 Bluetooth on Android

Bluetooth is a wireless communication standard for connecting various digital devices. It provides a medium to exchange information between several wireless devices such as laptops, mobile phones, printers, keyboards, headsets etc. It was invented by Ericson in 1994 and was intended as a wireless replacement for cables. Currently, development of Bluetooth standards and licensing of Bluetooth technology is being carried out by Bluetooth Special Interest Group (SIG). Bluetooth works by transmitting data over radio frequencies and designed for communication over short distances ranging up to 10m or 30ft. It operates in frequencies ranging from 2.4-2.485 GHz set apart from radio, television and cellphones, and reserved for use by industrial, scientific and medical gadgets.

Bluetooth Low Energy (BLE) is a wireless communication technology developed and marketed by Bluetooth SIG. It is specifically designed to enable energy efficient communication between devices which are compact and have low power requirements usually operating for months or years. The low power consumption is achieved by sending data using smaller data packets and sending data for shorter periods. The number of channels used for communication are smaller for BLE compared to standard Bluetooth. Most modern operating systems now support BLE which is supported by Android from version 4.3 API level 18, released in July 2013. Typically, BLE is used to make applications that do not need to exchange large amounts of data. There is a wide range of mobile applications in the field of healthcare, fitness and generic sensors which are

developed by utilizing BLE profiles. BLE profiles are discussed in detail further in this document [17].

The Bluetooth protocol stack can be divided into two categories namely the host and the controller. Each of them has sub-categories, which are intended to perform specific roles. The two sub-categories under discussion are Generic Access Profile (GAP) and the Generic Attribute Profile (GATT).

GAP defines the general topology of a BLE network stack. It defines the contract as to how two BLE enabled devices can make themselves available and communicate with each other. Broadcasting and connecting are two specific mechanisms in GAP which the devices joining the BLE network can adopt.

Devices adopting the broadcasting mechanism need not explicitly connect to one another to transfer data. Broadcaster and Observer are the two distinct roles devices can adopt in this mechanism. Broadcaster is a device that broadcasts public advertising data packets. Observer is a device that listens to the data in the advertising packets sent by the broadcaster. There is no explicit connection between the broadcaster and the observer.

Devices adopting the connecting mechanism must explicitly connect and handshake to transfer data. This mechanism is more often used compared to the broadcasting mechanism. Peripheral and Central are the two distinct roles devices can adopt in this mechanism. Peripheral is a device that advertises its presence so that central devices can establish a connection. Once a connection is established, peripherals no longer broadcast data to other central devices. Central is a device that initiates a connection with a peripheral device by listening to the advertising packets. A central device can connect to many other peripheral devices. In our application, BLE devices adopt connecting mechanism to transfer the data between peripheral and central devices.

GATT layer of the BLE protocol stack handles the data communication between two connected devices. Data is passed and stored in the form of characteristics, which are stored in memory on the BLE device. When two devices are connected, they are each in either a GATT server role or GATT client role. GATT server is the device containing the characteristic database, which can be read by a GATT client. GATT client is the device that is reading or writing data from or to the GATT server.

4.3 Android Components

This section briefly explains several important components of Android framework which were essential in building the ‘UHI SCOUTS’ application used in our research. A detailed explanation of the functioning of the application is also given in this chapter.

4.3.1 Activity

An activity is a single focused thing a user can perform. Activities act as the presentation layer of an android application forming an entry point for user interaction. For example, a messaging application might have one screen that shows a list of messages and another activity to compose messages. Although an application can have different activities working together, each one is independent of others. An activity helps keep track of the current context of the application and ensures that the system keeps running the process that is hosting the activity. There are several activities in this application which enable the user to interact with application and help in entering sensor related data and displaying information about the data collected.

4.3.2 Services

In Android, a service is a component that runs in the background and does not have any interaction with the user. Services are usually used to perform long running and potentially repetitive actions. Some examples of operations where services can be used are internet

downloads, playing music, data processing etc. The two types of services are started services and bound services. Started services are launched by other application components and run indefinitely in the background even if the component that created it is destroyed. It keeps running until the service is stopped or destroyed by the Android runtime system. A bound service is bound to the launching component and unlike started service, it is possible for the launching component to interact and receive results from the bound service. The bound service gets terminated when the last bound client component unbinds itself from it. In our application, we made extensive use of services. Services help making repetitive operations like connecting to sensor, obtaining data and backing up data take place in background making it possible for the user to seamlessly work on other applications.

4.3.3 Broadcast Receivers and Broadcast Intents

Broadcast intents are used as a means of communication between different application components. Android system use them to notify applications of various system events. Some of the examples of system events include low battery notification, screen turned off, picture taken etc. Application can also initiate broadcasts to let other applications or components know about an activity or task to be completed by them. Broadcast receivers is a component that listens specific broadcast intents. Broadcast receiver must be registered with specific broadcast intents for them to start listening.

There are several broadcast intents and broadcast receivers used in our application. Some of the use cases of broadcast intent in current application are to notify when internet connection is available, to notify when data is collected etc.

4.3.4 Database

SQLite is an open source database and forms the core of Android's database support. This database is used in several environments like Apple's iOS, Blackberry and was designed to utilize low memory footprint and processing power. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements. 'android.database.sqlite' package consists of the APIs that will be needed to access the database on Android. Due to the low-level nature of these APIs, they require considerable amount of time and effort to use.

Room is a persistence library and provides an abstraction layer over SQLite to allow for efficient database access. It is a part of Android architecture components and makes it easier to work with SQLiteDatabase objects in the application by reducing the amount of complex boilerplate code to access the database and verifying SQL queries at run time. In this application, Room is utilized to access the database to store and query data efficiently.

4.3.5 Handler and Handler Thread

In Android, each application is initiated by the Android system as a new Linux process with a single thread of execution. Components belonging to the same process are executed on the same thread. The primary role of this thread is to dispatch events and to handle user interface. Performing a long running operation on the main thread causes the application to freeze and become unresponsive. To make the application more responsive, long running tasks should be executed asynchronously. Therefore, long running operations like database queries, network access or complex calculations should be executed on a new thread. Android prevents updating or accessing main thread directly from a different thread which makes communication between different threads difficult in Android. 'HandlerThread', a component of Android framework is

used for executing long running operations without making the application unresponsive and Handlers enable communication between different threads possible.

Every thread in Java is an instance of 'java.lang.Thread' class. Each thread instance can be used only once and is killed by JVM after the completion of execution. For long running and repetitive tasks, it is not an efficient solution to create a new Thread instance for every single task.

HandlerThread, inherited from java.lang.Thread package, is used to efficiently perform repetitive tasks in Android. The only significant difference between HandlerThread and Thread is that the former incorporates looper, message queue and thread. Because of message queue and a looper, a HandlerThread instance stays alive whereas a regular thread stays alive long enough to serve the task that was assigned to it. Message queue is a queue which handles incoming messages to be processed. Each Handler instance is bound to only single thread and that thread's message queue. It helps in delivering tasks to the message queue and later processing them. Looper keeps the thread alive and assigns tasks to the respective handlers for execution.

CHAPTER 5

APPLICATION ARCHITECTURE AND DESIGN

In the previous chapter we discussed in brief about the various important components of Android which are relevant to building this sensing framework. This chapter discusses about the architecture and implementation details of the application.

5.1 Application Overview

The framework consists of an android application, named UHI SCOUTS which is capable of connecting to the Kestrel data logger and extracting weather data from it at specified intervals. The application is designed in such a way that the data collection process involves minimal participant intervention. UHI SCOUTS can function in parallel with other applications on the device in a manner that it does not affect the user experience of the participant. The application does not require the device to have an active network connectivity and is capable of storing the data on the device when there is network connectivity is unavailable. Also, upon the availability of network connectivity, the application can transfer the data in a serialized manner, thereby making it possible to perform real time data analysis on the server-side. The remainder of this chapter discussed about the architecture of the application and discusses in detail the role of each component in the functioning of the application.

5.2 Application Architecture

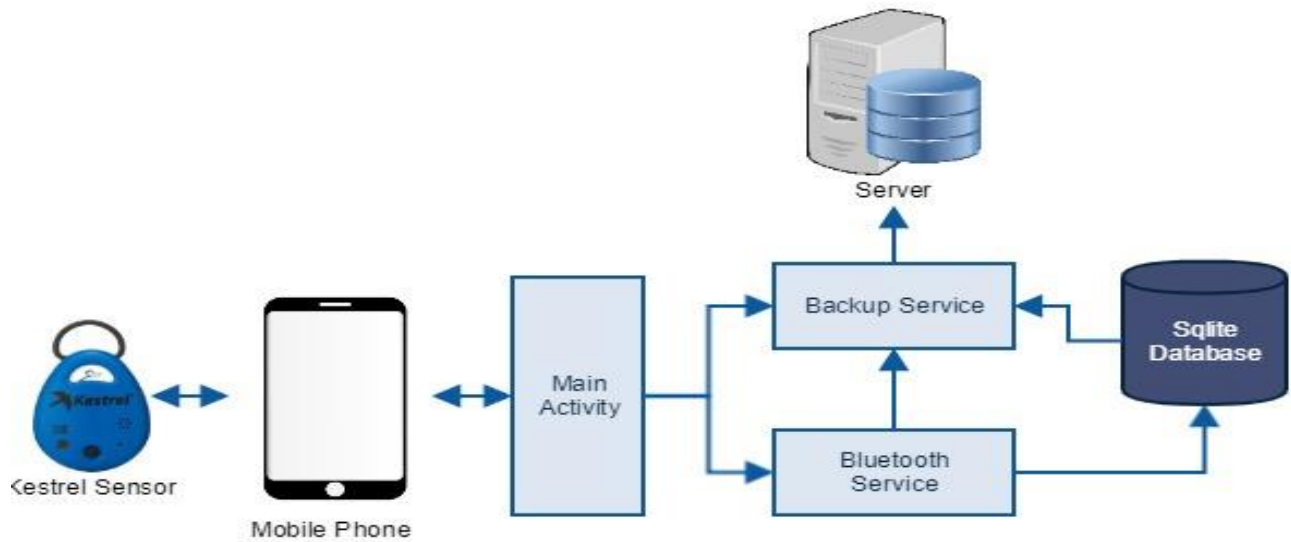


Figure 3: Overview of the architecture of ‘UHI SCOUTS’ application

The above figure gives a brief overview of the working of the application and indicates the important components that make up the architecture of the application. ‘Main Activity’ is the user-facing part of the application, which provides the user with the interface to provide details about the sensor and start the data collection process. It also provides the user with the details of the surroundings like latitude, longitude, temperature, device address of the sensor and the state of the device connection.

The application consists of two background services, Bluetooth Service and Backup Service. As discussed earlier, services help in performing long-running and computationally expensive tasks in the background enabling the user to work simultaneously with other applications during data collection.

Bluetooth Service performs the complex task of connecting to the sensor, discovery of services on the sensor device and extracting information about the weather from the sensor in the background. This service consists of a Handler Thread which helps in scheduling and executing

the tasks at given intervals. The data obtained from the sensor is stored on a local SQLite database instance and an intent is sent to the Backup Service indicating that a new record must be backed up.

Backup Service helps in transferring the data stored in the local SQLite database to the cloud. Every record in the SQLite database on the mobile phone consists of an attribute which indicates whether the record is already stored on the cloud. Based on the value of this attribute, this service identifies and schedules the records that are to be backed up. These records are ordered by their timestamp and transferred to the cloud. The data is backed up in a smart and efficient way. The data is transferred to the cloud only when network connectivity is available. In instances where network connectivity is unavailable, the data is stored on the database instance on the mobile. Every time a new data point is obtained from the sensor, an intent is sent from the Bluetooth Service to the Backup service, indicating it to transfer the record to the cloud if the network connectivity is available. In the absence of network connection, the record is stored on the device. Upon the availability of network connection, the service extracts all the records from the database that require to be stored in the cloud and transfers them to the cloud ordered by the timestamp they are collected.

The server is a Java servlet application is hosted on a Google Cloud Compute Engine instance. The server application exposes a set of endpoints which can be accessed by the mobile application to obtain several details which are important to the functioning of the application. These endpoints help the application obtain details like the address of a sensor device and help in storing data collected by the application. The server makes use of MySQL database instance to store the data persistently.

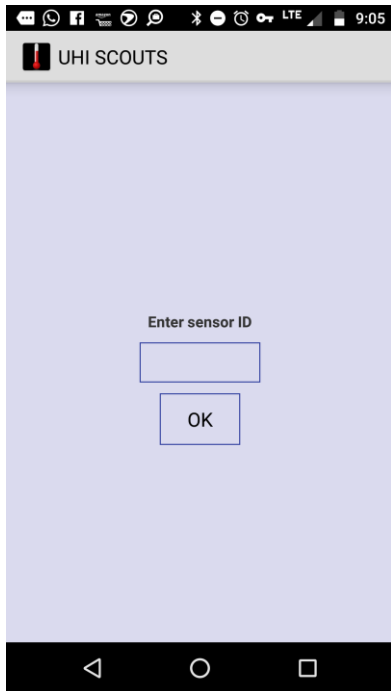


Figure 4: Screenshot showing the view where the participant can enter sensor details

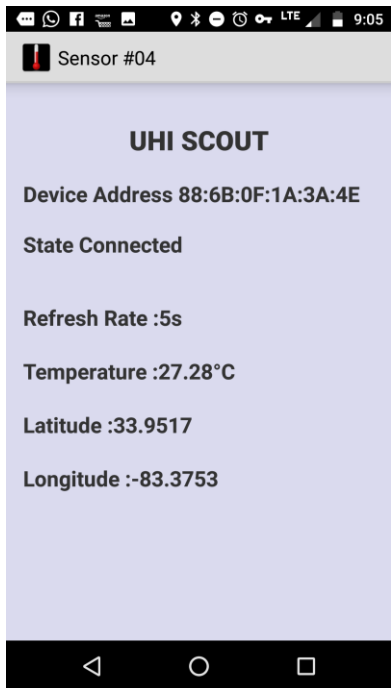


Figure 5: Screenshot showing the view where data extracted from sensor is displayed

Figure 4 and Figure 5 shows us the different views of the application. As mentioned, once the participant enters the information about the sensor, the application connects to the sensor and starts the process of data collection. The application detects the availability of network connectivity and appropriately transfers the data without any intervention from the participant. 'UHI SCOUTS' has been tested thoroughly on multiple mobile phone platforms including LG, Nexus, Samsung and Motorola. Some interesting conditions under which the application was tested include ensuring the application gracefully stops data collection when the Bluetooth connectivity is lost abruptly, ensuring the application starts the data collection upon availability of Bluetooth connectivity, ensuring that the data is stored persistently on the phone when the network connectivity is unavailable, ensuring that the data transfer is initiated automatically upon the availability of network connectivity and that data is transferred to the cloud in a serialized manner etc. 'UHI SCOUTS' is available to public download on Google Play Store for android devices and Apple Store for iOS devices.

CHAPTER 6

EXPERIMENTS AND RESULTS

This chapter discusses about the experiments performed to illustrate the results obtained from the implementation of the adaptive sampling strategies. Several experiments were performed as part of this research to test and validate the activity as well as temperature based adaptive sampling strategies. The goal of these experiments is to demonstrate the results derived from the implementation of the adaptive sampling strategies in terms of the amount of data transferred and compare them with data obtained from static sampling. To elaborate further, major part of sensor's energy is consumed in transferring the data to the central device [34]. Through our experiments, we want to illustrate the impact of adaptive sampling strategies in reducing the amount of data transferred and thereby leading to reduction in the consumption of the energy sources of the sensor as well as the smartphone.

6.1 Experimental Setup

As part of this research, two Android applications titled 'UHI SCOUTS' and 'Smart UHI' were developed. 'UHI SCOUTS' is now available on Google application store and is being used by participants for crowdsourcing weather data. 'Smart UHI' was developed exclusively for this research and it incorporates activity and temperature based adaptive sampling strategies. Both the applications communicate with Kestrel weather data logger to obtain weather data and are used extensively to carry out experiments in this research.

As discussed before, each sampling strategy adopts a specific sampling frequency based on the change in the environment. In the case of activity based adaptive sampling, each activity is assigned a predefined sampling frequency and frequency is adjusted based on the activity of the subject. The activity of the participant is interpreted based on the data obtained from the multiple sensors embedded in the mobile phone. Google's Activity Recognition API [35] is built on top of the sensors embedded in the mobile phone. Activity Recognition API periodically reads data from the sensors and processes them using machine learning models. There are four activities we considered in our research and each of them is assigned a specific sampling frequency. Our framework can be extended to include other types of activities if needed. The activities that are considered are as follows

- On Vehicle: s_1
- Running: s_2
- Walking: s_3
- Still: s_4

where $s_1=5s$, $s_2=15s$, $s_3=30s$, $s_4=60s$ are predefined frequencies for each of the activities. The idea is that when the subject is in a vehicle, the variation in the temperature could be more predominant and therefore more frequent sampling will help get relevant data points compared to when the subject is still.

Similarly, in the case of temperature based adaptive sampling, the mean of the difference in the temperature values, m , is divided into three categories. They are as follows

- Low sampling frequency: s_1 $m < a_1$
- Medium sampling frequency: s_2 $a_1 \leq m < a_2$
- High sampling frequency: s_3 $a_3 \leq m$

where $s_1=30s$, $s_2=15s$, $s_3=5s$ and $a_1=0.25$, $a_2=0.5$, $a_3=0.75$.

The traditional sampling strategy adopted by ‘UHI Scouts’ application performs sampling at a static frequency irrespective of the participant’s activity or the temperature the participant is being exposed to. This essentially means that $s_1=s_2=s_3=s_4$ or the ratio $s_1: s_2: s_3: s_4=1$. In the experiments conducted as part of this research, $s_1=s_2=s_3=s_4=5s$ for traditional sampling strategy.

Experiments conducted as part of this research can be broadly classified into two different categories. The first category of experiments illustrate the performance of adaptive sampling strategies by comparing them with traditional sampling strategies. These experiments were typically carried out over a span of 4 to 5 hours. These experiments were useful in showing the accuracy of the adaptive sampling strategies when applied over longer periods of time as it is more representative of the actual data sampling process that will be carried out by the participants. The second category of experiments were conducted to evaluate the accuracy of adaptive sampling strategies. Root Mean Square Error (RMSE) is used as measure of similarity between data obtained using adaptive sampling strategies and data obtained using traditional sampling strategies.

6.2 Experimental Dataset

The first set of experiments were conducted to illustrate and analyze the performance of adaptive sampling strategies over longer periods of time. Each experiment involved collecting two different data sets. This is achieved by collecting temperature data using two mobile applications, ‘UHI SCOUTS’ and ‘Smart UHI’ that are developed for this research and two Kestrel Drop 2 data loggers. ‘UHI SCOUTS’ employs traditional sampling strategy for data collection and is used in generating a temperature dataset by sampling data at a static sampling frequency. For all the datasets generated by using traditional sampling strategy in this research,

the sampling frequency is 5 seconds. ‘Smart UHI’ employs adaptive sampling strategies and is used in generating a temperature dataset by sampling adaptively using a predefined set of sampling frequencies. For temperature based adaptive sampling, Table 1 represents the set of sampling frequencies used for temperature based adaptive sampling Table 2 represents the predefined set of sampling frequencies that are assigned for various activities in activity based adaptive sampling.

Table 1: Predefined sampling frequencies for temperature based adaptive sampling

Id No	Base	Low Frequency (s_1)	Medium Frequency (s_2)	High Frequency (s_3)
1	5s	30s	15s	5s

Table 2: Predefined sampling frequencies for activity based adaptive sampling

Id No	Base	On Vehicle (s_1)	Running (s_2)	Walking (s_3)	Still (s_4)
1	5s	5s	15s	30s	60s

The second set of experiments were conducted to evaluate the accuracy of the adaptive sampling strategies. Several experiments were conducted by varying the ratio of sampling frequencies, $s_1:s_2:s_3:s_4$ for activity based adaptive sampling, $s_1:s_2:s_3$ for temperature based adaptive sampling and comparing how the accuracy varies for these ratios Each experiment involved collecting three different datasets. They are

- 1) Data collected using traditional sampling approach
- 2) Data collected using adaptive sampling approach with predefined sampling frequencies.
- 3) Data collected using adaptive sampling approach with variable sampling frequencies. comparing two different datasets.

The above datasets are collected during the same time interval by a single participant. For these experiments one instance of ‘UHI SCOUTS’ and two instances of ‘Smart UHI’ are used to connect to three different sensors. This is to account for errors due to external factors like differing temperature values during difference in time and location. In most of the Android based mobile phones, each device can support 6 concurrent Bluetooth connections.

6.3 Experimental Results

As mentioned before, several experiments were conducted to analyze the efficiency and accuracy of the adaptive sampling strategies. In each of the experiments, the participant attached the sensor to the bag or carried it by hand. Each experiment was carried out for a period of 5-6 hours, typically comprising the active part of the day. The participant carried out several common activities like commuting to workplace, moving in the city, moving within a building etc. during this time. These experiments illustrate the performance of adaptive sampling strategies over longer periods of time.

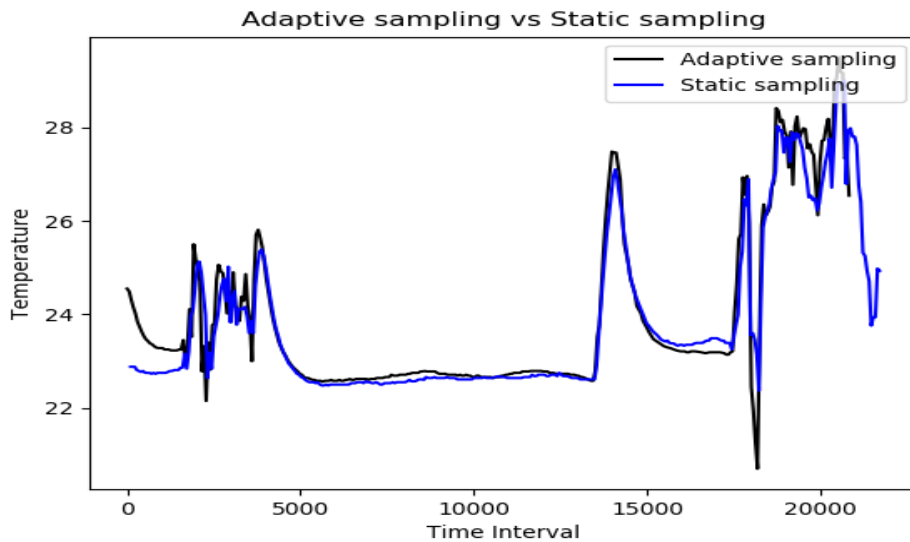


Figure 6: Activity based adaptive sampling Experiment 1

Table 3: Comparison between data collected by activity based adaptive sampling and static sampling for Experiment 1

Sampling Strategy	Data Points	Data Transferred
Activity Based Sampling	881	104 kB
Static Sampling	4000	476 kB

The above experiment illustrates activity based adaptive sampling. The x-axis represents time in seconds and y-axis represents temperature in Celsius scale. As we can see, the temperature profile generated by adaptive sampling and static sampling are similar. As for the data transferred during this experiment, data collected using adaptive is 78.15% lesser than data collected using static sampling.

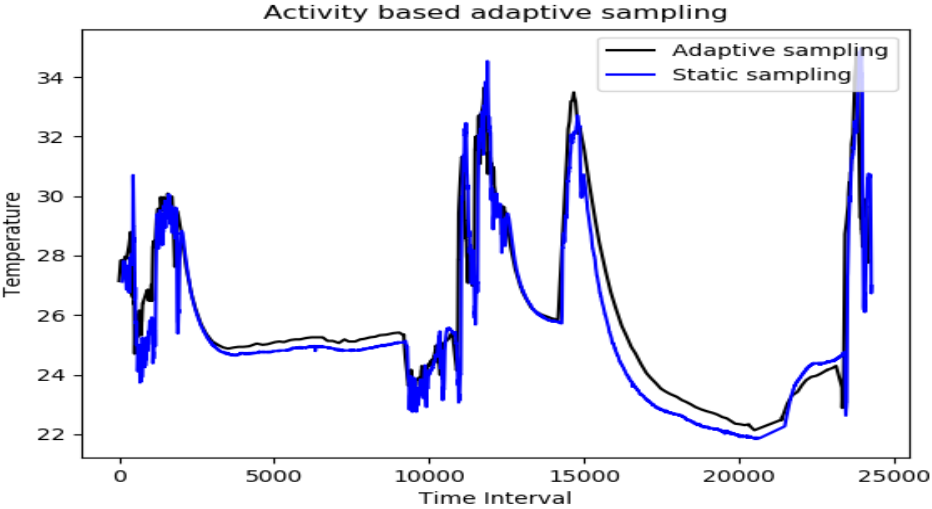


Figure 7: Activity based adaptive sampling Experiment 2

Table 4: Comparison between data collected by activity based adaptive sampling and static sampling for Experiment 2

Sampling Strategy	Data Points	Data Transferred
Activity Based Sampling	740	88.06 kB
Static Sampling	4070	488.33 kB

The above experiment illustrates activity based adaptive sampling. The x-axis represents time in seconds and y-axis represents temperature in Celsius scale. As we can see, the temperature profile generated by adaptive sampling and static sampling are similar. As for the data transferred during this experiment, data collected using adaptive is 81.96% lesser than data collected using static sampling.

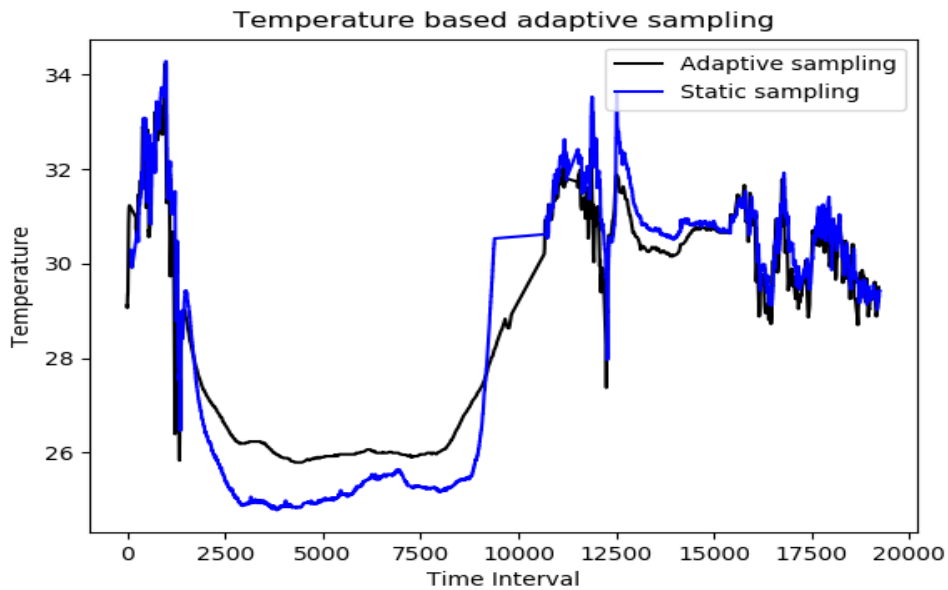


Figure 8: Temperature based adaptive sampling Experiment 1

Table 5: Comparison between data collected by temperature based adaptive sampling and static sampling for Experiment 1

Sampling Strategy	Data Points	Data Transferred
Activity Based Sampling	890	105.91 kB
Static Sampling	3073	365.687 kB

The above experiment illustrates temperature based adaptive sampling. The x-axis represents time in seconds and y-axis represents temperature in Celsius scale. As we can see, the temperature profile generated by adaptive sampling and static sampling are similar. As for the data transferred during this experiment, data collected using adaptive sampling is 71.03% lesser than data collected using static sampling.

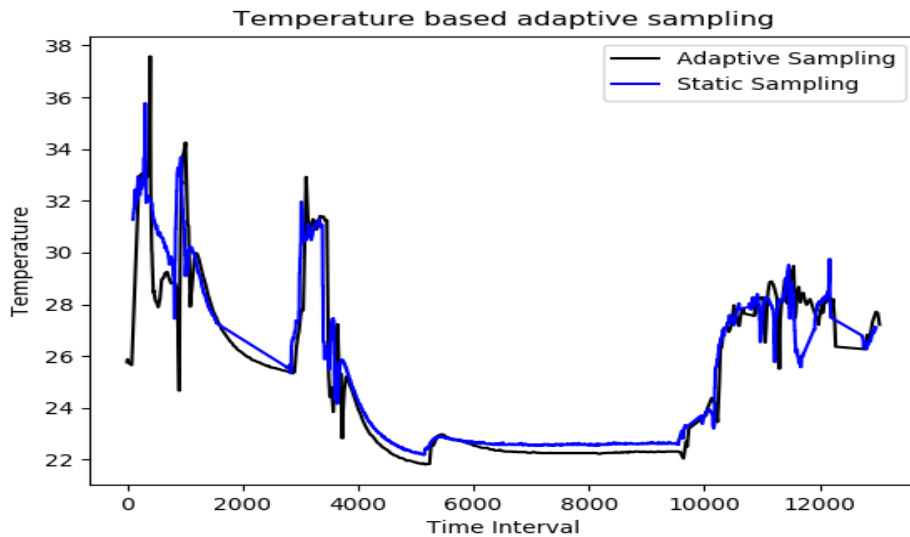


Figure 9: Temperature based adaptive sampling Experiment 2

Table 6: Comparison between data collected by temperature based adaptive sampling and static sampling for Experiment 2

Sampling Strategy	Data Points	Data Transferred
Activity Based Sampling	499	59.38 kB
Static Sampling	2043	243.1 kB

The above experiment illustrates temperature based adaptive sampling. The x-axis represents time in seconds and y-axis represents temperature in Celsius scale. As we can see, the temperature profile generated by adaptive sampling and static sampling are similar. As for the data transferred during this experiment, data collected using adaptive sampling is 75.72% lesser than data collected using static sampling.

Table 7: Sampling frequencies used to evaluate temperature based adaptive sampling

Ratio	Base	Low Frequency	Medium Frequency	High Frequency
1	5	5	5	5
2	5	20	10	5
3	5	45	15	5
4	5	80	20	5

Table 8: Sampling frequencies used to evaluate activity based adaptive sampling

Ratio	Base	Vehicle	Running	Walking	Still
1	5	5	15	30	60
2	5	5	10	20	40
3	5	5	15	45	135
4	5	5	20	80	320

Table 5 and Table 6 indicate the sampling frequencies used for conducting several experiments to evaluate the preciseness of adaptive sampling strategies. As mentioned before, ratios of

sampling frequencies $s_1:s_2:s_3:s_4$ is varied and the accuracy of the data collected is analyzed. The ratios used for temperature and activity based adaptive sampling are mentioned in the above tables. Predefined sampling frequencies are a chosen set of frequencies which are used as a benchmark to analyze adaptive sampling strategies across varying ratios of frequencies.

Root mean square error (RMSE) is used as a measure of similarity between data collected using traditional sampling strategy and data collected using adaptive sampling strategy. RMSE is calculated between data collected using traditional sampling approach and dataset collected using adaptive sampling approach with predefined sampling frequencies to observe the similarity between two datasets. RMSE is also calculated between dataset collected using traditional sampling approach and dataset collected using adaptive sampling approach with varying ratios of sampling frequencies to observe the change in RMSE value with the change in frequency ratios. These two RMSE values are compared with each other for every experiment to analyze the preciseness of adaptive sampling strategies.

RMSE is calculated using two methods. The first method interpolates every data point obtained using traditional sampling on to the graph of the data points obtained using adaptive sampling. The modulus of the difference of distance between the interpolated point and the corresponding point on the traditional sampling graph is used to calculate RMSE. The second method makes use of nearest value to calculate RMSE i.e. for every point obtained using adaptive sampling, the data point with immediate lowest value is taken from the set of data points obtained using traditional sampling. The difference between these two values is used to calculate RMSE. For simplicity purposes, we will call RMSE calculated using interpolation as Standard RMSE and RMSE calculated using nearest value as Nearest RMSE. The number of data points collected and

data transferred is also used as a metric to compare the efficiency and accuracy of adaptive sampling strategies.

Two sets of experiments were conducted to illustrate the accuracy of activity based adaptive sampling and one set of experiments were conducted to illustrate the accuracy of temperature based adaptive sampling. Each set of experiments involves collecting the data for all the ratios of sampling frequencies as mentioned in Table 6 and Table 7.

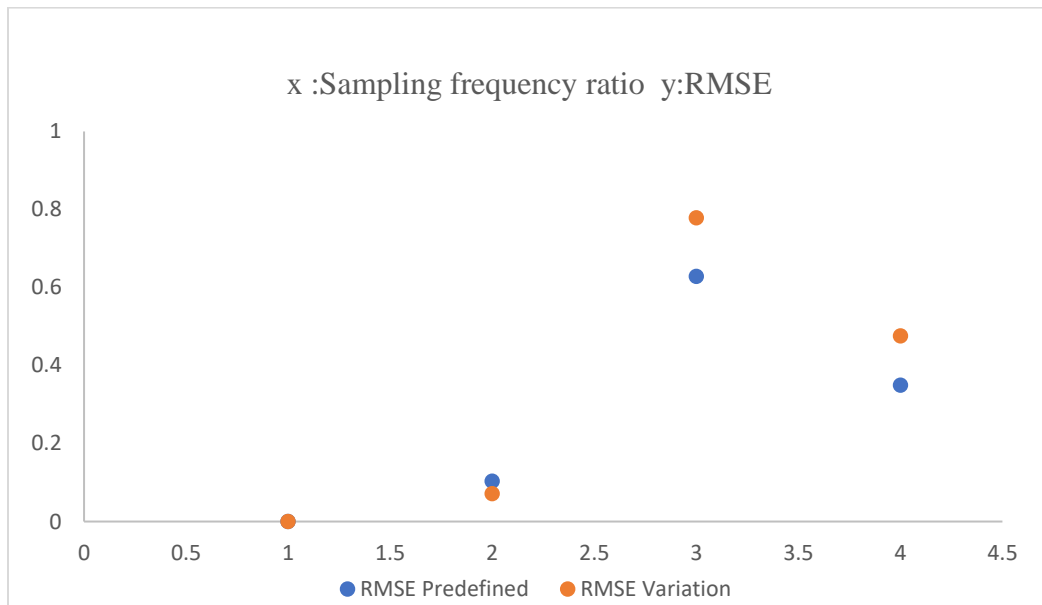


Figure 10: Standard RMSE Comparison for Activity Based Adaptive Sampling for Experiment Series 1

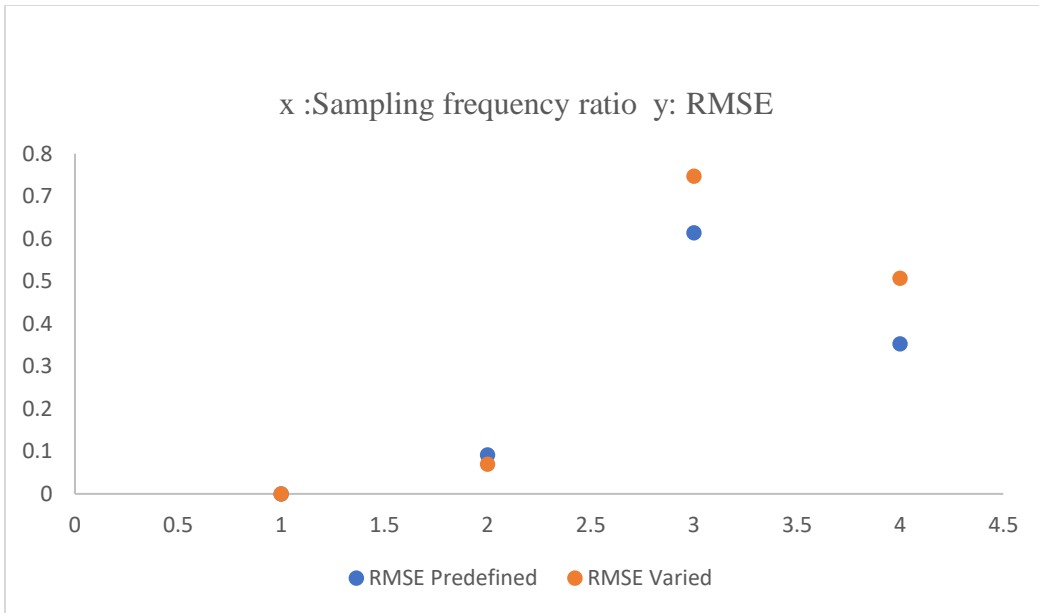


Figure 11: Nearest RMSE Comparison for Activity Based Adaptive Sampling for Experiment Series 1

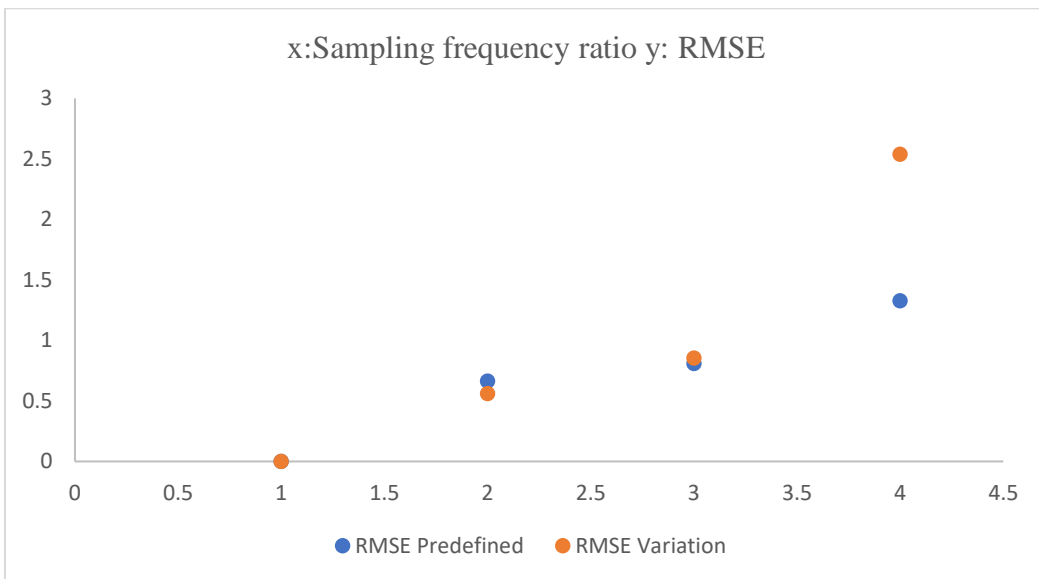


Figure 12: Standard RMSE Comparison for Activity Based Adaptive Sampling for Experiment Series 2

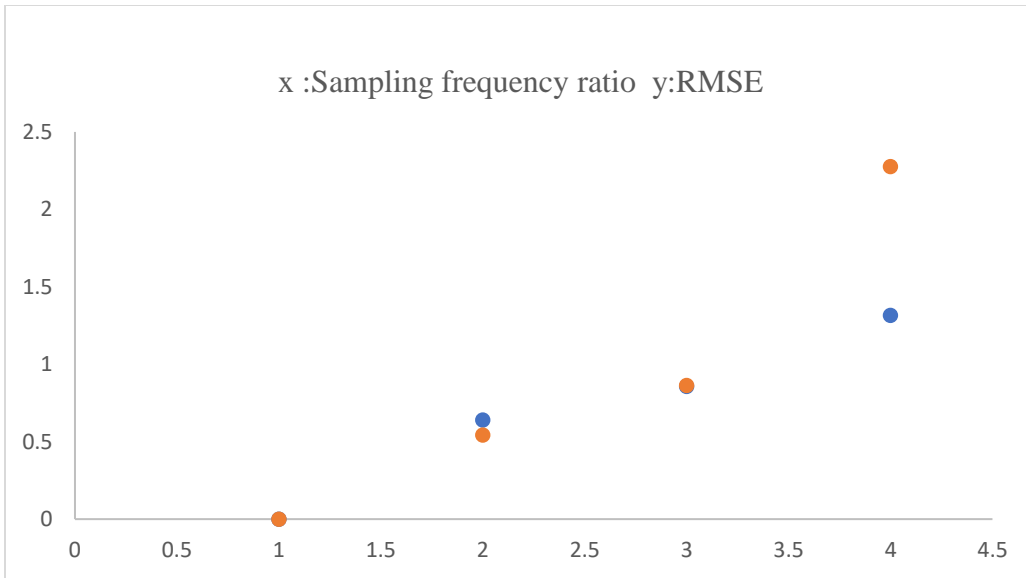


Figure 13: Nearest RMSE Comparison for Activity Based Adaptive Sampling for Experiment Series 2

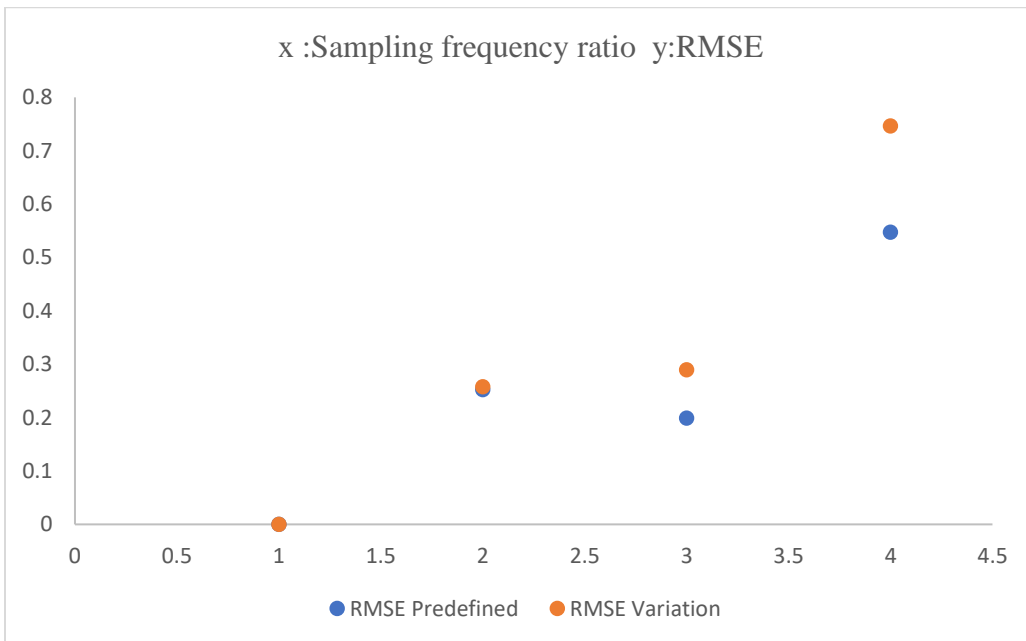


Figure 14: Nearest RMSE comparison for Temperature Based Adaptive Sampling

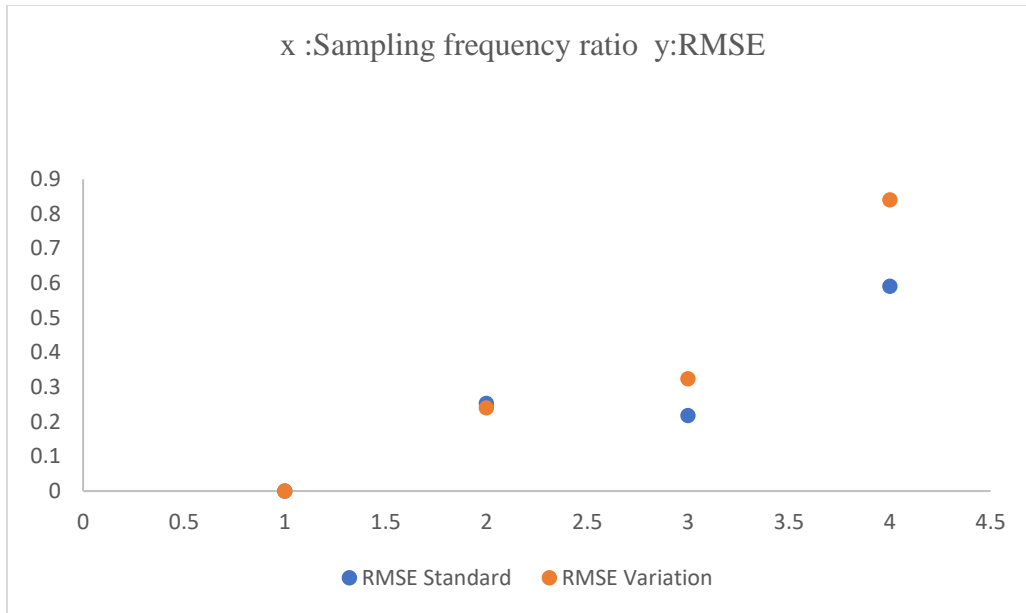


Figure 15: Standard RMSE comparison for Temperature Based Adaptive Sampling

The above graphs indicate the change in the values of RMSE with the change in ratios of sampling frequency. The x-axis represents the ratio of the sampling frequencies as mentioned in Table 5 and Table 6. The y-axis represents the RMSE between the dataset obtained using adaptive sampling and dataset obtained using static sampling. For each adaptive sampling approach, graphs are plotted using Standard RMSE and Nearest RMSE as measure of error. When the ratio is 1, $s_1=s_2=s_3=s_4=5s$ for activity based adaptive sampling and $s_1=s_2=s_3=5s$ for temperature based adaptive sampling. With same sampling frequencies, adaptive sampling and traditional sampling produces similar data-sets leading to a RMSE value of zero. It is expected that with the increase in the ratio of sampling frequencies, the number of data points obtained by adaptive sampling approach decreases. This explains the reason for the increase in the difference of RMSE values with increase in ratio of sampling frequencies, thereby indicating the preciseness of adaptive sampling.

6.4 Combining Activity Based Adaptive Sampling And Temperature Based Adaptive Sampling

Based on the experiments conducted to illustrate activity based adaptive sampling and temperature based adaptive sampling, there are some interesting observations that were made. Data collected by activity based adaptive sampling is more reflective of the data collected by static sampling when the participant is performing activities like walking, running, still etc. However when the participant is performing activities like walking, activity based adaptive sampling fails to capture some important aspects. For instance, if the participant is walking out or into a building, there is going to be an abrupt change in temperature. Activity based adaptive sampling identifies events like this based on the activity of the participant and therefore data collected during this time does not reflect the actual change in temperature experienced by the participant. Similarly, when the sensor is mounted on a vehicle, activity based adaptive sampling samples the data appropriately at high frequency when the vehicle is in motion. But when the vehicle is stationary, data is sampled at a lower frequency as the activity is classified as still. Due to the automobile exhaust and heating up of roads and pavements, participants are exposed to excessive heat conditions and sampling at a lower frequency fail to capture these events. On the other hand, temperature based adaptive sampling accurately reflects these events as the data is sampled at a higher frequency due to the temperature change observed. Therefore, a system which incorporates both activity and temperature based adaptive sampling strategies can more accurately reflect the actual temperature experienced the participant.

CHAPTER 7

CONCLUSION

Analyzing and quantifying the effects of UHIs can help in reducing health hazards due to heat waves and lead to better understanding the magnitude of heat exposure at an individual level. Data obtained from crowdsensing combined with data obtained from traditional sources like satellite imagery, air borne sensors, weather stations etc. leads to highly granular data sets which can help in better analyzing and studying Urban Heat Islands. However, there are no existing crowdsensing frameworks for obtaining spatial-temporal weather data. As part of this research, a robust cloud-based sensing framework is developed. With the help of modern smartphones and weather sensors, a crowdsensing system is developed which can collect and store spatial-temporal weather data. Although sampling with a static frequency produces granular data set, it leads to rapid depletion of sensor's energy sources and in most cases static aggressive sampling leads to redundant data. This thesis proposes activity and temperature based adaptive sampling strategies which reduces the number of data points being collected while maintaining accuracy. The sliding window approach is used to calculate temperature fluctuation among recently collected data. To adapt to this variation in temperature, the system assumes an appropriate sampling frequency. Activity based adaptive sampling adjusts the sampling based on the current activity of the participant. Root mean square error is used as a measure to determine the accuracy of adaptive sampling strategies. These strategies were tested extensively to analyze their performance over long periods of time and under varying environmental conditions. To validate

the preciseness of our approach, we varied the ratio of frequencies and analyzed the corresponding change in the root mean square error values. Our system could efficiently reduce the redundant data points without affecting the quality of the data collected.

REFERENCES

1. Yang, L., Qian, F., Song, D., & Zheng, K. (2016). Research on Urban Heat-Island Effect. *Procedia Engineering*, 169, 11-18. doi:10.1016/j.proeng.2016.10.002
2. Fabrizi, R., Santis, A. D., & Gomez, A. (2011). Satellite and ground-based sensors for the Urban Heat Island analysis in the city of Madrid. 2011 Joint Urban Remote Sensing Event. doi:10.1109/jurse.2011.5764791
3. Udristoiu, M. T., Velea, L., Bojariu, R., & Sararu, S. C. (2017). Assessment of urban heat Island for Craiova from satellite-based LST. doi:10.1063/1.5017443
4. Fabrizi R., Bonafoni S., Biondi R. Satellite and Ground-Based Sensors for the Urban Heat Island Analysis in the City of Rome. *Remote Sensing*. 2010, 2(5):1400-1415.
5. Price, J.C. Assessment of the urban heat island effect through the use of satellite data. *Monthly Weather Review* 1979, 107, 1554-1557.
6. Crowdsensing Brabham, D. C. (2008). Crowdsourcing as a Model for Problem Solving. *Convergence: The International Journal of Research into New Media Technologies*, 14(1), 75-90. doi:10.1177/1354856507084420
7. Muller, C.I., et al. "Crowdsourcing for Climate and Atmospheric Sciences: Current Status and Future Potential." *International Journal of Climatology*, vol. 35, no. 11, May 2015, pp. 3185–3203., doi:10.1002/joc.4210.
8. "NoiseTube." NoiseTube, www.noisetube.net/.

9. Scott, M. D., Ramaswamy, L., & Lawson, V. (2016). CyanoTRACKER: A Citizen Science Project for Reporting Harmful Algal Blooms. 2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC). doi:10.1109/cic.2016.058
10. Urban Crowd Sensing Demonstrator: Sense the Zagreb Air Aleksandar Antonic, Vedran Bilas, Martina Marjanovi ć, Maja Matija Ćevi ć, Dinko Oleti ć, Marko Pavelic, Ivana Podnar Ćzarko, Kre Ćimir Pripu Ćzi ć and Lea Skorin-Kapov
11. Overeem, A., et al., *Crowdsourcing urban air temperatures from smartphone battery temperatures*. 2013. 40(15): p. 4081-4085.
12. D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, vol. 1, no. 1, January 2002.
13. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of ACM WSNA*, 2002.
14. Cambra, C., Díaz, J.R., Lloret, J., 2015. Deployment and performance study of an Ad Hoc network protocol for intelligent video sensing in precision agriculture. In: *Proceedings of Ad-Hoc Networks and Wireless. INCS*, vol. 8629. Springer, Berlin Heidelberg, pp. 165–175.
15. E. Hussain, G. Chow, V. C. M. Leung, R. D. McLeod, J. Misic, V. W. S. Wong, and O. Yang, "Vehicular telematics over heterogeneous wireless networks: A survey," *Computer Communications*, vol. 33, pp. 775-793, May 2010.
16. Huo, Hongwei, et al., "An Elderly Health Care System Using Wireless Sensor Networks at Home", 2009 Third International Conference on Sensor Technologies and Applications.

17. R. Tan, G. Xing, J. Wang, and H. C. So, "Collaborative target detection in wireless sensor networks with reactive mobility," 16th Intl. Workshop on Quality of Service, pp. 150-159, 2008.
18. J. Wang, X. Ren, Y. Shen, and S. Liu, "A remote wireless sensor network for water quality monitoring," Intl. Conf. on Innovative Computing and Communication, pp.7-12, 2010.
19. I.F.Akyildiz, W. Su, Y. Sankarasubramaniam E. Capirci, "Wireless Sensor Networks: a Survey", Computer Networks, Vol 38, N. 4, March 2002.
20. A. D. Marbini and L. E. Sacks. Adaptive sampling mechanisms in sensor networks. In London Communications Symposium, London, UK, 2003.
21. An Energy Efficient Adaptive Sampling Algorithm in a Sensor Network for Automated Water Quality Monitoring Tongxin Shu 1,* , Min Xia 2 , Jiahong Chen 2 and Clarence de Silva 2
22. I. Lazaridis, Q. Han, X. Yu, S. Mehrotra, N. Venkatasubramanian, D. V. Kalashnikov, and W. Yang. QUASAR: Quality aware sensing architecture. ACM SIGMOD Record, 33(1):26–31, Mar. 2004.
23. Ganti, Raghu, et al. "Mobile Crowdsensing: Current State and Future Challenges." IEEE Communications Magazine, vol. 49, no. 11, 2011, pp. 32–39.,
doi:10.1109/mcom.2011.6069707
24. "World Urbanization Prospects - Population Division." United Nations, United Nations, esa.un.org/unpd/wup/.

25. Xian, George, and Mike Crane. "Assessments of Urban Growth in the Tampa Bay Watershed Using Remote Sensing Data." *Remote Sensing of Environment*, vol. 97, no. 2, 2005, pp. 203–215., doi:10.1016/j.rse.2005.04.017.
26. Tokairin T, Sofyan A, Kitada T. Numerical study on temperature variation in the Jakarta area due to urbanization. The seventh international conference on urban climate; Yokohama, Japan: Toyohashi University of Technology, Institute of Technology; 2009.
27. "Urban Heat Island Introduction." Metlink Teaching Weather and Climate, www.metlink.org/other-weather/urban-heat-islands/urban-heat-island-background/.
28. Changnon, Stanley A., et al. "Impacts and Responses to the 1995 Heat Wave: A Call to Action." *Bulletin of the American Meteorological Society*, vol. 77, no. 7, 1996, pp. 1497–1506., doi:10.1175/1520-0477(1996)077<1497:iarth>2.0.co;2.
29. Tran, Hung, et al. "Assessment with Satellite Data of the Urban Heat Island Effects in Asian Mega Cities." *International Journal of Applied Earth Observation and Geoinformation*, vol. 8, no. 1, 2006, pp. 34–48., doi:10.1016/j.jag.2005.05.003.
30. "Heat Island Impacts." EPA, Environmental Protection Agency, 20 June 2017, www.epa.gov/heat-islands/heat-island-impacts.
31. Mallick J, Kant Y, Bharat BD (2008) Estimation of land surface temperature over Delhi using Landsat-7 ETM+. *J Ind Geophys Un* 12:131-140
32. Rasul, Azad, et al. "A Review on Remote Sensing of Urban Heat and Cool Islands." *Land*, vol. 6, no. 2, Sept. 2017, p. 38., doi:10.3390/land6020038.

33. “The LightBlue Bean Is a Legacy Product and Is No Longer Being Sold.” Punch Through, www.punchthrough.com/products/bean/docs/guides/everything-else/how-gap-and-gatt-work/
34. Min, Rex, and Anantha Chandrakasan. “Energy-Efficient Communication for High Density Networks.” *Ambient Intelligence: Impact on Embedded System Design*, 2003, pp. 295–314., doi:10.1007/0-306-48706-3_15.
35. Activity Recognition API | Google Developers. (n.d.). Retrieved from <https://developers.google.com/location-context/activity-recognition/s>