Achieving Healthy and Quality Life of One-person Households Using IoT and Machine Learning

Niu, Long

(Degree)
博士（計算科学）
(Date of Degree)
2019-03-25
(Date of Publication)
2020-03-01
(Resource Type)
doctoral thesis
(Report Number)
甲第7520号
(URL)
https://hdl.handle.net/20.500.14094/D1007520

Doctoral Dissertation

Achieving Healthy and Quality Life of One-person Households Using IoT and Machine Learning

（IoTと機械学習を用いた独居者の
健康な生活の維持・向上に関する研究）

平成31年1月

Department of Computational Science
Graduate School of System Informatics
Kobe University

Name　Niu Long

Achieving Healthy and Quality Life of One-person Households Using IoT and Machine Learning

Niu Long

Abstract

Worldwide, there has been an increase in the number of individuals that live alone in *one-person households* (OPHs). Compared to those living with family, people in OPHs easily lose control of *life rhythm*. Given that the disturbance of life rhythm leads to chronic disease, they have a higher risk of illness. As such, there is an urgent demand for assistive technology that allows people in OPHs to enjoy healthy, high-quality lives.

For decades, there has been significant research and development of smart systems to assist people at home. However, there are still limitations on the practical use of these systems in actual OPHs. More specifically, they are often too intrusive to the lifestyle of users or home objects. In addition, they are often expensive to deploy and maintain. Furthermore, these systems are unable to evaluate the quality of life rhythm. As a result, it is difficult for individual users to determine what their healthy life rhythms should be, and how to improve their current situation.

The goal of my research is to develop a *new smart system for OPHs* that can minimize intrusiveness and cost, while also facilitating the assessment of life rhythms of individual users. The new system collects user position and environmental data inside the house in a non-intrusive way, using affordable IoT devices. From this data, the system then recognizes the daily activities of the user. Based on these activities, eventually, the system can quantitatively evaluate the user's life rhythms and provide practical advice for maintaining a healthy life.

In this dissertation, we address three technical challenges associated with the implementation of the proposed system. The first challenge is the collection of the user's indoor position and environmental data in OPH while minimizing intru-

siveness and cost. For the indoor position data, we propose a common data model for indoor location *DM4InL* and a Web-based integration framework *WIF4InL*, in order to reuse and integrate heterogeneous indoor positioning systems (IPS). For the implementation of an affordable IPS, we develop *BluePIN*, a zone-based positioning system with Bluetooth low energy (BLE) beacons. With respect to the environmental data, we exploit *Autonomous SensorBox*, which is a self-managed IoT for environment sensing that was developed in our laboratory.

The second challenge is the accurate recognition of the user's daily activities. For this, we propose a daily activity recognition method based on *supervised machine learning*. In the training phase, the proposed method asks the user to record their activities manually, using a lifelog tool. The activities of interests are sleeping, eating, bathing, cooking, PC working, cleaning, and going out. While the system accumulates the environmental and indoor position data, the proposed method generates *training data* by attaching the recorded activities as *labels* to the time-series sensing data. By applying machine learning of multi-class classification to the training data, the proposed method derives an activity recognition model. Once the model is obtained, the system moves to the operation phase where the seven types of activities are automatically recognized. In the experiment, we evaluate various configurations of learning algorithms and features engineering.

The third challenge is the quantitative assessment of an individual's life rhythm. For this, we propose an approach that derives a *personalized assessment model* based on the recognized daily activities and user self-assessment of quality of life (QoL). The proposed method characterizes the user's life rhythms based on statistical features of their daily activities, especially *eating* and *sleeping*. In addition, the method periodically requests that the user evaluates the degree of their QoL using a designated questionnaire survey. The method then establishes a *regression model* that explains the QoL based on the statistics of daily activities. Using this model, the user can better understand the characteristics of their current life rhythm. Moreover, the model can be used to produce personal advice on daily habits to maintain the user's healthy life rhythm.

Keywords Indoor Positioning System, Daily Activity Recognition, One-Person Household, IoT, Non-intrusive Environment Sensing, Assessment of Life Rhythm, Smart Home, Healthcare System

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Increasing rate of OPHs and issues of people in OPHs

Over the past several decades, due to global aging, the increasing number of unmarried people, and marriages late in life, we have witnessed high rates of one-person households (OPHs) across many countries. Among European countries, OPHs of 40% or more were reported in Denmark, Finland, Germany, and Norway in 2015. In the USA, in approximately seven states, the percentage of OPHs exceeded 30.3% in 2015. In China, there are more than 60 million people currently living alone and the number of OPHs is predicted to increase to 162 million in 2050. In Japan, 37.4% of all households will become OPHs in 2030.

There are several issues that affect people in OPHs. Given that there is no one else to provide assistive care on a daily basis, it is more difficult for these individuals to maintain healthy *life rhythm*. According to some health studies, it is more likely that individuals in OPHs will fail to manage life rhythm compared to those living with family or others. The research [1] shows that students living with families wake up and go to bed earlier than those living alone. With respect to the total number of meals and skipping breakfast, people in OPHs have a significantly higher rate [2]. It is well-established that a chaotic life rhythm often leads to a deterioration of health [3]. The results of research on life rhythm and the circadian rhythm (dian means day) show that a key factor for achieving a good *health-related quality of life* (HRQoL) is to maintain a healthy life rhythm. For instance, individuals with circadian rhythm disturbance have a higher risk of cardiovascular disease [4]. For example, sleep disturbance increases the risk of suffering from neutral fat [2]. As a result, individuals living in OPHs have a significantly higher risk of suffering from certain disease, which has been well-established in numerous health studies. Therefore, there is an increasing demand for the development of assistive technology for people in OPHs that can help to improve their live health and quality life.

1.2 Overview of assistive technologies using IoT

Recently, due to the proliferation of smartphones and the Internet of Things (IoT) technologies, there have been numerous studies and the development of applications that aim to monitor and improve a user's health by recording daily activities log (e.g., sleep, eat, cook), or via the recognition of recurring patterns throughout the day (e.g., hospital visit day). Several studies have been conducted to investigate real life and human-centric applications such as elderly care and healthcare. However, there are limitations on the practical use of OPHs. Specifically, they are often too intrusive to the user or the layout of their homes. They are also expensive to develop and maintain. Furthermore, they lack practical approaches for the evaluation of the quality of life rhythm.

Some approaches (e.g., [5] [6]) attempt to directly capture daily living data using cameras or microphone. However, such systems are often considered too intrusive to the user in the sense that all aspects of their daily living are exposed. Several studies have exploited state-change sensors or *indoor positioning systems* (IPS) to monitor daily activities. However, these systems (e.g., [7] [8]) are also viewed as being too intrusive to home objects due to it is essential to embed sensors in objects, such as doors, windows, a refrigerator, keys, and medicine containers. There have also been many studies based on wearable sensors to monitor the user's health status and motion (e.g., [9]). However, the wearable sensor is intrusive to the human body because it must always to be worn at home. In addition, the deployment and maintenance of those systems are usually expensive. Furthermore, most of these systems simply provide features that facilitate the recording and visualization of the activity logs, whereas the interpretation and assessment of specific achievements are left to the user. As a result, it is difficult for individuals to determine what their healthy life rhythm should be and how their current situations can be improved.

1.3 Goal of research and technology challenges

To address the limitations mentioned in Section 1.2, my research goal is to develop a *new smart system for OPHs* that can minimize intrusiveness and cost, and provide an assessment of life rhythms for individual users. The new system should collect the user's indoor position and environmental data within the home in non-intrusively, using affordable IoT devices. From the data, the system is able to recognizes the daily activities of the user. Based on the recognized activities, the system finally quantitatively evaluates the life rhythms and disseminates practical advice to maintain a healthy life. Using the proposed system, it is expected that people in OPHs would be able to achieve improved health and quality life on their own.

To develop such a system, there are three main technological challenges that must be addressed:

1. Collection of indoor position data and environmental data in OPHs minimizing intrusiveness and the cost for the user.
2. Accurate recognition of daily activities of the user.
3. Quantitatively assessment of the life rhythm of the user.

1.4 Approaches

In order to address the three technology challenges, this investigation is divided into three main projects, (1) Collection of living data of individuals in OPHs, (2) Recognition of daily activity and (3) Derivation of personalized assessment model for life rhythm. The first project is to address the first challenge, A *framework for collecting the user's living data* in OPHs is provided that also with minimizes intrusiveness and cost. The second project involves addressing the second challenge. In this case, a daily activity recognition system based on supervised machine learning is proposed. The final project involves addressing the third challenge. This entails a proposed method that derives a *personalized assessment* *model* based on the recognized daily activities and the user's self-assessment of quality of life (QoL).

- **Collection of living data of individuals in OPHs**

  The key idea of the first project is collecting the user's indoor position and environmental data in the OPH.

  For the indoor position data, in order to minimize the development cost and effort, a common data model for indoor location called *DM4InL* is proposed in addition to a Web-based integration framework called *WIF4InL*, which can reuse integrated heterogeneous indoor positioning systems (IPS). In this investigation, *BluePIN*, a zone-based positioning system with BLE beacons have been developed for affordable IPS implementation, which is emploied in the case where there is no existing IPS could be reused in OPHs.

  With respect to indoor environmental data in the OPHs, a self-managed IoT device called *Autonomous SensorBox* was exploited, which was developed in our laboratory and was designed to minimize the effort associated with deployment and operation. Once a power cable is connected, the SensorBox autonomously measures seven types of environmental attributes (temperature, humidity, light, sound, vibration, gas pressure, and motion) in the vicinity of the box and then periodically uploads the data to a cloud server.

- **Recognition of daily activity**

  For the second challenge, a daily activity recognition method based on *supervised machine learning* is proposed. In the training phase, the proposed method requests that the user should manually record their activities, using a lifelog tool. The activities of interests include sleeping, eating, bathing, cooking, PC working, cleaning, and going out. As the system accumulates environmental and indoor position data, the proposed method generates *training data* by attaching the recorded activities as *labels* to time-series sensing data. By applying machine learning of multi-class classification to the training data, the proposed approach establishes an activity recognition model. Once the model is obtained, the system moves to the operation phase, where the seven kinds of activities are automatically recognized.

Based on the proposed method, two kinds of daily activity recognition model were implemented.

The first recognition model is only based on non-intrusive environmental sensing. In the experiment, careful feature engineering was performed to determine essential environmental attributes that best explain activities in OPHs. Furthermore, three different classification algorithms were applied to the model to compare performance.

The second version activity recognition model was based on non-intrusive environmental sensing and BLE beacon technology. In the experiment, three different algorithms and feature engineering were investigated. In addition, 21 models with different lengths of training period were investigated to determine the appropriate length of the training phase for the utilization of the system.

- **Derivation of personalized assessment model for life rhythm**

  In the third challenge, a method that derives a *personalized assessment model* based on the recognized daily activities and the user's self-assessment of quality of life (QoL). The proposed method characterizes the user's life rhythms based on statistical features of daily activities, especially *eating* and *sleeping*. In addition, the method periodically requests that the user should evaluate their degree of QoL, using a designated questionnaire survey. The method then establishes a *regression model* that explains the QoL based on the statistics associated with daily activities. Using the model, the user can better understand the details of their current life rhythm. In addition, the model can be used to produce personal advice on daily habits to maintain or improve the user's healthy life rhythm.

  An experiment was conducted in an actual apartment where activity logs for 224 days and self-assessment QoL logs for 32 weeks were obtained. Based on the experimental results, of the assessment model personalized to the resident was interpreted and appropriate habits for maintaining high QoL were identified.

Eventually, the integration of the proposed system with a life monitoring system was considered, to allow the system to automatically intervene in OPH to encourage the maintenance of healthy life rhythms.

1.5 Scope of The Dissertation

The broader aim of this dissertation is to develop a new smart system for individuals in OPHs that can minimize intrusiveness and cost, and provide an assessment of life rhythms for individual users. Towards this aim, the long-term research is divided into three main projects, as indicated in 1.4. (1) Collection of living data in OPHs, (2) Recognition of daily activity and (3) Deriving personalized assessment model of life rhythm. In this dissertation, a separate chapter is dedicated to each of the three projects. Thus, this thesis is divided as follows.

In Chapter 2, a detailed description of the first project is provided. The *Data Model for Indoor Location (DM4InL)* is first proposed, which specifies a common data schema for representing indoor location information in an application-neutral way, and does not depend on any specific IPS or applications of IPS. Then, the *Web-based integration framework (WIF4InL)* is proposed, which can reuse integrated heterogeneous IPS and introduce *BluePIN*, which is a zone-based positioning system with BLE beacons. Finally, the *Autonomous SensorBox* is briefly introduced, which is an IoT device with seven kinds of environmental sensors, which was developed by our laboratory.

In Chapter 3, the second project is described in detail. Two kinds of daily activity recognition models are proposed for OPHs. A recognize model only using *Autonomously SensorBox* with supervised machine learning is first proposed. Then, a new daily activity recognition model using *Autonomous SensorBox* and *indoor position* is proposed. Various configurations of learning algorithms and features engineering are evaluated for both of the models.

In Chapter 4, the *personalized assessment model* is proposed. This model can quantitatively assess the user's life rhythm by analyzing the daily activity log and self-assessment of QoL logs. Based on experiments involving an actual resident, the method of evaluating personal life rhythms and the dissemination of practical advice to maintain and improve healthy living is introduced.

Finally, Chapter 5 concludes this dissertation and presents proposed future research.

Chapter 2

Collection of Living Data of Individuals in OPHs

2.1 DM4InL: Common Data Model for Indoor Location

2.1.1 Introduction

With the rapid development of wireless and sensor technologies, research, and development of an *Indoor Positioning System* (*IPS*, for short) are actively being conducted. An IPS identifies precise positions of individuals and objects in *indoor space* where a Global Positioning System (GPS) is unable to function. Various enabling technologies for IPS have been developed so far, including those based on Wi-Fi [10], infrared [11], ultrasound [12], IMES [13], pedestrian dead reckoning (PDR) [14], etc. Several commercial IPSs have already been introduced come into the market (e.g. PlaceEngine[15], Guardly[16]). These enabling technologies have different characteristics in terms of accuracy, resolution, and cost of infrastructure deployment.

By using IPS, various *Indoor Location Applications* (called *InL-App*, for short) can be implemented. An InL-App performs appropriate actions and behaviors autonomously, according to the indoor position of users or dynamic/static objects. Typical InL-App includes the navigation service of a shopping mall, an exhibition guidance service for a museum, a location-aware appliance control in a smart home, location-aware targeted advertising, lifelog, and so on.

When an InL-App is implemented with an IPS, it is necessary to determine an approach to represent and manage the indoor location information obtained by the IPS within the InL-App. Many existing systems individually represent and manage the indoor position information, considering the purpose of the InL-App and the characteristics of the IPS used. Such a proprietary representation and management method has the advantage of optimal performance. However, it causes *tight coupling* of the InL-App and the IPS, whereby the indoor-location data and processes cannot be reused among different services. Thus, the proprietary method increase the complexity of implementation of InL-App and in addition to the development cost and overall effort.

To improve efficiency and reusability in the development of InL-App, we are investigating the application of a cloud service, called *Indoor Location Query Service* (*InL-Query*, for short). InL-Query gathers indoor location information of various objects (room, equipment, appliance, people, etc.) from arbitrary IPSs. It then generates *application-neutral* API, whereby external systems can query the indoor location of a specified object or building. Thus, InL-Query achieves loose-coupling of the IPS and the InL-App, which facilitates sharing and reuse of indoor information and common procedures.

As the first step towards implementation of InL-Query, a *Data Model for Indoor Location* (*DM4InL, for short*) is proposed as part of this study. DM4InL specifies a common data schema for representing indoor location information in an application-neutral way, which does not depend on any specific IPS or InL-App.

The proposed DM4InL consists of three models: *location model*, *building model*, and *object model*. The location model represents any location in a building according to the relative position (3-dimensional offset) from the base coordinates of the building. The building model defines every building with attributes and global position. It also defines geographic elements in each building such as partitions, routes, and spots. The object model defines various objects in a building, such as people, appliance, furniture, etc. The current position of each object is represented by a point defined in the location model. By using DM4InL, developers of InL-App are able to manage indoor location information independent of specific IPS, which facilitates the sharing of data among different InL-App.

Several APIs were also investigated for basic queries to the InL-Query. For example, the API getObjectLocation() returns the indoor position of a given object. Another API getObjectsInPartition() returns all object that exist in a certain partition. Using the API, it is possible for the developers to obtain the indoor position of an object without knowing the implementation details of the IPS. Thus, the API significantly improves the development efficiency of InL-App.

2.1.2 Preliminary

Overview of Indoor Positioning Systems (IPS)

IPS is a generic name for systems that estimate the position of a subject or object inside a building. An IPS is a solution based on magnetic, other sensor data or a network of devices used to wirelessly locate objects or people inside a building [17]. Enabling technologies for IPS include the followings:

- Wi-Fi [15]
- 2D-Code [18]
- Visible light communication [19]
- IMES (Indoor Messaging System) [13]
- RFID tag [20]
- Hybrid methods of position recognition [21]
- PDR [14]

The aforementioned technologies have different characteristics in terms of accuracy, resolution and cost of infrastructure and deployment, which are generally chosen based on requirements and cost for the target solution. An increasing number of different technologies for IPS are being developed to complement the existing GPS, which cannot provide coverage inside buildings. However, unlike the GPS, there is presently no de-facto standard for the IPS.

Application of Indoor Position

In this dissertation, we refer to a service that performs appropriate actions or behaviors according to the indoor location information as *Indoor Location Application (InL-App)*. Practical InL-Apps have come to market, including:

- **Shoplat [22]:** Using an ultrasonic positioning technology, this service recommends coupons or the loyalty program of a business established when it is approached by a user. The user's location is estimated by broadcasting ultrasound that can be detected by the microphone of the user's smartphone.
- **PlaceEngine [15]:** This service can be used to monitor the motion and location of staff working in a hospital. The IPS is implemented within the beacon frame of a wireless LAN.

Generally, the conventional systems of InL-App have been implemented individually, considering the purpose of service and the IPS used. There is no standard for the representation and management of indoor location information. Thus, individual systems implement their own proprietary approaches.

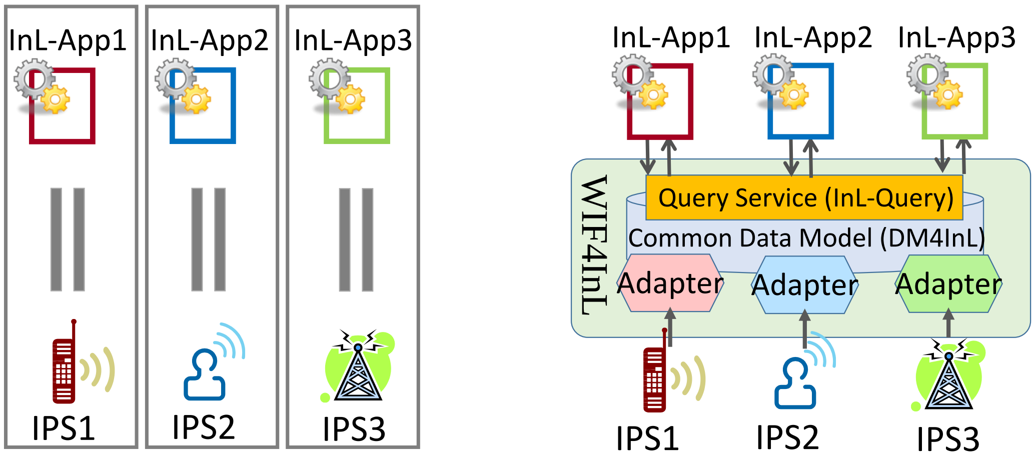
Geographic Information System (GIS)

GIS is a computer system designed to capture, store, manipulate, analyze, manage, and present all types of geographical data [23]. A wide range of data is used in a GIS. Meta-information is roughly divided into graphical information (maps, aerial photographs or satellite image, etc.), attribute information associated with the feature, geodetic system, projection method, reduced scale, accuracy, etc.

The GIS represents spatial data in two different formats: *vector* and *raster*. In this research, we focus on the vector format. The vector format data consists of individual *points*, which are represented coordinates. Multiple points may be joined in a particular order to create a *line* or joined into a closed ring to create a *polygon*. All the vector data fundamentally consist of a list of coordinates that define vertices, together with rules to determine whether and how those vertices are joined. Shapefile, a GIS file format, which uses vector data and a variety of attribute data (such as property, feature and numeric, etc.) have become a standard format in the GIS industry.

Approach

Figure2.1(a) shows the implementation architecture of a conventional InL-App. As mentioned in Sections 2.1.2 and 2.1.2, each InL-App is tightly coupled with an



(a) Conventional InL-App (b) Proposed InL-App

Fig. 2.1. Two Different Architectures of InL-App

IPS. Moreover, the indoor location data and the necessary program are managed independently within each system. Therefore, one system cannot share or reuse the data and program of another system. As a result, each system tends to be complex and difficult to manage.

To address the problem, our long-term goal is to establish an innovative architecture as shown in Figure 2.1(b). In the proposed architecture, the *Indoor Location Query Service (InL-Query)* accumulates spatial information from various IPS in a standardized way, and provides the information for various InL-App via an application-neutral query API. Using the InL-Query, we can achieve loose coupling between the InL-App and the IPS. This significantly improves the efficiency and reusability of InL-App development. Moreover, by deploying the InL-Query in the cloud, it is possible to share and use the indoor information globally across multiple buildings.

As the first step towards the long-term goal, a common data model called *Data Model for Indoor Location* (*DM4InL*) is proposed, in this section. As shown in the middle of Figure 2.1(b), DM4InL aims to prescribe a common data schema which does not depend on any specific IPS or InL-App. Given that there is no de-facto standard format, DM4InL is developed by referencing the GIS introduced in section 2.1.2.

According to Yuan et al. [24], every spatial object must have theme, space and time attributes, to represent what, where and when, respectively. Hence, we divide theme and space attributes into two models, location model and object model. Moreover, the space attribute can be represented by the following forms:

- **Geographic coordination:** This form indicates the position information by latitude-longitude and altitude, which is generally used in GPS research field.
- **Relative coordination:** This form refers to two components. One is movable object's position reference a building. Another one is the global position of the building.

To adopt the proposed model to InL-App, the second form is more suitable. The second form represents two spatial objects, movable object (such as a person, vehicle and so on) and building (such as a house, parking allocation and so on). Therefore, we divide the spatial object into building model and object model. Every object in the object model has an indoor-location point (in the location model), each of which is explained by spatial elements of a building (in the building model). For time attributes, we defined a time-series indoor location log for every object in the object model.

2.1.3 Requirements of DM4InL

The primary requirement of DM4InL is to associate any object inside a building with location information, without depending on any purpose or application. More specifically, the objective to satisfy the following requirements R1-R4:

- **Requirement R1:** The data model should be able to represent any position inside a building as location information with spatial and subjective attributes.
- **Requirement R2:** The data model should be able to associate any object inside a building (such as a person, device, or furniture) with the location

information.

- **Requirement R3:** The data model should be able to represent spots, routes, and partitions as geographic elements within a building.
- **Requirement R4:** The data model should be able to provide seamless positioning, integrating indoor and outdoor location information.

2.1.4 Overview of DM4InL

In order to satisfy these requirements, we construct DM4InL as a composition of the following three data models:

- **Location Model:** It defines indoor location information. Every position in a building is represented as a *relative coordinate* (3-dimensional offset) from a *reference point* of the building. Using the coordinate, four geometric primitives are constructed: *local point*, *local line*, *local polygon*, and *local space*. It also defines *global positions* using the triplets of [longitude, latitude, altitude].
- **Building Model:** It represents spots, routes, and partitions as geographic elements within a building. Each spot (route or partition) is identified by a local point (a local line or a local space, respectively) in the location model. It also identifies every building with a reference point represented by a global position.
- **Object Model:** It represents various objects (such as people, furniture, and appliances) within a building. Each object refers to a local point in the location model to represent its current position.

We will explain the details of each model in the following subsections.

2.1.5 Location Model

The location model defines *indoor positions* and *geometric primitives*. Initially, the indoor position must be defined based on the foundation that every position is associated with a single building. In addition, as discussed in [25], the used of 3-dimensional Cartesian coordinates is convenient for the use and representation

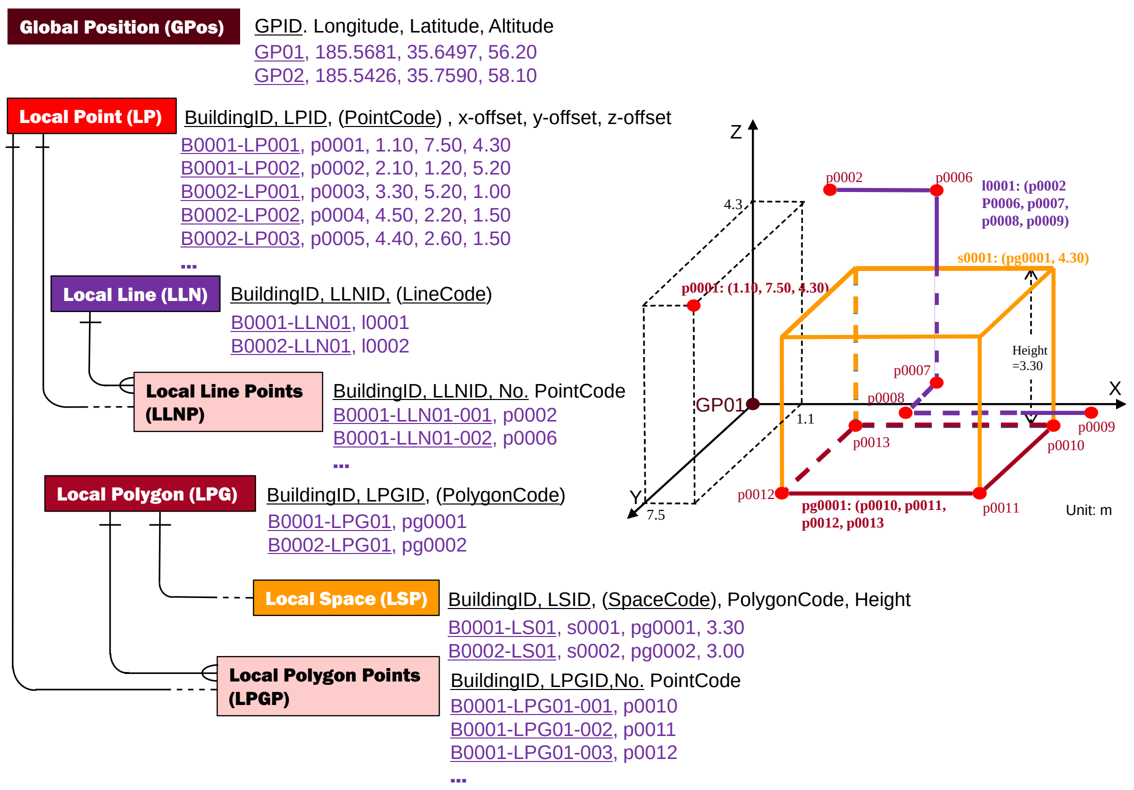


Fig. 2.2. Entity Relationship (ER) Diagram of Location Model and Schematic Representations of Instances

of indoor locations by applications. Based on these two reasons, we represent every indoor position using a 3-dimensional Cartesian coordinate system. Each position is defined as a *relative coordinate* (3-dimensional offset) from a *reference point* of the building to which the position belongs. Furthermore, utilizing the fundamental concept of GIS, we define *local point*, *local line*, *local polygon* and *local space* in the location model.

Figure 2.2 shows the ER diagram of the proposed location model. The diagram follows the notation defined in [26]. A square represents an entity whose schema is defined by multiple data items arranged in the right side. An underlined item represents a primary key, and an underlined item with brackets defines a secondary key. Other items represent attributes. Data instances are listed under each entity. A relationship may be defined between a pair of entities, where

- (+——∈) represents a parent-child relationship,
- (+——· · ·) represents a reference relationship,

- (+——○+) represents a sub-type relationship

The details of each entity are explained as follows.

(1) Local Point (LP)

LP defines every indoor position of a building to which the position belongs and a 3-dimensional offset (x, y, z) from the reference point of the building. The primary key is a composite key consisting of a building ID (defined later) and a point ID. It also has a sequential code as a secondary key, so that external entities can easily refer to the point. The attributes are the coordinate values of LP in the 3D Cartesian system that allows external applications to locate the point in a building.

Figure2.2 contains five instances of LP, where two points belong to building B0001 and three points belong to building B0002. Points p0001 and p0002 are depicted on the right side of the ER diagram. Point p0001 represents a position, whose coordinate is 1.10m in the x-direction, 7.50m in the y-direction and 4.30m in the z-direction from a reference point (GP01) of the building B0001.

(2) Local Line(LLN)

LLN represents a line constructed using two or more local points. Similar to LP, LLN has a composite key consisting of a building ID and a line ID. It also has a sequential code as a secondary key. Each line is defined by multiple points called local line points (LLNP). Each LLNP has a composite key consisting of an ID and a sequence number of the line, and a point code referring to a local point to locate the LLNP. Thus, a parent-child relationship exists from LLN to LLNP and a reference relationship exists from LLNP to LP.

Figure 2.2 contains an instance of LLN (l0001) that belongs to building B0001, constructed by p0002, p0006, p0007, p0008, and p0009. The line is depicted in the right figure.

(3) Local Polygon (LPG)

LPG represents a polygon constructed using three or more local points. Similar to LLN, each polygon is defined by multiple points called local polygon points

(LPGP). Each LPGP has a composite key consisting of an ID and a sequence number within the polygon, and a point code referring to a local point to locate the LPGP. Thus, a parent-child relationship exists from LPG to LPGP, and a reference relationship exists from LPGP to LP.

Figure 2.2 contains an instance of LPG (pg0001) that belongs to building B0001, constructed using p0010, p0011, p0012, and p0013. The polygon is drawn as a square in the right figure.

(4) Local Space (LSP)

LSP represents a 3D space in a building. A 3D space is generally constructed using several polygons. However, taking the convenience and characteristics of an ordinary indoor space into account, local space is defined as a pillar-shaped space by extending a local polygon to the z-axis direction. LSP has a composite key consisting of a building ID and a space ID. It also has a sequential code as a secondary key. Attributes are a reference to a local polygon as the bottom of the space, and the height of the pillar from the bottom.

Figure 2.2 contains an instance of LSP (s0001) in building B0001, which is defined as a pillar of height 3.30m made from a local polygon pg0001. The space is represented as a cube in the right figure.

(5) Global Position (GPos)

GPos represents a global position that is used for the reference point of a building. Since Requirement R4 suggests seamless positioning among indoor and outdoor locations, the reference point of every building is located at a global position. Each global position is represented by the triplet of longitude, latitude, and altitude. It is referenced from a building in the building model, as is explained later.

2.1.6 Building model

The building model defines spots, routes, and partitions within a building. Such a structure can be regarded as a container that can contain various objects. From this viewpoint, the building model should represent the container itself and geographic elements inside the container. The location of each geographic element

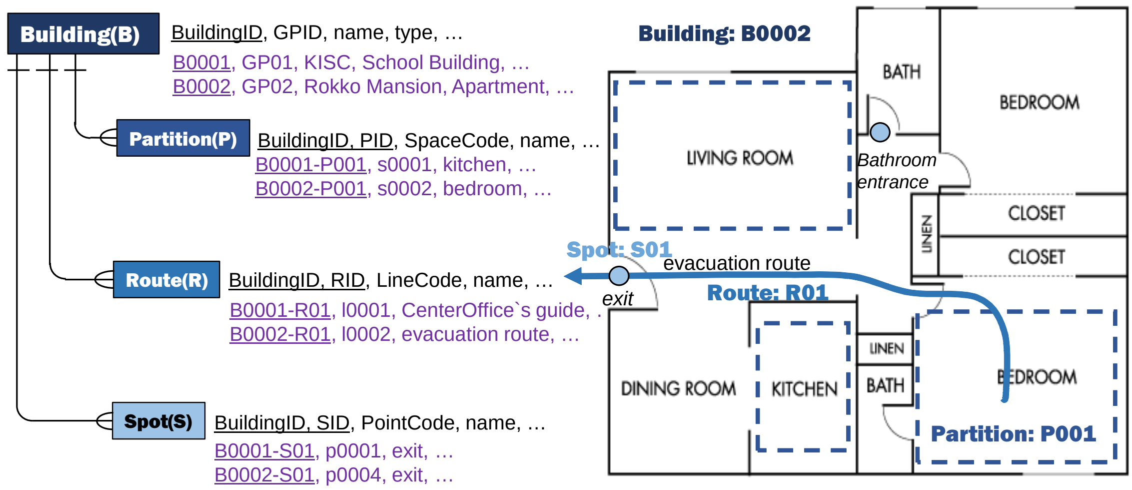


Fig. 2.3. ER Diagram of Building Model and Schematic Representation of Instances

is associated with a corresponding entity in the location model. Figure 2.3 shows the ER diagram of the building model. Each entity is explained below.

(1) Building (B)

The building entity represents the existence of a building. A building is identified by a building ID, which is a primary key. Attributes include a reference to a global position where the building exist (see (5) of Section 2.1.5), building name, type, etc. As mentioned in Section 2.1.5, every entity in the location model is associated with a building ID, which constitutes a parent-child relationship. As such, when a building is eliminated, all the location information in the building is also removed. The global position of a building serves as a reference point of the indoor position. Because every local position is represented by a relative coordinate from the reference point, the local position can be converted into a global position. This satisfies Requirement R4.

In the real world, a building can assume various granularity. For example, when we consider the collective housing of apartments, there are choices: a building refers to the entire building, or a building can refer to an apartment within the building. As a result, we consider it to be reasonable to define a building for every living unit a single IPS can provide coverage.

(2) Spot (S)

The spot entity represents a geographic element treated as a point, including a bathroom entrance, emergency exit, and so on. A spot has a composite key consisting of a building ID and a spot ID. Attributes include a reference to a local point that locates the spot, the name of the spot, etc.

(3) Route (R)

The route entity represents a geographic element treated as a line, such as an evacuation route, a traverse of a residence, etc. A route has a composite key consisting of a building ID and a route ID. Attributes include a reference to a local line that draws the route, the name of the route, etc.

(4) Partition (P)

The partition entity represents a geographic element treated as a space, such as a living room, a bedroom, a kitchen, etc. A partition has a composite key consisting of a building ID and a partition ID. Attributes include a reference to a local space that surrounds the partition, the name of the partition, etc.

The right side of Figure 2.3 shows schematic representations of instances over a floor plan of a house. A circle represents a spot, an arrow represents a route, and a dotted rectangle represents a partition.

2.1.7 Object Model

The object model represents various objects in a building. It is assumed that every object is movable and is represented by an indoor position when the object is in a building. In the real world, there are various kinds of objects (e.g., people, appliance, furniture) with attributes that may vary. Therefore, we first define an abstract entity that associates any object with location information. We then define each concrete entity as a sub-type of the abstract entity. Further sub-types can then be added as needed. Figure 2.4 shows the ER diagram of the object model. Each entity is explained in the following sections.

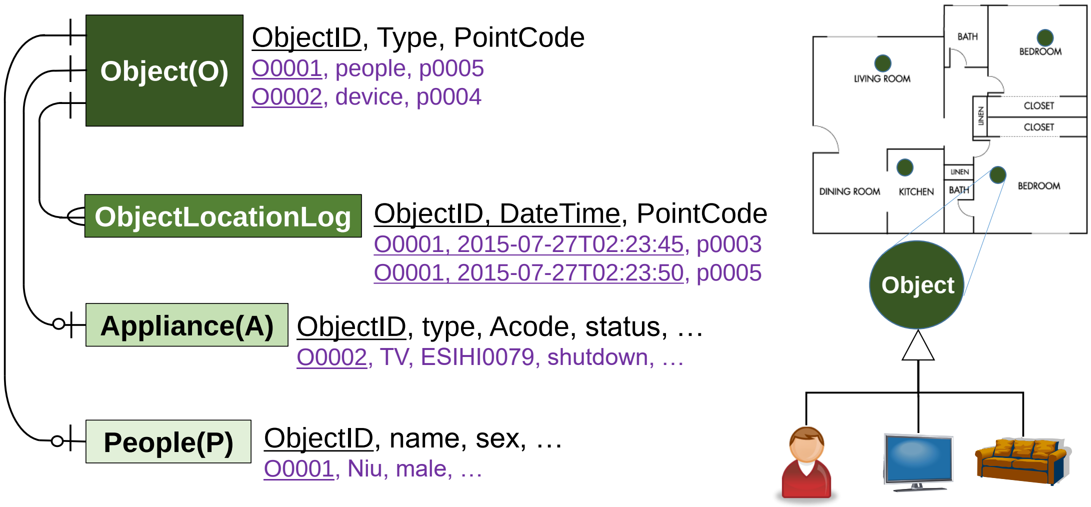


Fig. 2.4. ER Diagram of Object Model and Schematic Representations of Instances

(1) Object (O)

The object entity abstracts an arbitrary object in a building. An object is identified by an object ID, which is a primary key. Attributes include the type of object and the reference to a local point where the object currently exists. The type is a reference to a concrete object entity. The coordinate of the local point establishes a positional relationship between the point and a space (or a line). Hence, we can deduce a geographic element of the building (i.e., a spot, a route, or a partition), where the object currently exists.

The right side of Figure 2.4 illustrates four instances of the object entity. We can identify the current position of each object from its local point. We can also deduce a space from the position. For example, it is evident that an object exists in the bedroom. If the type of object is ’people’, the object is explained in details by a ’people’ sub-type entity.

(2) Sub-Types of Object

A concrete object (e.g., people, appliance, furniture, etc.) is defined by a sub-type entity of the abstract object. A sub-type object defines the attributes that are necessary for the type of object. The sub-type and the abstract have the same

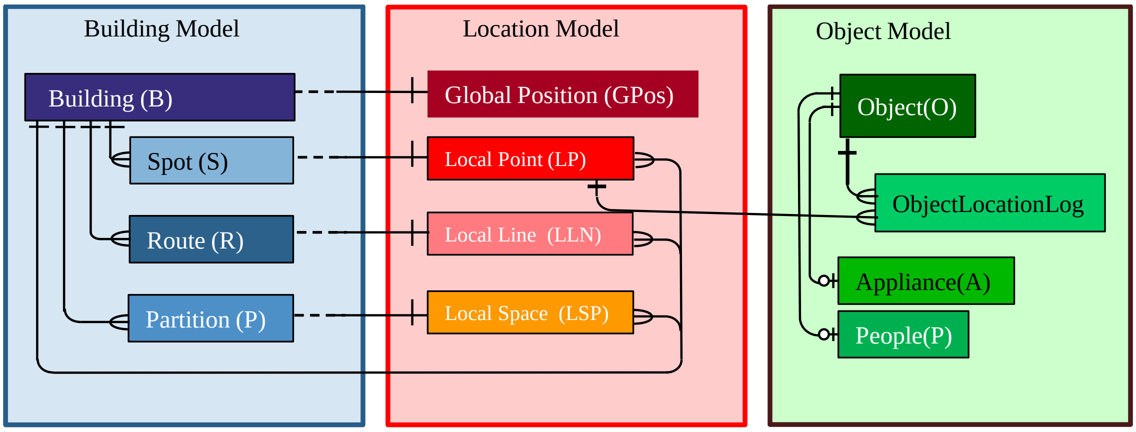


Fig. 2.5. DM4InL as A Composition of Three Models

object ID as a primary key. Therefore, we consider that a sub-type extends an object based on the type attribute.

Figure 2.4 illustrates two sub-types: people and appliance. For instance, the 'people' entity defines the attributes of name and gender. The 'appliance' includes attributes of the appliance type, model number, and state, etc.

Object Location Log

In addition to record the current position of each object in Object(O), we also defined Object Location Log that stores the history of time-series indoor locations for every object. As mentioned in Section 2.1.2, according to Yuan et al. [24], every spatial object must have theme, space and time attributes, to represent what, where and when, respectively. Thus, we defined object location log records three aspects of information: what, when and where. For the theme attribute, each log refers to an object ID to represent its theme. As for time attribute, we defined UTC (Coordinated Universal Time). And each log refers to a local point in the location model to represent its position. Figure 2.4 illustrates two logs for one person. The data representing that person `Niu` moved from `p0003` to `p0005` for 5 seconds.

2.1.8 Data Schema of DM4InL

DM4InL is composed of the aforementioned three data models (i.e., location model, building model, and object model). Figure 2.5 shows an ER diagram of

DM4InL, representing the relationships between the three models. Every indoor location (i.e., LP, LLN or LSP) is associated with a single building, whereas every building involves more than one indoor location. A building (B) is identified by a global position (GPos). A geographic element in a building (i.e., spot, route or partition) refers to a location entity (LP, LLN or LSP, respectively). An object location log is referring to a local point and an object entity. These observations are well-suited to general intuition regarding the indoor locations, objects, and buildings in the real world.

2.1.9 Discussion

Sufficiency of Requirements

This section discusses the extent to which the proposed DM4InL satisfies requirements R1-R4 in Section 2.1.3. Based on the definition of the location model (using the relative coordinate bound with a building) any position inside any building can be represented. In addition, an element of the location model can be an attribute of a geographic element in a building. Hence, requirement R1 is satisfied. Based on the definition of the object model, it is possible to associate every object with a local point. Thus, requirement R2 is satisfied. Given that the building model represents spots, routes, and partitions within a building, Requirement R3 is satisfied. Every indoor location is associated with a single building and every building has a reference point identified by a global position. Combining the global positions and the relative coordinate of an indoor location, it is possible to convert indoor location into a global position. Therefore, requirement R4 can be satisfied.

It should be noted that requirements R1-R4 are derived by purely considering the properties of the common data model for the indoor location; they do not depend on any specific application or purpose. Thus, I believe that the constructed DM4InL is a *neutral* data model, which can be shared and reused by various IPS and InL-App.

Towards Query API for DM4InL

Using DM4InL, we can represent indoor location information of objects and geographic elements in an application-neutral form. To allow a variety of services and applications to easily and efficiently use the data, it is essential to implement APIs that implements typical queries for the DM4InL. In this regard, the following APIs are currently being developed:

(1) Position API: It queries location information of a given spatial element or an object. For instance, getObjectPosition(objectID) returns the current position of a given object, and getBuildingPosition(buildingID) returns a global position of a given building.

(2) Attribute API: It queries attributes with respect to an object or a spatial element. For instance, getObjectsInPartition (partitionName) returns all objects within a given partition.

The design and implementation of the API will be left for the next step of the research project.

Limitations

Indeed, there are a few limitations in the current version of DM4InL. An example of such a limitation is that *uncertainty* [27] of the location data, or to estimate the future location of an object. Although the uncertainty concept enhances the expressivity of the model, it also increases the complexity. The determination of whether time should be included in DM4InL is currently being investigated.

Another limitation is in the representation of local space. As seen in Section 2.1.5, each space is defined as a pillar-shaded space for convenience. It is necessary to evaluate this definition to ensure that it can adequately address practical InL-App.

Finally, the object model may be refined further to represent the dynamic *context* of every object. Typical contexts include the current activity of a person, the status of a device, and the direction and placement of furniture.

Related Work

Leonhardt et al. [27] proposed an approach that constructs a *generic query layer* between the location sensors and the positioning systems. Although their approach is similar to and achieves some of our long-term goals (see Figure 2.1(b)), it is meant to be applied to a lower layer between the IPS and physical sensor devices. Their hierarchical model also addresses *uncertainty* and *data conflicts* in the case of an unreliable indoor positioning environment. These issues are not yet considered in DM4InL. Together with the time concept discussed in Section 2.1.9, these aspects will be investigated in our future work.

Kim et al. [28] proposed and *Indoor Spatial Data Model* (*ISDM*), which uses CityCML to define location data for 3D indoor location-based services. ISDM can define *topologies* among spatial elements using the 3D object topology model, which is more expressive than the proposed building model. However, it lacks a way to explicitly associate indoor objects with locations, as defined in the presented object model.

2.1.10 Summary

In this section, a *Data Model for Indoor Location (DM4InL)* is proposed, which prescribes a common data schema to represent indoor location information. By combining three data models (i.e., the location model, the building model, and the object model), DM4InL represents location information of various objects inside a building in a standard format. The proposed method contributes to loose coupling of InL-App and IPS, which will significantly improve the efficiency and reusability in the development of InL-App. My future work includes the evaluation of the data model based on practical use cases of InL-App, as well as the design and implementation of the query API for DM4InL. Using the DM4InL and the query API, we will also develop the Indoor Location Query Service (InL-Query), which is a long-term goal. Finally, the extension of DM4InL (w.r.t. the time concept, object contexts, local space, etc.) should be considered carefully in order to address pragmatic issues (such as uncertainty and data conflicts).

2.2 WIF4InL: Web-based Integration Framework for Indoor Location

2.2.1 Introduction

Following the achievement of DM4InL (see Section 2.1), the main objective of this work is the construction of a framework in which various InL-Apps can easily share and utilize indoor location information gathered using different IPS. In this regard, a Web-based integration framework is presented for indoor location called (WIF4InL) in this work. Using DM4InL, the proposed WIF4InL integrates indoor location data obtained from existing heterogeneous IPS, and provides common operation as a service for various InL-Apps. In this respect, the two main challenges must be addressed. The first challenge is data integration, i.e., conversion of the indoor location data produced by heterogeneous IPS into DM4InL. The next challenge is operation integration, i.e., the task of implementing comprehensive location-based queries to retrieve data from DM4InL, to be shared by various InL-Apps.

To address these challenges, WIF4InL is designed based on the following three components: (1) InL-Adapter, (2)InL-Database and (3)InL-Query. InL-Adapter adapts the proprietary indoor location data to common data model DM4InL. The InL-Adapter converts the uploaded location data into DM4InL. InL-Database is a large-scale shared database that manages the translated data. InL-Query provides application-neutral API for various InL-Apps to query the indoor location information. The operations for these components are published as cloud services, and thus they are loosely coupled to service-orient architecture.

To evaluate practical feasibility, the proposed framework is applied to integrate two different IPSs. The first IPS is RedPin [29], which uses Wi-Fi fingerprints to locate mobile devices. The second IPS is BluePin, which uses Bluetooth beacons to detect the proximity of devices. The proposed WIF4InL integrates the two different IPSs so that applications can transparently use indoor location information gathered by both systems. It is unnecessary for the applications to manage the differences between RedPin and BluePin. Given that WIF4InL facilitates the loose coupling between IPS and InL-Apps, it improves the reusability and interoperability of indoor location information and operation. Thus, it is promising for the reduction of development cost and effort of InL-App.

2.2.2 Preliminary

Classification of IPS

In general, IPS can be divided into several categories:

- **Vision Based Indoor Localization:** Visual information can be collected and used for indoor navigation as outlined in numerous examples in the literature (e.g., [30][31]). However, image-based localization consumes more computing resource (image analysis) and power.
- **Wireless Based Indoor Localization:** Unlike light waves, significantly longer waves in the electromagnetic spectrum such as radio waves and microwaves can penetrate doors and walls, and provide ubiquitous coverage of a building. Development based on existing wireless technologies (e.g., Wi-Fi, Bluetooth) is relatively easy and microwaves do not restrict human activity in buildings. Moreover, the power and computing resources required is also significantly less compared to vision-based indoor localization. Most current work in indoor localization use this approach.
- **Other Methods:** There are many other ways to achieve indoor localization. They include the exploitation of ultrasound [32], acoustic background fingerprints [33], accelerometers [34], and campus by adopting a dead-reckon method [35].

Among the recent literature, wireless-based indoor localization methods are prominently highlighted. According to the mathematical techniques used, they can be categorized into the following three groups:

- **Proximity:** This method assumes that if a user enters within the range of a known station, then the location of the user is approximated to the station. An IPS is been developed using Bluetooth Beacon technology which is called *BluePin*. On detecting the proximity of a user, BluePin produces symbolic location data. The following data "L1" indicates that user `niu` is close to an entrance of room S101.

```
L1:{personId:niu, locationId:22, locationName:
S101 Entrance, LastUpdate: 2015/07/27 11:23:45
JST}
```

- **Triangulation:** This method uses geometric knowledge to obtain the user's location. The location is determined by either the distance to the fixed known measurement points, or the received signal angles.
- **Fingerprint:** Fingerprint refers to the characteristic features of a signal. The method assumes that each position in the area has a unique fingerprint. Relying on prior knowledge associate a fingerprint with a position, the current location of a user is obtained. For example, Redpin [29] is an open-source IPS which uses Wi-Fi fingerprint for zone-based positioning. The following data "L2" is produced by RedPin, which represents a user's location by a point (345, 567) on a map KU-System-1F:

```
L2:{locationId:45, mapName:KU-System-1F, map
Xcord:346, mapYcord:567, symbolicId:S103,
macAddress:'08:60:6e:32:b6:0b'}
```

Cases of InL-App

In this section, *Indoor Location Application (InL-App)* refer to any location-aware service or application that performs appropriate actions according to indoor location information. To assist in the description of the approach, the following examples are considered:

- **SmartShop:** This service advances coupons or loyalty program associated with a commercial entity such as a shop, to a smartphone, when a user approaches the shop. The user's location is estimated by BluePin, whereby a static beacon station is installed at the entrance of the shop. When user '1' gets close to the station, their smartphone uploads the location data {`personId:1, locationId:7, locationName:shop1, lastUpdate:2015/07/27 11:23:45 JST`} to a server. When SmartShop determines that user '1' is in the shop, it advances shop coupons to that user's smartphone.

- **LocEyes:** This service determines the location of all staff working in an institute. It is assumed that every staff member has a smartphone with RedPin, and that the smartphone uploads the current indoor location every 10 seconds. An instance of the location is {`locationId:45, mapName:KU-System-1F, mapXcord:346, mapYcord:567, symbolicId:S103, macAddress:'08:60:6e:32:b6:0b'`}. Based on the data, the server visualizes the latest location of every staff member on the map KU-System-1F.

It is easily understood that there is no compatibility between SmartShop (with Bluepin) and LocEyes (with Redpin). Indeed, they are individually developed and operated, considering the service objectives and the underlying IPS. Figure2.1(a) shows the implementation architecture of these InL-Apps. It is evident that each In-App is tightly coupled with an IPS, and that indoor location data and program are managed independently within each system. Therefore, one system cannot share or reuse the data and operation of another system. As a result, each InL-App must be developed independently of a preexisting framework, which increases cost and effort.

Research Goal

As mentioned in Section 2.2.2, my research goal is to establish an application platform as shown in Figure 2.1(b), which achieves loose coupling between InL-Apps and the underlying IPS. This dissertation focuses on the implementation of a framework that horizontally integrates the existing IPS and provides common operation as a service shared by various InL-Apps.

To implement such a common framework, it is necessary to address **two main challenges**: *data integration* and *operation integration*. The data integration considers how to convert the indoor location data produced by various IPS into one that conforms to a common data model. We investigate how to convert the proprietary indoor location data into DM4InL. On the other hand, the operation integration deals with the implementation of comprehensive queries to retrieve application-neutral location data from DM4InL.

2.2.3 Overview of WIF4InL

Architecture

To address the challenges mentioned highlighted in Section 2.2.2, WIF4InL *(Web-based Integration Framework for Indoor Location)* is proposed. WIF4InL works as an abstract layer between InL-App and IPS. This layer first integrates indoor location data gathered by heterogeneous IPS and then provides application-neutral APIs for various InL-App. As such, InL-App can transparently access different IPS transparently.

Figure 2.1(b) shows its architecture. The WIF4InL consists of three components: InL-Adapter, InL-Database, InL-Query. The features of each component are described as follows:

- **InL-Adapter (Indoor Location Adapter Service)**
  This is a Web service that adapts the proprietary indoor location data to the common data model DM4InL. When a client uploads proprietary indoor location data via a Web-API, the InL-Adapter converts the data into a format for DM4InL and inserts the converted data in a database (InL-Database, see below). Because different IPSs create location data in a different format, we need to implement a dedicated adapter must be implemented for every IPS.
- **InL-Database (Indoor Location Database)**
  This is a large-scale shared database that manages the indoor location data provided by InL-Adapter. Every record of indoor location data complies

with DM4InL, and is stored with a time-stamp to keep the history.

- **InL-Query (Indoor Location Query Service)**
  This is a Web service for querying indoor location data stored in the InL-Database. It provides application-neutral APIs for various InL-Apps to query the indoor location of any object. InL-Query provides two types of APIs: fundamental API and composite API.

The entire WIF4InL is deployed as a web service, where the aforementioned components are loosely coupled using service-oriented architecture (SOA).

Approach Overview

In order to manage the data integration and the operation integration, WIF4InL is designed specifically as follows:

- **Data Integration:** Heterogeneous indoor location data are managed in a single schema of DM4InL. The conversion from proprietary data format into DM4InL is conducted by individual InL-Adapters. As such, it is unnecessary to modify the existing IPS. A client uploads the location data via the Web-API of a designated InL-Adapter, where all the tasks for the data conversion and storing are delegated to WIF4InL. The details of the InL-Adapter will be described in Section 2.2.4.
- **Operation Integration:** Heterogeneous operations for the existing IPS are consolidated by InL-Query, whereby every InL-App can retrieve indoor location data in DM4InL. Each application does not need to know the technical details of the underlying IPS. As will be shown in Section 2.2.5, InL-Query provides fundamental API and composite API.

2.2.4 InL-Adapter for Data Integration

Overview

To achieve data integration of heterogeneous IPS, the proposed WIF4InL implements InL-Adapter *(Indoor Location Adapter Service)*. InL-Adapter is a Web service that adapts proprietary indoor location data to DM4InL format.

As mentioned in Section 2.2.3, the clients for each type of IPS first uploads the proprietary location data to InL-Adapter, and then the InL-Adapter converts the data into a DM4InL format.

To implement this process, two main issues need to be addressed: *topology adaptation* and *data conversion*. The topology adaptation considers the structure of uploading the measured data to InL-Adapter, which will be described in Section 2.2.4. The data conversion considers how the InL-Adapter converts the uploaded data into DM4InL format, which will be described in Section 2.2.4.

Topology Adaptation

As a first step, it is necessary to modify the existing IPS slightly, so that the measured indoor location data are uploaded to an InL-Adapter. For this modification, it is necessary to consider the *system topology* of the IPS. However, the topology varies from one IPS to another. Therefore, different adaptation patterns are proposed for different topology IPSs.

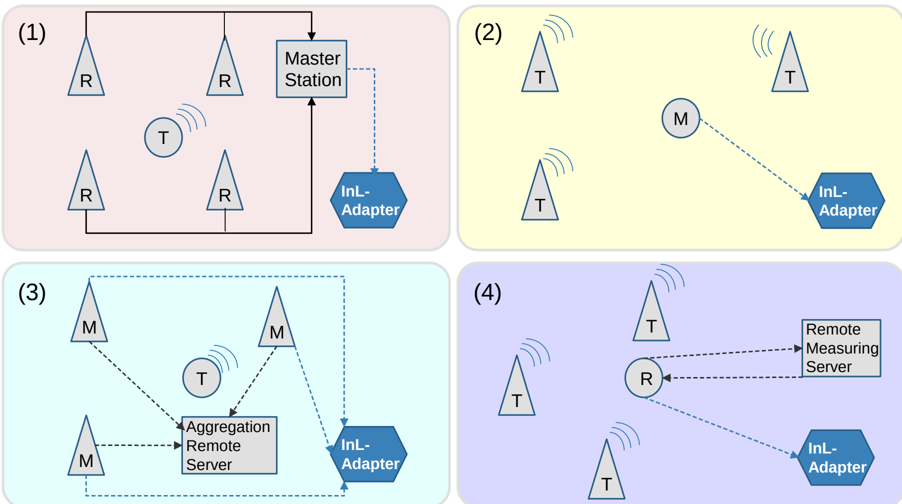


Fig. 2.6. Four Different IPS Topologies and Adaptation Patterns

According to Liu et al. [36], there are four different system topology for IPS: *remote-positioning, self-positioning, indirect remote-positioning*, and *indirect self-positioning*.

Figure 2.6 shows the four topologies. In this figure, a triangle is used to represent a static device or station deployed in the infrastructure, a circle represents the mobile device to be located, a rectangle represents a server and a hexagon represents an InL-Adapter that is newly adapted to the existing IPS. The labels “M”, “R” and “T” identify the roles measuring unit, signal receiver, and signal transmitter, respectively.

Figure 2.6 (1) shows the remote-positioning topology, in which remote server locates mobile device. Static stations receive the signal transmitted from the mobile device and forward the signal to the server. The server then computes the location of the mobile device. An IPS with presence sensors in [37] belongs to this topology. In the remote-positioning topology, all the location data are managed in the server. Therefore, we modify the server to upload the measured data to an InL-Adapter.

Figure 2.6 (2) shows the self-positioning topology, where the mobile device measures the location. This mobile device receives signals from the infrastructure, and computes the current location from the signals. IMES [13] and GPS belong to this topology. In the self-positioning topology, all the location data are managed by the mobile device. Therefore, we modify the mobile device to upload measured location data to an InL-Adapter.

Figure 2.6 (3) shows the indirect remote-positioning topology, where the mobile device is indirectly identified by the remote server. To provide wide area or multiple buildings coverage, several stations with the measuring capability collaborate to transmit location data to the aggregation server. In the indirect remote-positioning topology, the location data are managed by either the measuring stations or the aggregation server. Considering that the aggregation server is often complex and is implemented using un-modifiable patent products, a choice is made to modify the stations to upload the data to an InL-Adapter.

Figure 2.6 (4) shows the indirect self-positioning topology, where the mobile device indirectly obtains its location via the remote server. The mobile device first receives signals from the infrastructure, then forwards the signals to the remote server. The server computes the current location and returns the location data to the mobile device. With the development of IoT and cloud technologies, this type of IPS has gained popularity. RedPin and BluePin introduced in Section 2.2.2

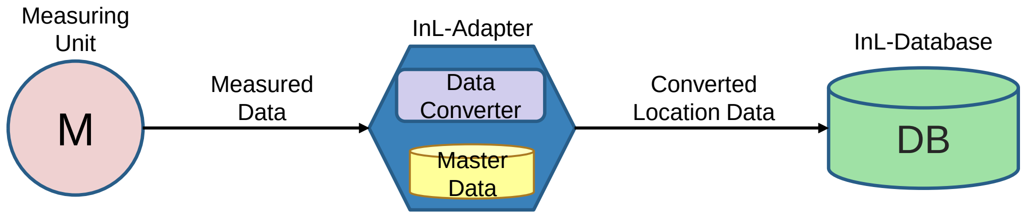


Fig. 2.7. Configuration of InL-Adapter

belong to this topology. In the indirect self-positioning topology, the location data are held by either the mobile device or the measuring server. Considering the complexity and workload of the measuring server, a choice is made to modify the mobile device to upload the returned location data to an InL-Adapter.

Data Conversion

The second step takes the conversion of the location data that is uploaded by the InL-Adapter to DM4InL into consideration. In general, every IPS defines its own location data format by relying on specific infrastructure and/or positioning algorithm. It is, therefore, impossible to enumerate data converters for all possible IPS in this work. Instead, we present a *template* that describes the process involved in the conversion of proprietary data by an InL-Adapter to DM4InL format. Then the template is validated using practical examples with RedPin and BluePin.

Figure 2.7 shows the configuration template of InL-Adapter. As seen in the figure, the role of the InL-Adapter is to convert the measured location data in a proprietary format into DM4InL format. To achieve this conversion, two types of data are essential depending on IPS: *measured data* and *master data*. The measured data is real-time location data (i.e., raw data) measured by the given IPS. The master data is static data specifying various configuration information of the IPS, such as users, devices, stations, buildings, and indoor maps. As shown in Figure 2.7, every InL-Adapter contains *data converter*, which defines a specific mapping from the measured data into DM4InL based on the master data.

To provide a more complete description I demonstrate the process of data conversion of BluePin and RedPin using instances of location data “L1” and “L2” (See Section 2.2.2). According to section 2.1.2, data items in “L1” and “L2” can be divided into three elements: time, object, position. With respect to the time information, it is easy to introduce the common representation in UTC.

For example, `2015/07/27 11:23:45 JST` in "L1" can be converted into `2015-07-27T02:23:45`. For "L2", since RedPin does not define a time attribute, we need to modify the RedPin client to add the timestamp to "L2" as `2015-07-27T01:11:01`.

For object information, a mapping is created from a proprietary ID of the located object into an object ID. For instance, the person ID `niu` in "L1" of BluePin is bound to the object ID of DM4InL. Using the master data of BluePin, other data items of the object model can be filled. On the other hand, the macAddress in "L2" of RedPin can be mapped to the object ID of DM4InL, since it is a unique string.

The conversion of position information is complex. For "L1", we need to convert the symbolic information "22" and "S101 Entrance" into a spot in the building model of DM4InL. In addition, the spot should be represented by a local point. For this, we use the master data of BluePin to look up the detailed location information of `22`. Considering the detailed information point positions (3.50m, 5.5m, 1.5m) of a building `B001`. A spot "S101 Entrance" is then created in `B001` with coordinate (3.50, 5.50, 1.50). Finally, a mapping is defined from "L1" to the spot.

However, as seen in "L2", RedPin represents the position based on 2-dimensional coordinates over a given map, i.e., an image of the floor plan. Therefore, multiplying the coordinates by the map scale derives the actual X and Y offsets. The Z offset can be derived from the altitude of the floor. Thus, the coordinates of a local point can be calculated. The spot information can be derived from the meta-data of the floor map. For instance, suppose that `KU-System-1F` represents a map of the first floor of building B001 with altitude 1.5m and that the map scale is 1/51.6. Then, "L2" is converted into a spot bound to a local point (6.70, 10.44, 1.50).

Based on this conversion, the heterogeneous measured data "L1" and "L2" are converted into DM4InL format shown in Table 2.1, 2.2, 2.3.

Table 2.1. LocalPoint

Pcode	x-offset	y-offset	z-offset	Building-Seq
P001	3.50	5.50	1.50	B001-01
P002	6.70	10.44	1.50	B001-02

Table 2.2. Spot

BuildingID	SpotID	SpotName	PointCode
B001	s00001	S101-Entrance	P031
B001	s00002	S103	P001

Table 2.3. ObjectLocationLog

ObjectID	P-Code	DateTime
niu	P001	2015-07-27T02:23:45
08:60:6e:32:b6:0b	P002	2015-07-27T01:11:01

An InL-Adapter has been developed for RedPin and its client was modified. The modification of the RedPin Android client required approximately 418 lines of code, and the InL-Adapter of RedPin required approximately 536 lines of code. The technologies used for the implementation are as follows: **Language:** Java 1.7.0, **Database:** MySQL 5.1, **Web server:** Apache Tomcat 7.0.57, **Web service engine:** Apache Axis 2 1.6.2.

2.2.5 Inl-Query for Operation Integration

Overview

To achieve operation integration, the proposed WIF4InL implements InL-Query (Indoor Location Query Service). InL-Query is a Web service that provides application-neutral API for querying indoor location data stored in InL-Database. It is meant to be deployed on the web.

According to the data schema of DM4InL, we develop two types of API for InL-Query: *fundamental API* and *composite API*, as mentioned in Section 2.2.3 The fundamental API includes an interface for querying entities within a signal model individually: location, building or object model. The details will be described in Section 2.2.5. The composite API allows advanced queries accessing multiple models simultaneously, which will be explained in Section 2.2.5.

Fundamental API

DM4InL represents the location information of every indoor object using the three models: location, building and object. (See Section 2.1). Depending on the model to which a given query belongs, we define three groups of API: *location query API, building query API*, and *object query API*.

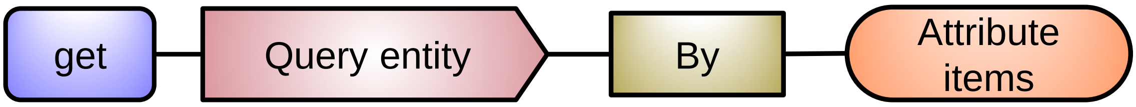


Fig. 2.8. Derivation Process of Building and Object Query API

Table 2.4. Detail Entity and Attribute Item Table

QUERY ENTITY	ATTRIBUTE		ITEM
BUILDING	Theme Attribute		BuildingID, Name, Type
	Spatial Attribute		GPID
PARTITION	Theme Attribute		PartitionID, Name
	Spatial Attribute		SpaceCode, BuildingID
ROUTE	Theme Attribute		RouteID, Name
	Spatial Attribute		LineCode, BuildingID
SPOT	Theme Attribute		SpotID, Name
	Spatial Attribute		PointCode, BuildingID
OBJECT	Theme Attribute	NULL	ObjectID, Type
		People	ObjectID, name, sex, …
		Appliance	ObjectID, Appliance Type, status, …
	Spatial Attribute		PointCode
	Time Attribute		Date Time

- Location Query API This API provides a set of methods (i.e., functions) for querying any entity within the location model. As shown in Figure 2.5, the location model consists of four entities. Each entity has a set of methods that returns appropriate instances based on given known attributes. The naming convention of the methods is `get[TargetEntity]By[given attribute]`. For instance, `getLPByPointCode(pointCode)` returns a local point designated for the given point codes.

  The local query API also facilitates methods querying *spatial relation* among geometric primitives in the location model. The spatial relation can be used to investigate how a spatial object in a space is located in relation to another object. In Location Query API, two types of spatial relationships are defined:

  - **Topological relation:** This represents the topological relationship between two objects. The operators that manipulate the topological relation include: `within`, `covers`, `coveredBy`, `intersects`, `touches`, `equals`, `disjoint`, `crosses`, `overlaps`. For instance, `getLPwitninLSP(LocalSpace)` returns all the local points within a given local space.
  - **Distance relation:** This represents the distance of an object from another object. The operators that manipulate the distance relation include: `at`, `nearby`, `vicinity`, `far`. For instance, `getLPnearbyLP(LocalPoint)` returns all local points that are nearby a given local point.
- Building Query API This API provides different methods for querying entities defined in a building model. In order to address all possible queries, methods are derived based on the structure shown in Figure 2.8. The figure shows that every method is constructed by varying *query entity* and *attribute items*. The query entity represents an entity to be returned by the method. Each entity has a designated set of attribute items that describes the entity from a theme or spatial perspectives. Table 2.4 summarizes the entities and the attribute items contained in the building model (as well as in the object

model). These methods are derived based on all the possible combinations of the entities and the attribute items.

For instance, `getBuildingByGPID(GPID)` returns a building identified by a given global position ID. In addition, `getRouteByName(buildingID, spotname)` searches for routes in a given building by its name.

- Object Query API This API provides methods for querying entities in the object model. Similar to the Building Query API, the methods are constructed based on the structure shown in Figure 2.8. The attribute items are shown at the bottom of Table 2.4. It is evident that time attributes exist for the object entity because an object usually occupies different locations with the progression of time. The methods are derived from all the possible combinations of the entities and the attribute items.

  For instance, `getObjectsAtPoint(pointcode, dateTime)` returns a set of objects that exist in a given local point for given date and time.

Composite API

The fundamental API allows only basic queries that are limited within a signal model. Hence, it is often too primitive to meet the sophisticated requirements of InL-App, which requires developers to integrate multiple API manually. For example, to implement the query "Who is in Room S101?", the developer need to integrate the building query API and the object query API. This motivated the development of the *composite API*, which facilitates high-level queries by internally combing fundamental APIs.

Several methods for the composite APIs have been derived based on typical use cases of location query in InL-App to minimize development cost and effort. Investigating typical use cases, three types of composite API have been developed. The first type is *building−object* API, which returns geographic elements of a building based on known information of an object. For instance, `getPartitionContainObject(objectID)` returns a partition that contains a given object. This method is implemented by re-using multiple methods of the fundamental API, specifically:

```
Partition getPartitionContainObject(objectID)
 Object o=geObjectByID(objectID)
 LocalSpace ls=getLSPContainsLP(o.pointCode)
 Partition p=getPartitionByCode(ls.spacecode)
   return p
```

The second type is *object–building* API, which searches for objects based on known geographic elements of a building. For instance, `getPeopleWithinPartition(bName, pName)` returns people within a partition `pName` of a building `bName`. This method can be implemented as follows:

```
Person [] getPeopleWithinPartition(bName,
 pName)
 Partition p=getPartitionByName(bName, pName)
 LocalSpace ls=getLSPBySpaceCode(p.spaceCode)
 LocalPoint[] lps = getLPcontainedInLSP(ls)
 Person [] H = empty
 foreach ls in lps
  Person h=getPersonAtPointIntim(
  lp.pointCode, NOW)
  push(H, h)
 return H
```

The last type is *calculation API*, which measures a specific metric among object and geographic elements using the spatial relations. For instance, `getDistanceBetweenSpotAndObject(buildingID, spotId, objectId)` returns the distance between a given spot and a given object. This method can be implemented as follows:

```
double getDistanceBetweenSpotAndObject(
```

```
buildingId, spotId, objectId)
Spot s=getSpotBySpotId(buildingId, spotId)
Object o=getObjectByObjectId(objectId)
double d = getDistanceBetweenLP(
  s.pointcode, o.pointcode)
return d
```

The implementation of the API is currently under way. The technologies used for implementation are as follows:

- **Language:** Java 1.7.0,
- **Database:** MySQL 5.1,
- **Web server:** Apache Tomcat 7.0.57,
- **Web service engine:** Apache Axis 2 1.6.2.

2.2.6 Evaluation

To evaluate the practical feasibility of WIF4InL, a comparative study was conducted among the three IPS: RedPin, BluePin and WIF4InL (that integrates RedPin and BluePin).

Capabilities for Location-Dependent Queries

The comparison is based on the sufficiency of *essential capabilities of location-dependent queries* [38]. Location-dependent means that any change of the locations of an object significantly affects the result of a query for the object. For example, suppose that a user *A* wants to find friends within a range of 100 m from *A* while navigating a shopping center. The result of the query depends on *A*'s current position, as well as the location of the friends. According to [38], the following capabilities should be specifically supported in indoor location queries:

- **Position Queries** return the locations of mobile and static objects and are processed according to either a geometric or a symbolic model of space.

Table 2.5. Comparison of Three IPS W.R.T. Capabilities of Location-Dependent Queries

IPS	Position	Navigation	Range	kNN	Time
RedPin	○	×	△	△	×
BluePin	○	×	×	×	○
WIF4InL	○	△	○	○	○

- **Navigation Queries** encompass all queries that directly help the users to find and reach points of interest by providing them with navigational information, while optimizing certain criteria such as total traversed distance or travel time.
- **Range Queries** are used to find and retrieve information on objects of interest or places within a user-specified range or area.
- **k Nearest Neighbor(kNN) Queries** search for the k closest qualifying objects relative to a moving user with respect to his or her current location.

Moreover, Liu et al. [36] suggested that time is also an essential attribute for location-dependent query.

- **Time queries** search a target object and a location by time, or retrieves the time from an object and a location. Each query depends on a record that the object remained in the location at the time.

Result of Comparison

Table 2.5 compares the three IPS with respect to the aforementioned five capabilities. In the table, labels ○, △ and × represent that the capability is "satisfied", "partially satisfied", and "not satisfied", respectively. Initially, it is evident that all three IPS supports the position queries. Although their representations of the position are different, they all have methods for querying the indoor location of a target object.

The navigation queries cannot be supported by RedPin or BluePin. In both cases, topology information among multiple locations is not maintained. In addition, it cannot be derived from individual location data, because each location is represented as a pre-defined venue (not a position). However, once WIF4InL converts their location data into DM4InL, the topology can be defined using the coordinates of the locations. Using the topology, WIF4InL can support applications to implement navigation queries. However, it should be noted that the queries are limited to the spots or positions that are already registered in InL-Database.

WIF4InL binds indoor location with a three-dimensional coordinate in DM4InL. Using the topological and distance relations and the calculation API, range queries can be easily implemented. However, the range queries cannot be supported by BluePin because it specifies each location as a symbolic label, from which we cannot calculate the range. The location data of RedPin contains a two-dimensional coordinate on the map, from which we can calculate the distance between two locations. However, when two locations are represented in separate maps (e.g., different floors), the distance cannot be calculated. In that sense, RedPin cannot fully satisfy the requirement of range queries. The same discussion applies to the kNN queries because the essentials these queries are almost the same as those of the range queries.

RedPin cannot support time query because the data does not contain any time attributes. BluePin contains a time-stamp in the location data, while WIF4InL manages time-series data of ObjectLocationLog in DM4InL. Therefore, these two IPS can support time queries.

Based on the preceeding discussion, it is evident that WIF4InL supports more capabilities for the location-dependent queries. Through data and operation integration, WIF4InL even enhances the existing proprietary IPS. Thus, it is expected that application developers will be able to develop InL-App more efficiently and intuitively using WIF4InL.

2.2.7 Related Work

Our work is closely related to the intersecting fields of indoor location framework, indoor positioning platforms, and data modeling techniques for indoor spaces. A number of indoor positioning framework or platforms have been proposed so far.

Brachmann et al. [39] proposed a multi-platform software framework. It aims to manage sensor data from different smartphone platforms to better understand RSSI(Wi-Fi)-based and other sensor-based IPS. The key idea of this framework lies in the normalization techniques for individual sensor data including data from magnetometers, accelerometers and gyroscopes. Thus, the framework is limited for wireless IPS, which are part of indirect self-positioning systems (see Figure 2.6-(4)). It does not consider other IPS topology. In this sense, the application scope is narrower than WIF4InL.

Gubi et al. [40] presented a platform that can dynamically facilitate efficient location technologies. As a user moves around a building, the platform suggests a best-available indoor positioning method based on the current position of the user. The platform manages building data in the form of a symbolic map, and markup of associated RF infrastructure, Wi-Fi and Bluetooth. However, this platform assumes that applications manage their own maps individually. Therefore, it does not provide application-neutral API that can re-use the indoor location data over different applications.

INSTEO Inc. [41] presented an IPS technology that relies on optimal hybridization algorithms of multiple information sources. The data source includes power measurement of Wi-Fi, Bluetooth Low Energy signals, smartphone sensors (accelerometer, compass, barometer, and so on). However, this approach is similar to Gubi's, which focuses on the combination of location technologies at the IPS level. Neither of them aims to achieve the loose coupling between IPS and InL-App.

2.2.8 Summary

In this section, a web-based integration framework called WIF4InL has been proposed to achieve data and operation integration for heterogeneous IPS. To realize this objective, WIF4InL implements InL-Adapter which provides different adaptation patterns for different system topology of IPS. InL-Query, which provides fundamental API and composite API based on the data schema of DM4InL has also been implemented. WIF4InL contributes to loose coupling of IPS and

InL-App, which will significantly improve the efficiency and re-usability in the InL-App development. The proposed framework has been applied to integrate two existing IPS, RedPin and BluePin. In addition, the WIF4InL has been evaluated by investigating the sufficiency of the five capabilities required for location-dependent queries.

2.3 Autonomous SensorBox

Autonomous sensorBox (referred to as *SensorBox*) is an IoT device with multiple environmental sensors which was developed by our research group [42]. It can measure seven environmental attributes around the box including temperature, humidity, lighting intensity, atmosphere pressure, sound volume, human motion and vibration, every 10 seconds. Figure 2.9 shows its physical form. It was designed to minimize cost and configuration labor. Once connected to power and a network, SensorBox autonomously measures environmental attributes in the vicinity of the box and uploads the information to a cloud server. Thus, all the operations for deployment and maintenance are performed without human intervention or expensive infrastructure.

Fig. 2.9. Prototype of SensorBox

Figure 2.10 shows a screenshot of the raw sensor data that be modified to JSON formal text. Figure 2.11 shows a screenshot of an application that represents collected raw sensor data on cloud service. It can be readily determined that changes in environmental attributes are man-made.

```
{
        "_id" : ObjectId("598c1de22c2f3f759203b90e"),
        "info" : {
                "boxid" : "sbox-phidget-149180",
                "date" : "2017-08-10",
                "location" : "CT001/H00003/R001/キッチン",
                "owner" : "longniu",
                "time" : "2017-08-10T17:48:29+09:00",
                "timeOfDay" : "17:48:29"
        },
        "data" : {
                "motion" : "false",
                "sound" : "55.369951428718224",
                "humidity" : "49.95379999999999",
                "temperature" : "34.444599999999994",
                "vibration" : "495.0",
                "light" : "1",
                "gasPressure" : "97.39113043478261",
                "presence" : "0"
        },
        "tag" : "iot.sensorbox.sbox-phidget-149180",
        "time" : ISODate("2017-08-10T08:48:29Z")
}
```


Fig. 2.10. Raw Sensors Data

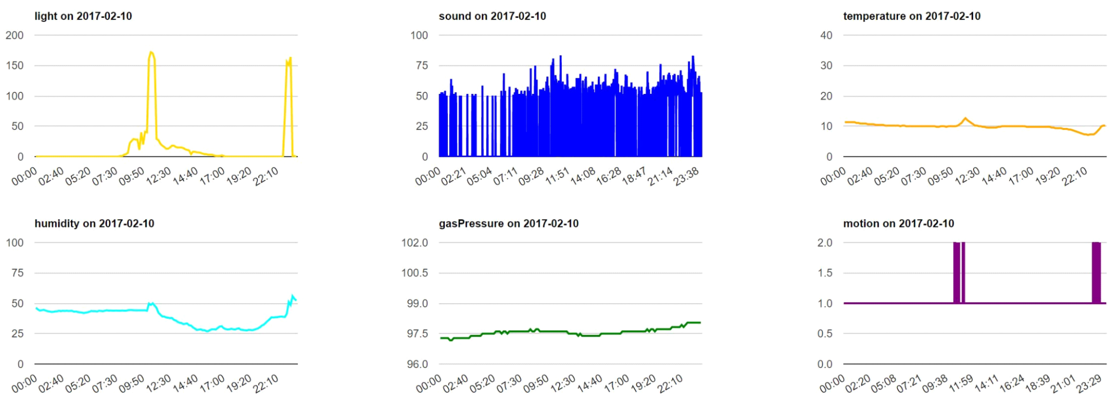


Fig. 2.11. Screenshot of SensorBoxLogService

Chapter 3

Recognition of Daily Activity

3.1 Recognizing Activity based on Non-intrusive Environmental Sensing

3.1.1 Introduction

In the world, the number of people living alone is continuously increasing. Because people in OPHs are susceptible to the loss of control of the healthy life rhythm, and because a chaotic life rhythm often leads to a deterioration in health, it is essential to maintain a good life rhythm, especially in the context of OPHs. In general, a life rhythm is characterized by the activities of daily living. Typical activities in OPH include eating, taking baths, and sleeping. If the cycle of activities becomes very different from those associated with a healthy life rhythm, the resident is losing their life rhythm. To maintain a life rhythm, it is necessary to maintain a regular record of activities. However, keeping manual records requires discipline and patience.

To automate activity recording in OPH, pervasive sensing technologies combined with machine learning are promising technologies, because they can *recognize activities* from automatically measured data. There have been many studies based on activity recognition. Some approaches (e.g., [5] [6]) attempt to directly capture the living metrics using cameras, or microphones. However, these systems are too intrusive to the user in the sense that their daily living situation is under constant surveillance. There are also many studies that involve wearable sensors, and/or indoor positioning systems to recognize activities (e.g., [7] [9]). However, a wearable sensor is intrusive to the human body in that the user must always wear the sensor device at home. Because the home is a place where the user is free from tedious things. Indoor positioning is intrusive at home, in the sense that sensors and beacons must be installed into the house and objects. It is expensive to deploy and maintain these systems.

To address these limitations, a new system is proposed that recognizes activities of OPH based on non-intrusive environmental sensing with machine learning. In the proposed system, *SensorBox* (see section 2.3) is exploited, which was developed in our laboratory [42], and is designed to minimize the effort required for deployment and operation.

Given the proposed method based on *supervised machine learning*, the proposed system also requires *initial training*, where the resident manually records activities using a designated lifelog tool. The initial training is supposed to be performed over several days, to associate labels of activities with sensor data. In the proposed system, I define seven basic activities (cooking, PC working, cleaning, bathing, sleeping, eating and going out), which are the most typical activities for maintaining a life rhythm. For the labeled dataset, supervised learning algorithms are applied to construct a model of activity recognition for the house. For this purpose, careful *feature engineering* is performed to determine essential predictors that best explain the activities in OPH. Furthermore, we attempted to try several different classification algorithms to compare performance.

To evaluate the proposed system, one SensorBox has been deployed in the apartment of a single person, and an experiment was conducted for ten days. Experimental results show that the average accuracy of all the seven activities was approximately 87% for "Decision Forest" supervised learning. The accuracy of some specific activities was over 90%. From this result, it is confirmed that the proposed system achieves non-intrusive and practical activity recognition in OPH, using SensorBox.

3.1.2 Preliminary

Activities of Daily Living

In the field of health, activities of daily living (ADL) is a professional terminology that was originally used at a hospital. It is the minimum action required to maintain daily life such as sleeping, eating, taking baths, etc. It is used as an indicator of aging and the degree of disability. The discovery and recognition of activity is an essential function of the system that provides necessary assistance to the residents of an OPH. Based on the results of this process, the intelligent system can determine which action to take in order to support the resident's well-being and to understand their life rhythm based on the regular records of activities.

Related Work

Given that the need for activity recognition is great, researchers have investigated the development of several methodologies to tackle this problem. The approaches to activity recognition can generally be divided into two categories, depending on the type of contextual information that is analyzed. The first category uses multimedia data acquired using a video camera or microphone recordings, to directly capture key aspects of daily living. The second category uses time-series data measured using various sensors including accelerometers, gyroscopes, RFIDs, and power-meters sensors.

Multimedia data: Brdiczka et al. [43] proposed a smart home that captured videos of residents and processes the video to recognize activities. Although individuals have been generally resistant to at-home video monitoring [44], the acceptance of this technology in the home is increasing. On the other hand, processing the video is computationally expensive. The process relies on the initial tracking of the resident period to being captured and analysis of the appropriate video data.

Sensor data: Since video acquisition and audio exposes too much information related to our daily living, it is considered to be intrusive. Therefore, it is better to use passive information. Hence, most of the current research activities involve recognition using sensor data. Researchers have determined that combining different types of sensors is effective for classifying different types of activities.

Kusano et al. [7] proposed a system that derives life rhythm by tracking the movement of the elderly by using RFID positioning technology. They installed many RFID readers on the floor of a house, and ask participants to wear slippers with RFID tags. The readers captured the indoor location of the resident. The system then determined the life rhythm of the user from the time-series location data. However, it is difficult to determine the exact activity using movement history. As a result, the accuracy of activity recognition is low.

Munguia-Tapia et al. [8] focused on the interactions of a resident with an object of interest such as a door, a window, a refrigerator, a key, and a medicine container. Munguia-Tapia et al. installed state-change sensors on regular items to collect interaction data.

Philipose et al. [45] attached an RFID tag to some items (such as television, fridge, bed), and asked a participant to wear gloves with an RFID tag reader. When the participant was close to the item, the interaction was recorded.

Pei et al. [9] combined a positioning system and motion sensors of a smartphone to recognize human movements in natural environments. However, when switching on the motion-sensors, Wi-Fi and GPS simultaneously, the battery drain is very high. Another problem is that a user may not want to carry a smartphone at times at home, which is a limitation of collecting data.

Challenges and Research Goal

Activity recognition has been widely studied over the past few years. By keeping track of activities, a smart pervasive system can provide reminders to residents, and react to hazardous situations [46]. Most of these studies apply to elderly people, cancer patients, and ordinary families. However, there are few studies on One-Person-Households (OPHs). The unique characteristics of OPH are as follows: the resident is living alone and is often required to do everything by themselves. Typically, they do not want to change their way of living or pay for expensive systems simply to monitor activities.

As mentioned in Section 3.1.2, there are many existing systems that use wearable sensors, object-embedded sensors, or indoor positioning systems. However, it is considered that it is difficult for people in OPHs to accept these technologies, because they are often exaggerated and intrusive to their life. One can easily imagine that most residents will forget or give up on wearing sensors, since the home is the place where the resident attempt to make themselves comfortable. Although labs or companies can manage large-scale equipment, it is still too expensive to deploy in an OPH.

The goal of this research is to minimize the limitations of conventional approaches, and to achieve high-quality activity recognition of an OPH.

3.1.3 Outline of Proposed System

In order to achieve this research goal, a new activity recognition system for OPH was proposed. To minimize intrusion and cost, the proposed system relies on the environmental sensing by the *SensorBox* [42]. Figure 3.1 shows the architecture of the proposed system. Using the figure, we explain the proposed system from left to right.

The system was initially set up within a target OPH. A single (or multiple if necessary) SensorBox was deployed in a position where activities are well observed as environment measures. A software called *LifeLogger* was then installed on the user' s PC. To apply supervised machine-learning algorithms, the proposed system requires *training data* at the initial phase of operation. For this, LifeLogger is used to attach correct labels of activities (as lifelog) to the environmental sensing data.

Then, the system begins to collect time-series data. SensorBox uploads the measured data to MongoDB in a cloud server, whereas LifeLogger inserts the lifelog into MySQL in the cloud data.

Finally, the system joins the two time-series data with the timestamp to form the training data. Machine learning is applied to the training data to construct a prediction model of activity recognition.

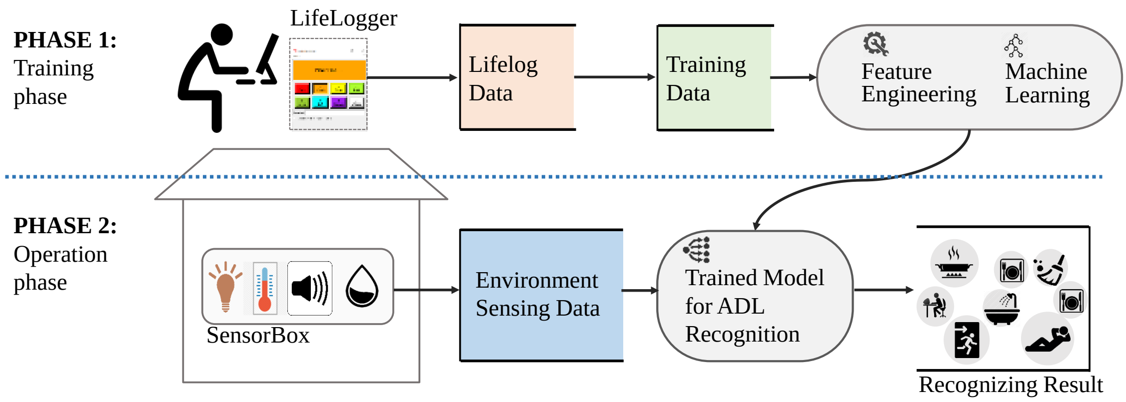


Fig. 3.1. Proposed System Architecture

3.1.4 Data Collection

Environmental Sensing

In the proposed system, *SensorBox* was exploited, which was described in detailed, in section 2.3.

To be able to detect activity by analyzing non-intrusive environment attributes, the target attributes must be sensitive to the changing of resident's ADL. Considering that the range of sensible is only around the SensorBox, the box should be put on where resident's activity is frequently conducted. However, the layout of each house and living circumstance of every single resident is different among OPHs. Hence, the most suitable position of SensorBoxdiffers for different OPHs.

Activity Labeling

During the initial several days, the resident needs to input *correct labels* for activities, so that the system can *learn* these activities from the environmental sensing data. For this purpose, the residents were asked to use *LifeLogger*. Figure 3.2 shows the user interface of LifeLogger. As shown in this figure, LifeLogger has 8 Buttons, each of which corresponds to an activity. When the resident initiates an activity, he/she simply depresses the corresponding Button to record the current activity. Based on relevant studies [47] [5], 8 types of typical activities were

Fig. 3.2. Screenshot of Lifelogger Tool

```
[2017/05/29T00:04:52]:<user@DESKTOP-G3JFFRR>:Button 2: Starting PCwork
[2017/05/29T01:16:36]:<user@DESKTOP-G3JFFRR>:Button 2: Ending PCwork
[2017/05/29T01:16:31]:<user@DESKTOP-G3JFFRR>:Button 8: Starting Others
[2017/05/29T01:19:38]:<user@DESKTOP-G3JFFRR>:Button 8: Ending Others
[2017/05/29T01:19:38]:<user@DESKTOP-G3JFFRR>:Button 5: Starting Sleep
[2017/05/29T09:55:37]:<user@DESKTOP-G3JFFRR>:Button 5: Ending Sleep
[2017/05/29T09:55:37]:<user@DESKTOP-G3JFFRR>:Button 8: Starting Others
```

Fig. 3.3. Raw Data of Life Log

chosen (sleeping, eating, bathing, cooking, PC working, cleaning, going out and other), and registered them in LifeLogger. When the resident depresses a Button, the system recorded the label and stored it in MySQL in a cloud server. Figure 3.3 shows the raw data of lifelog in the local PC. The tool's record of the start time and end time of the activities is evident.

Integration of Environmental Sensing and Activity Labeling Data

For supervised learning, the system required to training data that have a correspondence between the activities and data in advance. In order to establish training data, we integrate the two time-series data collected by SensorBox and LifeLogger by joining based on the timestamp. Since data labeled as 'other' was beyond the scope of the activity recognition, these noise data must be filtered. Table 3.1 shows the training data.

Table 3.1. Training Data

DateTime	vibration	light	motion	gaspressure	temperature	humidity	sound	activityID
2017/2/19 3:33:02	495	1	0	98.8	13.33	35.84	50.15	5
2017/2/19 3:33:12	494	1	0	98.8	13.33	36.04	0	5
2017/2/19 3:33:22	494	1	0	98.8	13.33	36.04	51.62	5
2017/2/19 3:33:32	494	1	0	98.8	13.33	36.04	0	5

3.1.5 Establishing Machine Learning Recognition Model

Analysis Activity-sensitive Environment Sensing Sensors

For accurate activity recognition, it is essential to identify the environmental values in the sensing data that best predict activity. From the seven environmental attributes of SensorBox, Only temperature, humidity, light, sound volume, and motion were chosen because the remaining attributes (vibration and atmosphere pressure) seem irrelevant to the target activities. According to compared about 20 recognition models based on different combinations of environmental attributes, the determination was made that sensing data of gasPressure and vibration is almost not affected by the resident's activity.

Feature Engineering

Feature value is the data that is effective in the identification of the activities. In this study, the feature values are obtained from training data according to the following process.

The size of time-window is first determined. To enhance the features of the time-series data, the raw data within the same time-window is aggregated into one data. In this case, the window size affects the accuracy. If the size is too large, the window is likely to contain different activities. If it is too small, the window will not contain sufficient data to reason and predict an activity. Hence, 3 variations of 1, 2 and 3 minutes were tested. In order to facilitate the discussion in Section 3.1.6, the symbols ('A', 'B', 'C') were used to present different datasets with different time-window sizes. The detail is such that **A:** 1minute, **B:** 2 minutes and **C:** 3 minutes.

Finally, for each of the five environmental attributes chosen, an *aggregation* *function* was determined. An aggregation function aggregates all the data within the same time-window. Typical, aggregation functions include Maximum value (MAX), Minimum value (MIN), Average value (AVG), Standard deviation (STDEV), and so on. Based on the nature of each environment attribute, an appropriate function was carefully chosen. A different aggregation function must be applied to each environmental attribute. By analyzing all the tests, the optimal combination of aggregation functions is determined. However, if all situations need to be tested, then hundreds rounds of tests need to be performed, which is time-consuming. To effectively tests all cases of function combination, a tool called PICT [48] was used. PICT generates a compact set of parameter value choices that represent the test cases required to achieve comprehensive combinatorial coverage of the parameters. Table 3.2 shows the 9 cases of combinations generated by PICT.

Table 3.2. Nine Groups of Aggregation Funcations

Groups	light	motion	temperature	humidity	Sound
G1	MIN	MAX	AVE	AVE	MAX
G2	MAX	MAX	STD	STD	STD
G3	AVE	AVE	STD	STD	MAX
G4	MAX	AVE	AVE	AVE	MAX
G5	MIN	AVE	AVE	STD	AVE
G6	AVE	AVE	AVE	AVE	STD
G7	MAX	MAX	STD	AVE	AVE
G8	AVE	MAX	AVE	AVE	AVE
G9	MIN	AVE	STD	STD	STD

Establishing Recognition Model

For the developed features of the training data, machine-learning algorithms are applied, to construct a prediction model for activity recognition. Popular classification algorithms are then used, including Logistic Regression, Decision Forest, and Neural Network. Using these algorithms, it is possible to construct predic-

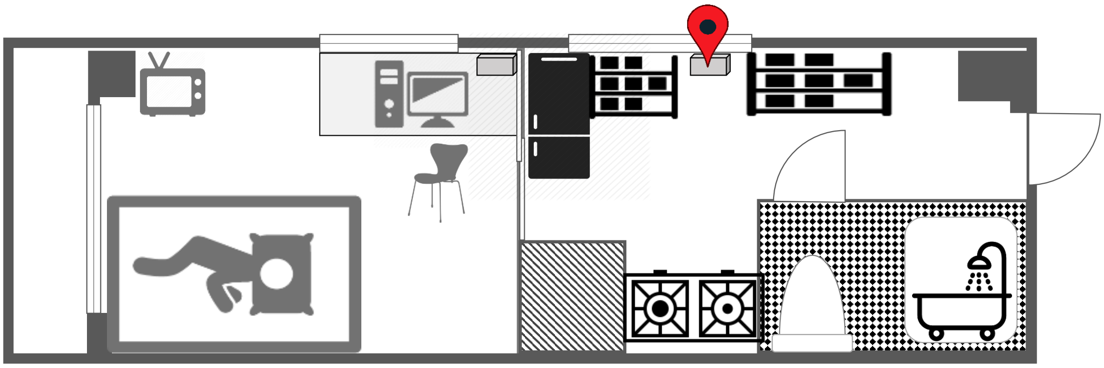


Fig. 3.4. Apartment of Testbed, Position of SensorBox

tion models that classify given environmental sensor data into one of the seven activities.

3.1.6 Evaluation of Experimental

Experimental Setup

The proposed system was deployed in an actual apartment of a single resident. As shown in Figure 3.4, the apartment is an ordinary apartment in Japan, consisting of a bed/living room, a bathroom and a kitchen. A single SensorBox was placed in the kitchen room so that this device could also observe the activities of the resident. The position of SensorBox is represented as a red pin in Figure 3.4. A total of 45,693 rows of labeled sensor data which did not include the data labeled with 'other', was collected during 10 days within the apartment.

Result

A total of 81 recognition models were established based on the 3 sizes of time-windows, 9 combinations of *Aggregation Functions* and 3 machine-learning algorithms. These models were tested by training and learning the collected data. In this subsection, we show the test result of all the models, the *Average Accuracy* of each trained activity recognition model. Accuracy measures the goodness of a classification model as the proportion of true results to the total numbers of cases. Average accuracy is the average of each accuracy per class (sum of accuracy for each class predicted/number of class).

In order to facilitate the observation of each model's performance, all the models were divided into three tables according to the size of time-window, Table 3.3, 3.4 and 3.5, and draw a bar graph for each table, Figure 3.5, 3.6 and 3.7.

For the tables, each column represents the method of data processing in feature engineering and is identified by a name that contains two characters, a capital letter and a number. The capital letter means the size of time-windows, as mentioned in the subsection Feature Engineering. The number represents the combination of the group number of the aggregation functions in Table 3.2. In an example in Table 3.3, A4 indicates that the size of time-window is 1 minute and the utilized combination of Aggregation Functions is G4 for the data process of feature engineering. The row of the table is identified by the name of the machine-learning algorithm. Each cell indicates the average accuracy of each model. In one example, the average accuracy of the recognition model exceeded 88.10% for the case where the size of time-window was 3 min, the utilized combination of aggregation functions was G4 and the algorithm is the multiclass decision forest.

For the three bar graphs, the vertical axis represents the average accuracy and the lateral axis represents the column of the relevant table. The color of the bar represents the algorithms, the row of the relevant table.

By comparing the average accuracy of models for different time-window sizes such as the blue bars of A1, B1, and C1, it is evident that the size of time-window slightly influence the average accuracy of the model. By comparing the results of models on different methods of feature engineering in one graph, it can be seen that the models have significantly different performance for different combination of aggregation functions. By observing the three graphs of table, Figure 3.5, 3.6 and 3.7 it can be seen the models utilized multiclass decision forest, represented by the orange bars, have better performance than the other models.

3.1.7 Evaluation

In this subsection, the relationship between the three factors and the accuracy of recognition activities based on the part of the representative data selected from

Table 3.3. All Results for Time-Windows of One Minute

Multiclass Algorithm	A1	A2	A3	A4	A5	A6	A7	A8	A9
Neural Network	84.36%	83.98%	84.27%	83.95%	83.96%	84.61%	84.28%	85.27%	84.23%
Decision Forest	85.83%	86.56%	86.24%	83.91%	84.43%	86.78%	87.54%	87.83%	86.41%
Logistic Regression	83.51%	85.23%	85.21%	83.16%	85.73%	83.18%	85.92%	82.05%	85.22%

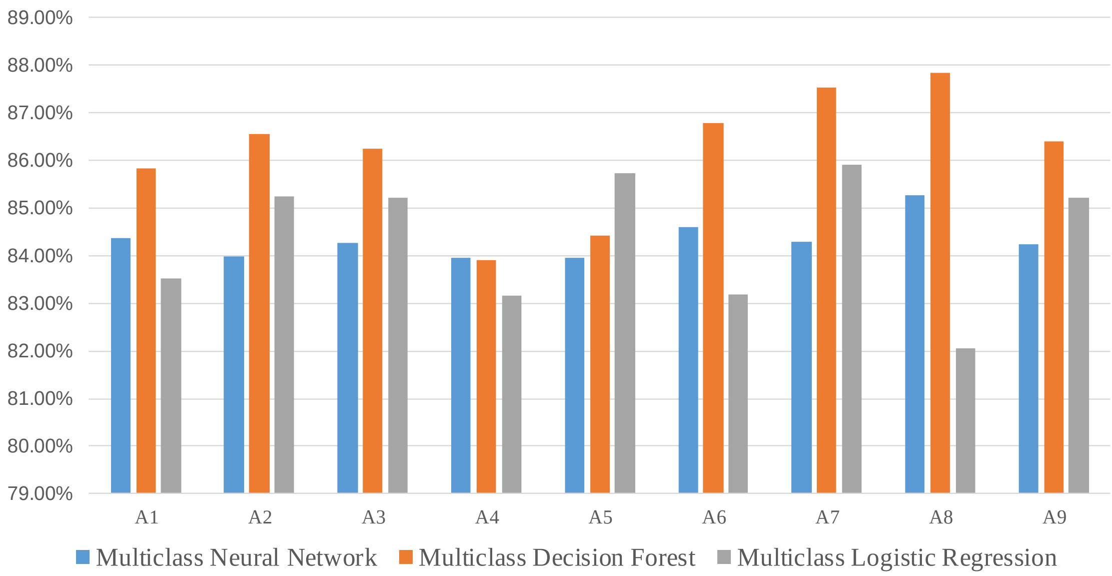


Fig. 3.5. Visualization of Table 3.3

Table 3.4. All Results for Time-Windows of Two Minute

Multiclass Algorithm	B1	B2	B3	B4	B5	B6	B7	B8	B9
Neural Network	84.10%	84.92%	84.87%	85.14%	83.08%	85.66%	84.45%	85.14%	85.09%
Decision Forest	84.52%	87.68%	86.18%	87.56%	84.62%	84.59%	87.30%	83.72%	86.49%
Logistic Regression	83.57%	85.79%	85.14%	83.05%	85.57%	83.92%	85.94%	82.53%	85.71%

Table 3.5. All Results for Time-Windows of Three Minute

Multiclass Algorithm	C1	C2	C3	C4	C5	C6	C7	C8	C9
Neural Network	84.62%	85.14%	85.07%	84.99%	82.11%	85.44%	84.25%	84.29%	84.70%
Decision Forest	85.92%	86.44%	85.10%	88.10%	85.25%	84.62%	87.10%	84.66%	86.21%
Logistic Regression	83.00%	86.77%	84.99%	82.59%	85.44%	84.29%	85.99%	82.59%	86.44%

huge volumes of experimental data is evaluated.

The effect of the time-window on the accuracy is initially evaluated. Table 3.6 shows the accuracy of three recognition models for several activities (cooking, sleeping, and eating). The three recognition models utilized the same aggregation function and the algorithm expects the size of time-window. From the results, it can be seen that the accuracy of the three activities is slightly affected by the

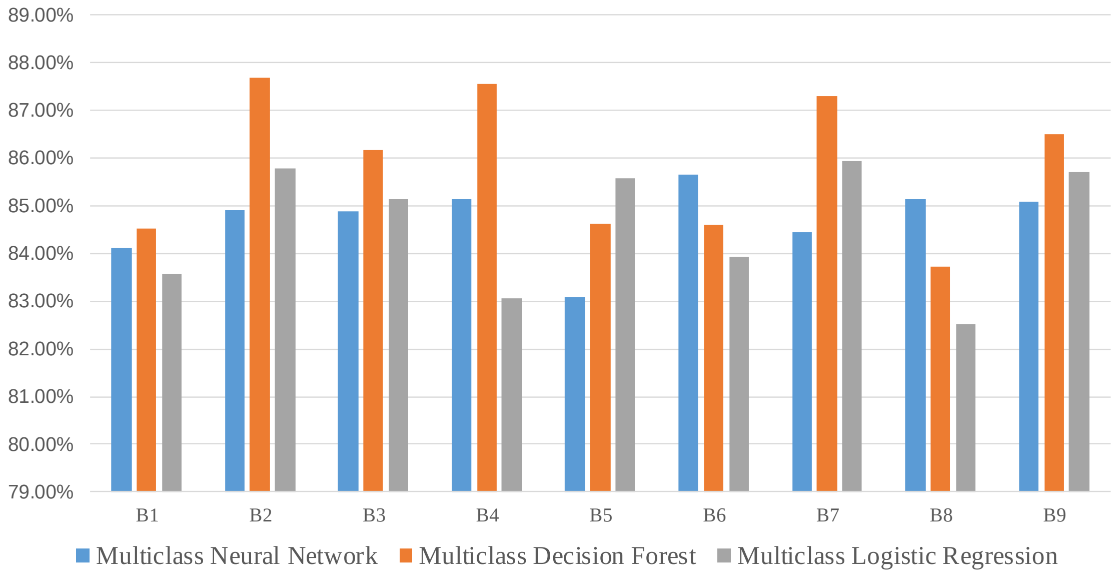


Fig. 3.6. Visualization of Table 3.4

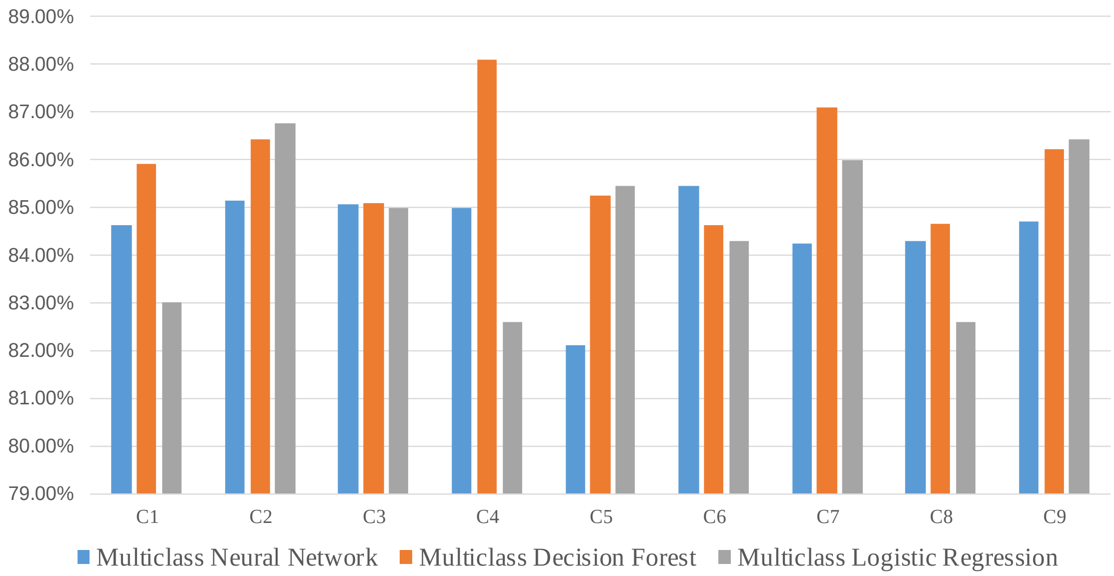


Fig. 3.7. Visualization of Table 3.5

size of time-window, and the 2 min is likely an appropriate value because the accuracy of recognition activities is the highest in this case, expect eating that was recognized in the highest accuracy when time-window was 3 min. The change in accuracy can be caused by a very small change in the size of the time-window, which confirms the aforementioned view that the size of time-window should not be too large or too small.

Table 3.6. Comparison of The Accuracy of The Three Models for Three Time-Windows

Time windows	Cooking	Sleeping	Eating
1 minute (A)	86.00%	89.90%	54.10%
2 minutes (B)	89.70%	89.90%	55.70%
3 minutes (C)	87.80%	89.70f%	55.90%

The effect of the aggregation function on accuracy was then examined. Table 3.7 shows the accuracy of three recognition models for the recognition of several activities (cleaning, sleeping, and going out). The three recognition models utilized the same sized time-windows and algorithms, except that the combination of aggregation functions for the five environmental attributes. From the result, it is evident that the aggregation functions have a great influence on the accuracy of each activity recognition. For the G4, the accuracy of going out recognition is only 17.6%, which is 45% less than G8. However, the accuracy of predicting sleeping is almost equal to G8. For the three models, the accuracy of sleep recognition achieves the highest value using the combination of G7. However, the model of G7 performs poorly for the recognition of cleaning and going out. For G8, the accuracy of predicting sleeping is less than G7, but the performance for predicting cleaning and going out is much better than G7. Hence, from these comparisons it can be seen that the system need to apply different combinations of functions to each activity, i.e., the feature value for various activities is different.

Table 3.7. Comparison of The Accuracy of Three Models on Three Aggregate Functions

Aggregate Function	Cleaning	Sleeping	Absence
G4	47.10%	73.00%	17.60%
G7	39.40%	95.50%	21.10%
G8	62.60%	72.70%	62.40%

Finally, the effect of the algorithms on accuracy was investigated. Table 3.8 shows the accuracy for three recognition models for recognition of several activities (cooking, sleeping, and eating). Those models utilized the same time-window size and combination of aggregation functions, except for the algorithms. From these results, it is evident that Decision Forest performs better than Logistic Regression for the recognition of the three activities. In the case of the Neural Network, it has the best performance for predicting sleeping and eating, but it has the worst performance for predicting cooking. Based on the abnormal result, it appears that the amount of data used is not suitable for a neural network. In practice, the amount of cooking and eating data used was only 3.8% and 5.2% of the total data, or less than 400 elements. For these three models, Decision Forest exhibited a more robust performance compared to than others models with a limited amount of training data.

Table 3.8. Comparison of The Accuracy of The Three Models With Three Algorithms

Multiclass Algorithm	Cooking	Sleeping	Eating
Logistic Regression (LR)	59.60%	44.30%	41.30%
Decision Forest (DF)	67.50%	72.70%	62.60%
Neural Network (NN)	20.20%	85.30%	93.50%

3.1.8 Summary

In this section, a new system is proposed that automatically recognizes activity in OPH. Considering the characteristics of OPH, the proposed system exploits only environmental sensing by the SensorBox. This minimizes the cost of deployment, as well as the level of intrusion of the residents and their homes. To evaluate the proposed system, the system was deployed in an actual apartment of a single resident and collected sensor and lifelog data for 10 days. Using supervised learning with careful feature engineering, it was possible to construct practically feasible models for seven types of activities. The average accuracy of all activities was achieved by more than 88%. For sleeping recognition, the accuracy of recognition was more than 90%. Moreover, the influence of the time-windows, aggregation function and machine-learning algorithms on the accuracy of recognition activities was investigated.

3.2 Recognition of Activity Using Environmental and Indoor Location Sensing

3.2.1 Introduction

In Section 3.1, a system was proposed that recognizes 7 daily activities of residents in OPHs based on non-intrusive environmental sensing using machine learning. The non-intrusive environmental sensing data covers environmental attributes (sound volume, light, temperature, humidity, presence) that are collected by an IoT-based device called SensorBox. The system has been investigated in a real OPH[49]. More than a hundred *Activity Recognition* (AR) models for the system were established. Although the average accuracy achieved was approximately 90 and some special activities were recognized with accuracy in excess of 90%, the Macro-averaged recall [50] of most models was low at approximately 60% [51] and the Micro-averaged recall [50] of some models was approximately 75%. The accuracy of the recognition of some activities was very low. For example, the predicted accuracy of PC work was 18.2% and bathing was only 4%.

By analyzing the experimental results, it was determined that there are two main reasons for the unsatisfactory results of previous work. The first reason is that the system could not classify some activities using sensing environmental information of the entire house or apartment. On the contrary, the information for irrelevant rooms whereby some activities have a limited influence on the room environment, disturbed the process of activity recognition. For instance, when a user was bathing in a bathroom that was near the kitchen, the system incorrectly predicted that the user was doing PC work with a probability of 48% and sleeping with a value of 16%. However, the user performed PC work and Slept in the living room, which was far from the bathroom. The second reason is that environmental sensing data does not contain enough feature values for every recognized activity, which impacts of some activities for which the indoor environmental attributes are similar. For instance, when the user was sleeping in the morning, by analyzing environment sensing information (such as light, sound volume, presence etc.), the system couldn't distinguish between Absence and Sleeping. As a result, the system faulty predicted Absence in 11.6% when the user was sleeping.

In order to improve on the precision accuracy, it is necessary to remove undesired environmental information of irrelevant rooms for every activity and to collect more information created by residents to improve the feature value of each activity.

In order to achieve this objective, a single house or apartment was divided into at least two zones such that the resident always perform some special daily activities. The division is based on the arrangement of the houses or the rooms in a house or apartment, and the user's habits of daily activity. For example, the living zone is area where the user typically eats and sleeps and the kitchen zone is an area where the user typically cooks. A sensorbox was deployed in each zone. The system was then set to recognize activity by only analyzing the environmental sensing information of the zone where the activity occurred, instead of mining the information of all rooms. Moreover, the indoor location of the residents was sensed and the non-intrusive environmental sensing was integrated with the indoor location information for activity recognition.

For the proposed system, three research questions (RQ) were set for evaluation.

- RQ1: What is the percent improvement in accuracy when the proposed method is utilized?
- RQ2: Which multi-classification algorithm is more suitable for the recognition model?
- RQ3: How long can the recognition model achieve stable, high-quality of activity recognition for training?

In order to provide a rigorous answer to each research question, the proposed system was deployed in an actual setting and the apartment with a single resident (OPH). Experiment was conducted during the period May 29th 2017 to July 31st 2018. Valid data was retrieved for 31 days from the period. Approximately one hundred AR models were investigated in addressing the three research questions.

For RQ1, the result of comparison between the proposed system and the previous system shows that the former improved the macro-averaged recall by approximately 10and significantly improved the accuracy of several activity recognition such as sleeping, going out, and bathing. For RQ2, three popular multi-classification algorithms were then compared for AR models: Neural Network, Logistic Regression and Decision Forest. From the result of this comparison, it was determined that the Decision Forest is best suited for activity recognition compared to the other models. Finally, for RQ3, in order to determine the necessary time for the training phase for the utilization of the system, a detailed comparison of the 21 AR models were conducted with different length of training periods for the proposed and previous system. From the results, it was determined that the proposed system required less time for training phase than the previous system. The required training pahse of the proposed system was approximately 7 days while that of the previous system was approximately 15 days, until the system achieved acceptable stable and high quality.

Problems of Previous System

To achieve the research goal (see Section 3.1.2), an AR system based on non-intrusive environment sensing technology was proposed (see Section 3.1). Over one hundred AR models were established based on careful feature engineering to determine essential predictors that best explain daily activities in OPH. Furthermore, three classification algorithms were tested to compare their performances.

The system was deployed in the actual apartment of a single person and experiments were conducted. Experimental results show that the average accuracy of for all seven daily activities was approximately 90% and the accuracy of some daily activity recognition excessed 92%, as was the case for cooking, sleeping and going out. However, the other four daily activities could not be correctly measured, so the precision these activities was lower than 50%, which resulted in a Macro-average recall of around 60%. Macro-averaging represents the unweighted mean of precision, recall, and accuracy metrics [52]. Macro-averaged recall is more important than micro-averaged recall in this instance because the latter tends to weigh the most frequent daily activities heavily while macro-averaging considers all daily activities to be equally significant [53].

By analyzing the results of multiple trained models, it was determined that there are two main reasons for unsatisfactory performance. Firstly, the environmental information of irrelevant room in where human activity almost cannot influence disturbed recognition of some daily activities. Secondly, the environmental attributes are insufficient to distinguish between several basic daily activities. To illustrate these two reasons, the analysis of the results for a trained AR model is presented. Figure 3.8 displays the confusion matrix of the result.

The 4th row of the matrix shows that the system incorrectly predicted that user was doing PC work in 48% and sleeping in 16%, when the user was actually bathing in the bathroom near the kitchen. However, the user performed PC work and slept in the living room which was far from the bathroom. It was considered that this result was obtained because the system could not classify some special activities based on the environmental information acquired for the entire house or apartment. On the contrary, the information of irrelevant room where some activities almost cannot influence the room environment, which influenced the process

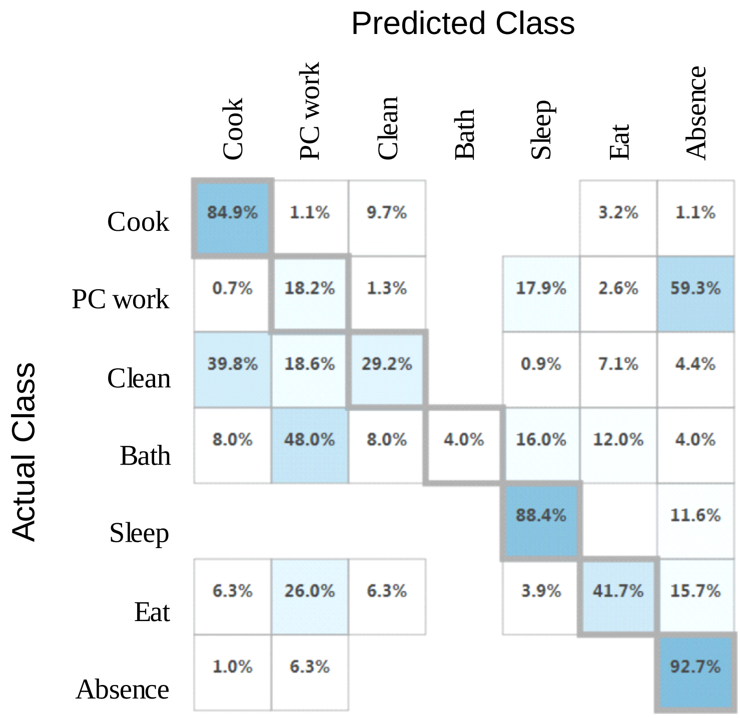


Fig. 3.8. Confusion Matrix of Predicted Result

of activity recognition. Hence, we should make system analyze environment of zone in where activity was occurred include irrelevant zone or room.

Then, the 2nd row of the matrix shows that the AR model cannot differentiate between "PC-working" and "Absence". When the resident is working in front of his desktop PC, the system predicted that no one was in the room with 59.3% probability and predicted that the user was sleeping with 17.9% probability. The 3rd and 6th rows of the matrix indicate that the system also fails to recognize cleaning and eating. Moreover, by observing the 5th row of the data, it is evident that the system also incorrectly concluded that there was no one in the room when the user was sleeping. Obviously, one reason for these results is that the environmental states that result from caused by the daily activities are similar. However, those daily activities can be easily identified by the user's position and motion information. For example, for PC working and going out, the current locations are of these two daily activities are obviously different. Therefore, it is necessary to incorporate new information to the previous system to improve the feature values of every activity.

3.2.2 Proposed Method

Key Ideas

In order to address these problems, it is essential that location information is collected for the residents as they perform daily activities. Hence, three key ideas were considered.

1. A house or apartment was divided into at least two zones where resident always performs special daily actives. The division of the house is based on the arrangement of houses or rooms in a house or apartment and the user's pattern of daily activity. For example, the zone of living is the area where the user always eats or sleeps and the zone of kitchen is the area where the user typically cooks. A sensorbox was deployed in each zone. We then classified activity according to the zone where the activity occurred, such as the activity of the living zone or the activity of the kitchen zone.

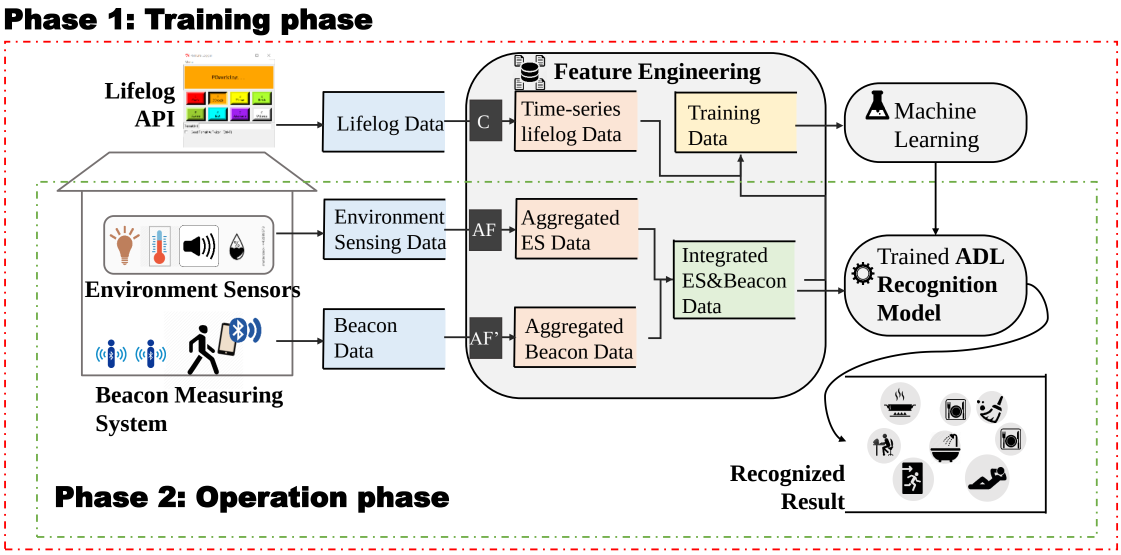


Fig. 3.9. Architecture of Proposed System

2. The system was set to recognize activity by only analyzing the environmental sensing information of the zone where the activity occurred, instead of mining the information of all rooms.
3. Regarding the measurement and collection of the indoor position of individuals, the precision of the measurement position information is much higher than that of the motion sensor data, which was only able to detect the presence of human activity. Therefore, it is necessary to install an indoor positioning system. In order to adhere to the original research goal mentioned in Section 3.1.2, a complex and expensive *Indoor Positioning System* (IPS) was excluded from the design and as such, we considered using BLE Beacon which allows smartphones and other devices to determine location by measuring the Received Signal Strength Indicator (RSSI) [54] transmitted from another beacon. Finally, the environment sensing data was integrated with the Beacon RSSI data with a timestamp, an ADLs recognition model was built based on the integrated information.

Architecture of System

Figure 3.9 shows the architecture of the proposed system. Based on the figure, a brief explanation of the function of each component and the data processing will be provided in this subsection.

The system was initially set up in a target OPH. A SensorBox was deployed in each zone and Beacon stations were introduced to measure the indoor position of the user. A *LifeLogger* software on resident's PC and a beacon measuring app 'BluePIN' on their smartphone. To implement the supervised machine-learning algorithms, the system requires training data at the initial phase of operation. This system consists of two phases: training and operation.

In the training phase, the user was asked to manually record lifelog data using the LifeLogger. The lifelog data was used to attach correct labels of activities to the environmental sensing data and the beacon RSSI data. Then, the system converts the raw lifelog data into time-series data using an application that is represented as a black square identified by a 'C'. The system converts beacon and sensor data using applications that are represented as a black square identified with 'AF' and 'AF', and then integrates the converted sensor and beacon data based on the timestamp and location. Next, the system creates training data by joining the time-series activity log and integrated data based on the timestamp. Finally, a multi-classification is applied algorithm to the training data to construct an AR model.

In the operation phase, the system automatically classifies the stream data into daily activities based on the AR model established in the training phase.

Data Collection

During the initial several days, the resident needs to input correct labels for different activities so that the system can learn their daily activities from the sensing data, which is similar to the previous system. To accomplish this, the residents were asked to use LifeLogger (see the Section 3.1.4), which logs the beginning and end time of an activity.

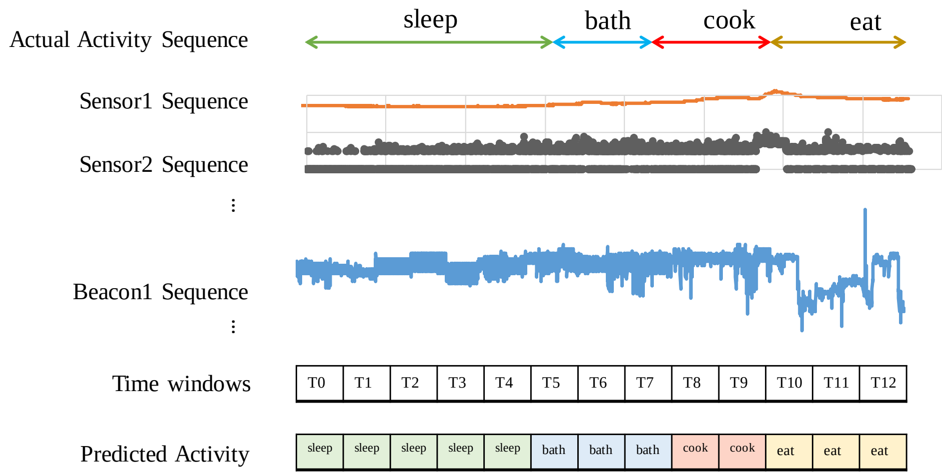


Fig. 3.10. Visualization of Raw data

Table 3.9. Raw Environmental Sensing Data

Light	Sound	Motion	Temperature	Humidity	Vibration	gasPressure	Presence	Datetime
0	68.61	false	23.7	34.7	495.0	98.2	0	2017-05-28T00:31:46+09:00
0	69.60	false	23.7	34.7	495.0	98.2	0	2017-05-28T00:31:56+09:00
0	68.09	false	23.7	34.8	496.0	98.2	0	2017-05-28T00:32:05+09:00
0	68.61	false	23.7	34.7	495.0	98.3	0	2017-05-28T00:32:15+09:00
0	69.60	false	23.7	34.7	495.0	98.3	0	2017-05-28T00:32:24+09:00
0	69.60	false	23.7	34.7	495.0	98.3	0	2017-05-28T00:32:34+09:00
0	69.60	false	23.7	34.8	496.0	98.3	0	2017-05-28T00:32:43+09:00
0	68.61	false	23.7	34.7	496.0	98.3	0	2017-05-28T00:32:53+09:00

For the collection of sensor data, considering that the sensitive range of these sensors is generally in the vicinity of the SensorBox, the box should be placed near the area where the resident frequently performs different activities. However, the layout of each house and the living styles of individuals vary significantly for different OPHs. Hence the most suitable position for the SensorBoxes also differs depending on the characteristics of the OPH. Considering that apartments normally consist of multi-zones for human activity, such as a living room for eating, studying and relaxing, and a kitchen for cooking, the SensorBoxes should be distributed in each main zone, as far apart as possible. Therefore, based on the location of the zone where the activities occurred, the system selects an appropriate SensorBox to measure the activity of a resident. The SensorBox measures seven environmental attributes, such as sound volume, lighting intensity, temperature, humidity, vibration, gas pressure and presence, every ten seconds. Table 3.9

Table 3.10. Raw Beacon Data

Lastupdate	Minor	RSSI
2017/05/29 01:18:51	03	-70
2017/05/29 01:18:51	03	-70
2017/05/29 01:18:51	03	-70
2017/05/29 01:18:51	03	-72
2017/05/29 01:18:51	03	-72
2017/05/29 01:18:51	03	-73
2017/05/29 01:18:51	03	-68
2017/05/29 01:18:52	03	-68

shows the raw sensor data obtained from a SensorBox using a MongoDB server by exporting the information as a CSV file. Figure 3.10 is a representation of two sets of sensors data.

To collect beacon RSSI data, a simple Beacon Measuring System is deployed in the user's home. In the system, multiple beacon stations are deployed in the apartment, and a mobile device with a beacon measuring app installed should be utilized. The RSSI of each beacon station is then measured and uploaded to the cloud server ten times every second. The position of the user can be calculated based on the RSSI value. Figure 3.10 shows the representation of the Beacon data and Table 3.2.2 shows the raw data of one beacon. The data is obtained from a MongoDB server by after being exported as CSV files. The ID of one of the beacon stations in the station group of one of the buildings is 'Minor'.

Feature Engineering

Feature engineering is the process of using domain knowledge about the data to create feature values that allow the machine learning algorithms to function. This is fundamental to the application of machine learning. The feature value is data that effectively identifies daily activities. In this report, a methodology is proposed for the Beacon data and the integration of the Beacon and sensor data. In this subsection, the data processing of feature engineering will be described in detail.

For lifelog data, a lightweight application was developed to convert semantic daily activity log data into time-series log data for each the second, thereby it is convenient to label sensor and beacon data with a timestamp.

For environmental sensing data, the feature value was obtained by the following process. The activity-sensitive environment attributes were initially analyzed. For accurate activity recognition, it is necessary to identify the environmental values in the sensing data that best predict the activities. Then, the size of the time-window is determined and the raw data is aggregated within the time-window into one dataset to enhance the features of the time-series data. Typical aggregation functions include MAX, MIN, AVG, STDEV, and so on. A detailed methodology of this process has been presented in section 3.1.5.

For the beacon data, the processing is similar to that of the environmental sensing data. The system extracts the feature value from aggregate data within the time-windows. Since the beacon signal is susceptible to changes in the environment such as humidity and the number of users present and is also easily reflected by surfaces (walls, ceiling, floors, etc.), many of the signals received by the mobile device are noise data. Hence, the first step is noise reduction, in particular, the filtering of signals reflected from surfaces. Signals that are farther away from the transmission are weaker signals. The max RSSI is extracted and saved in one second.

The aggregated sensor data and beacon data are then integrated based on this consistent time-window. To achieve the first key idea; the mining of the environmental attribute only in the room or zone where activity occurs, the sensor data and beacon data are integrated based on the location information, which is calculated by the RSSI. When the RSSI of one Beacon is larger than a defined value, then it is determined that the activity of interest is currently occurring in that room. The room's environmental sensing data is then integrated with the Beacon data.

Finally, training data is created by joining the time-series activity log data and integrated data based on the timestamp. Table 3.2.2 shows the real training data.

Table 3.11. Training Data

datetime	light	sound	temperature	humidity	presence	b2.ave	b2.min	b3.ave	b3.min	ADLid
2017/5/29 1:20:00	5.00	86.94	0.10	0.08	88.00	-58.35	-55	-66.94	-63	5
2017/5/29 1:20:30	6.00	88.22	0.00	0.00	67.00	-57.62	-57	-64.75	-62	5
2017/5/29 9:55:30	3.00	88.16	0.00	0.08	92.33	-56.35	-55	-67.60	-64	5
2017/5/29 9:57:00	163.00	17.65	0.00	0.08	78.33	-79.65	-686	-61.44	-52	4
2017/5/29 10:33:30	193.00	68.54	0.00	0.41	3.33	-78.37	-73	-63.00	-56	4

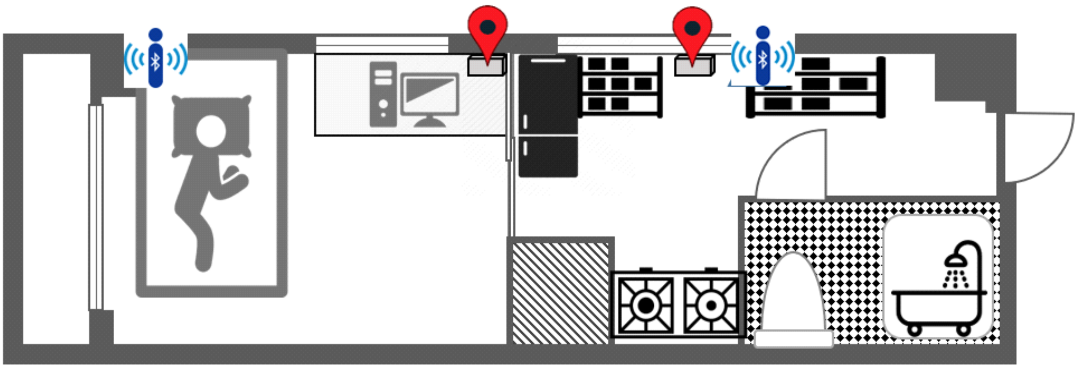


Fig. 3.11. Testbed

In the table, the 'b2.ave' represents the average RSSI value of the beacon whose Minor is '02'.

Establishing an AR Model

Machine-learning algorithms were applied to the developed features of the training data, in order to construct an AR model. A popular classification algorithm was used, such as *Multiclass Decision Forest Multiclass Logistic Regression* and *Multiclass Neural Network* [55]. By using the algorithm, a prediction model was constructed that classifies the given environmental sensor data and beacon RSSI data into one of the seven daily activities.

3.2.3 Evaluation of Experiment

Experimental Setup

The proposed system was deployed in an actual apartment of a single resident. As shown in Figure 3.4, the apartment is an ordinary apartment in Japan, consisting of a bed/living room, a bathroom and a kitchen. Two SensorBoxes were positioned as indicated shown by the red pins in Figure 3.11, one in the kitchen and one in the living room. For beacon stations represented by blue triangles, two stations were placed in the apartment, one at the head of the bed and one near the SensorBox in the kitchen.

A total of 645,705 rows of raw sensor data was collected from the kitchen SensorBox. The living room SensorBox collected 483,862 rows of raw data. The living room beacon collected 368,047 rows of raw data and the kitchen beacon collected 370,372 rows of raw data. For the feature engineering, the time window size was set at 30 seconds, and the aggregation function of environment sensing data at {Min(light), Ave(sound), Std(temperature), Std(humidity), Ave(presence)}. We applied *Multiclass Decision Forest*, *Multiclass Logistic Regression* and *Multiclass Neural Network* algorithms to build the AR model and learn the experimental data.

Evaluation

In order to evaluate the performance of the proposed system, three research questions (RQ) were mentioned as identified in Section 3.1.1. In this section, the three RQs will be examined based on three groups of experiments for the system.

For RQ1: "what will be the percent improvement in accuracy when the proposed method is utilized?" A comparison was performed between one propose AR model with a previous AR model. For RQ2: "which multi-classification algorithm is more suitable for the recognition model?" Three popular multiclass algorithms were evaluated: Decision Forest, Logistic Regression and Neural Network. Finally, for RQ3: "how long can the recognition model achieve stable, high-quality of activity recognition for training?" Detail comparisons were conducted for 21 patterns of length of the training period for the proposed system and the previous system.

Evaluation of Improvement of the Proposed Method

The same raw data and aggregation functions were used for the sensor and beacon data in the feature engineering, then the same algorithm was used to recognize the 7 types of daily activities. Figure 3.12 shows the confusion matrix of obtained for the previous work. Figure 3.13 shows the recognition results for the

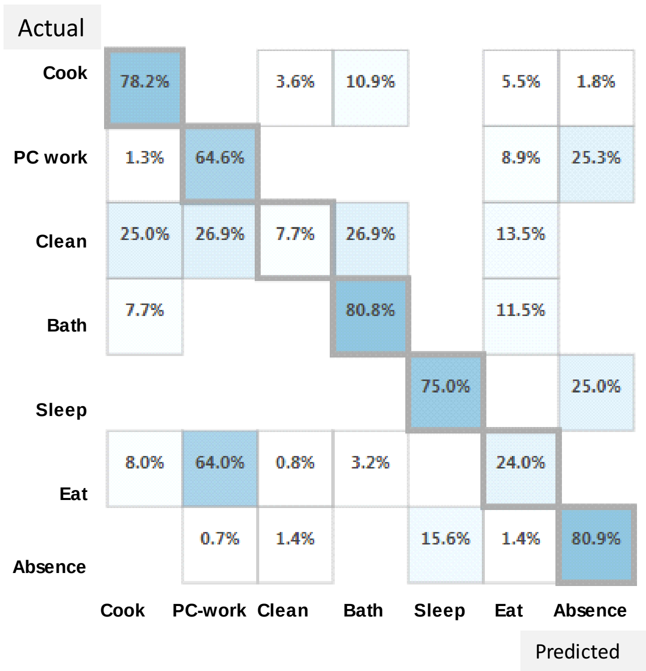


Fig. 3.12. Confusion Matrix for Previous Method

proposed method along with the labeled integrated data for training and testing. Figure 3.14 shows the precision of each of the daily activity recognition and the micro-averaged recall and macro-averaged recall for each experimental system.

By comparing the micor&macro-averaged recall of the system, it is evident that the new version, which uses integration data, performs significantly better on the 7 typical activity recognition.

From Figure 3.14 it was determined that the accuracy of cooking, sleeping and absence significantly increased when BLE-based location information was integrated. This may be because these three activities typically occur in the same zone.

However, when activities overlap in the same zone, such as PC working and eating, the accuracy of the proposed system is no better than the previous system. When only labeled sensor data are used, the precision of overlapping activity recognition is lower than that of the previous system. Here the location information is unable to classify these activities. The results are counterproductive.

When activities occur space across multi-zones, neither the previous nor the proposed systems can accurately classify them. For instance, the precision of

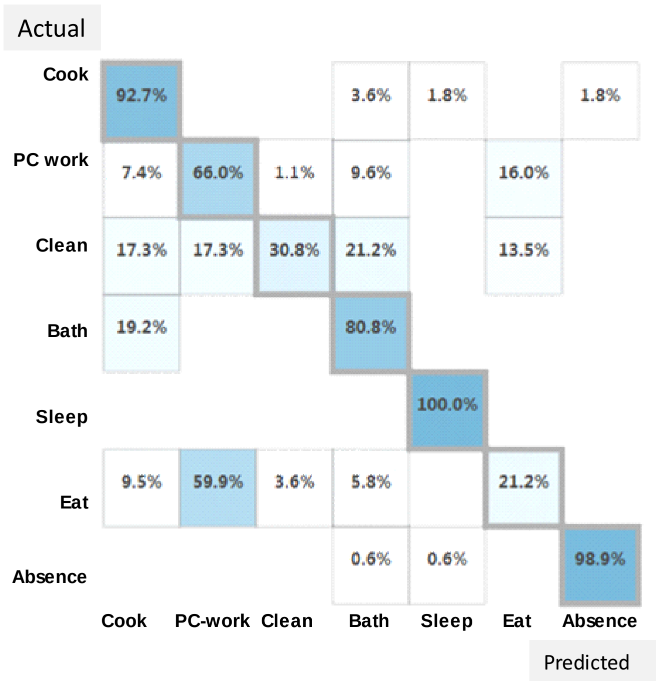


Fig. 3.13. Confusion Matrix for Proposed System Using Integrated Data

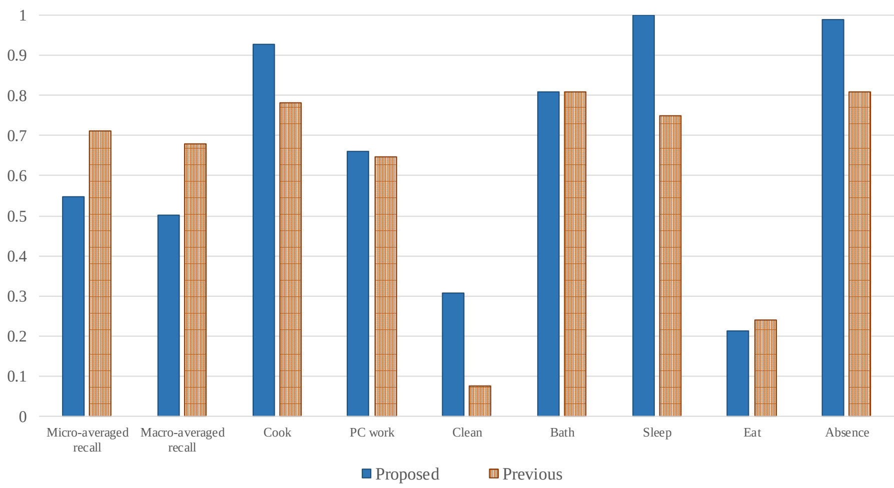


Fig. 3.14. Comparison Results for The Three AR Systems

cleaning recognition is likely slightly better than a random return.

Evaluation of 3 Algorithms

In this evaluation, a comparison of three popular multiclass supervised machine learning algorithms was performed: Decision Forest (DF), Logistic Regression (LR) and Neural Network (NN). To identified the most suitable algorithm for AR model, two groups of comparison were performed: the test group of the previous

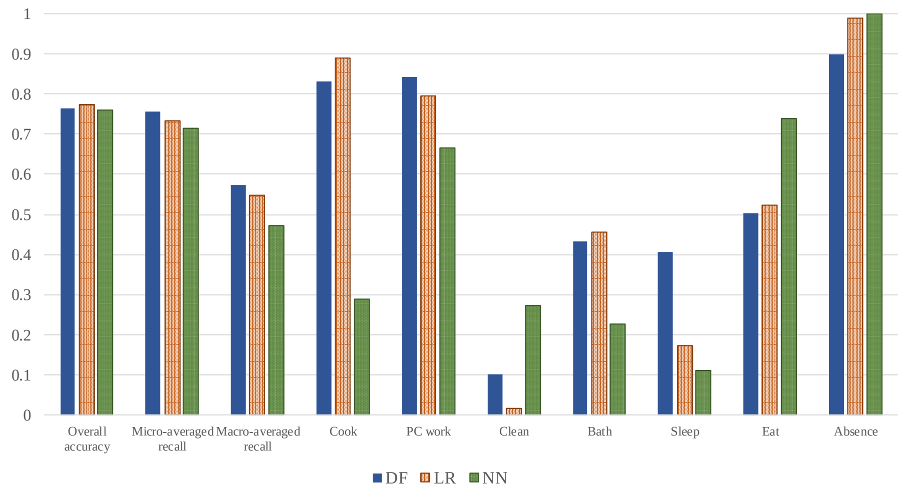


Fig. 3.15. Comparison of Algorithms: Previous Work

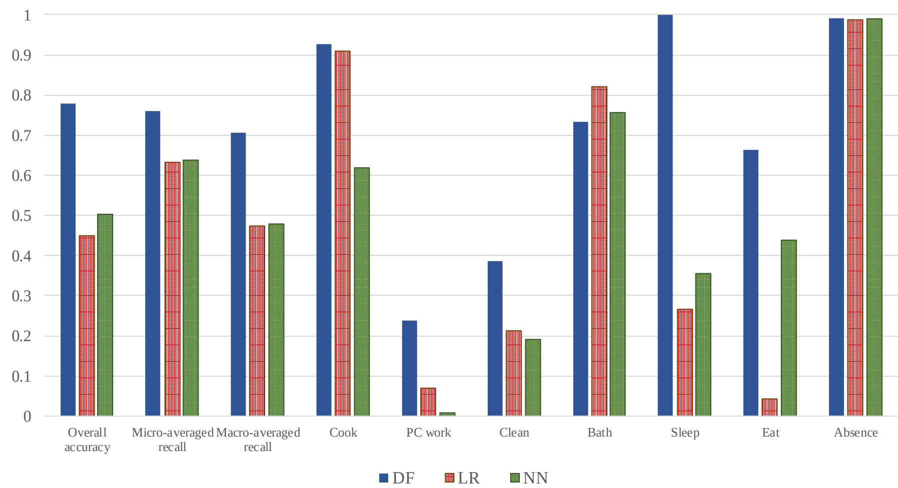


Fig. 3.16. Comparison of Algorithms: Proposed Work

work and the proposed work test group.

Figure 3.15 shows the result of the previous system using the three algorithms. Figure 3.16 shows the result of the proposed system for the three algorithms. By comparing the value of micro&macro-averaged recall, it was determined that the DF is significantly better than LR and NN.

Considering the performance of special activity recognition in the proposed system, DF recognized Sleeping, Eating, Cleaning, PC working and Cooking with higher accuracy than the others, however, PC working and Cleaning were recognized with a low accuracy of less than 40%. In the case of Bathing and Go-

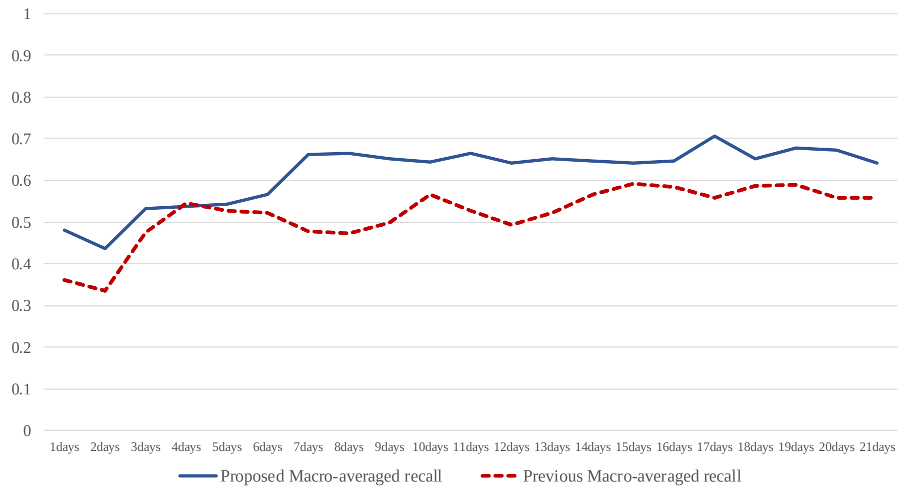


Fig. 3.17. Comparison of Macro-Averaged Recall with Different Length of Training Period

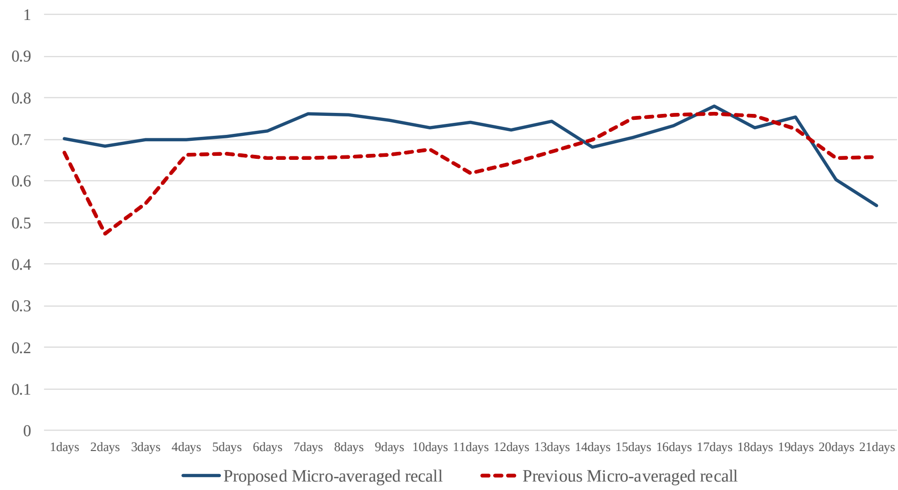


Fig. 3.18. Comparison of Micro-Averaged Recall with Different Length of Training Period

ing out, the prediction accuracy performance of the three algorithms were similar. For the performance of special activity recognition in previous systems, it was determined that DF had a clearly higher accuracy on Cooking and PC working recognition. On the contrary, using the proposed system, NN had a better performance than the others in Eating, Cleaning and Going out recognition.

Therefore, based on the general performance, it was determined that the DF is the best for the AR system.

Identifying Suitable Length of Training Period

In this evaluation, 21 lengths of training period were performed from 1 day to

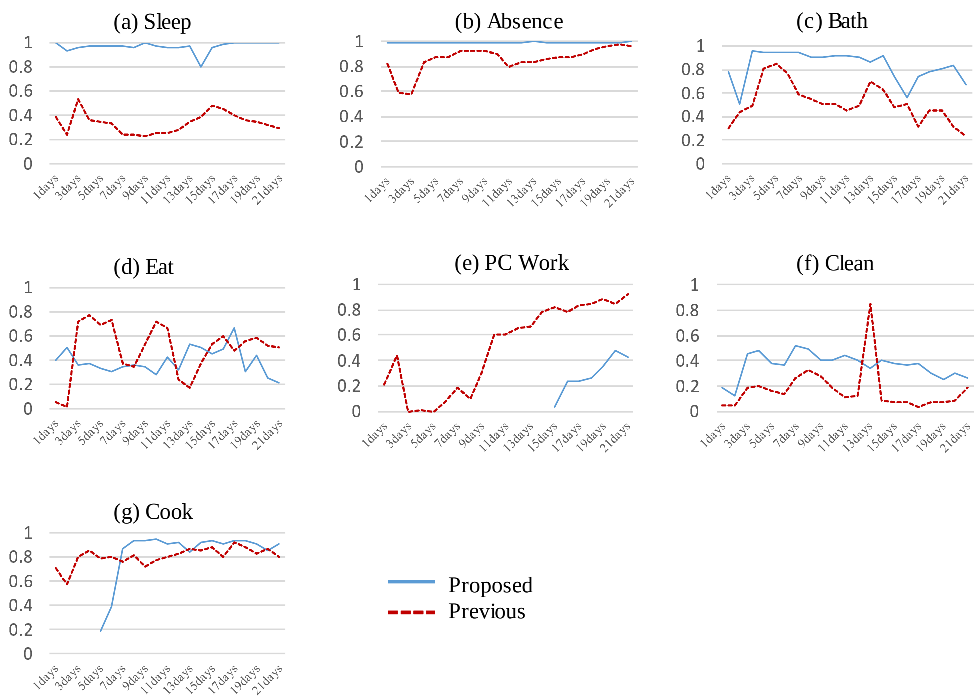


Fig. 3.19. Predicted Accuracy of Each Activity with Different Length of Training Period

21 days for the previous system and the proposed system. It should be noted that the duration of the training period in the experiment is actually one day and the recorded activity log is personalized. Given that the overall accuracy of both systems significantly decreases after 19 days of the training period, the comparison was until 21 days.

Figure 3.17 and 3.18 show the change of the Macro&Micro-Averaged recall of two AR systems with the change of length of the training period. For the two figures, the vertical axis represents the accuracy of the predicted and the horizontal axis represents the length of the training period. From the evaluation of the general performance, it was determined that the previous system achieved stable and high-quality of activity recognition after 15 days. However, the proposed system achieved stable and high-quality after only 7 days, which is approximately half of the previous system.

Figure 3.19 shows the change in the accuracy of each activity recognition with the change in the length of the training period.

From 3.19 (a) & (b), it was determined that the proposed system achieved a stable and high quality of activity recognition and required only 1 day for training. It is suggested that this is because the location information of sleeping and absence (Going out) is clearly different, so the proposed system can classify the two activities at almost 100% accuracy by training for only one day.

From 3.19 (c) it was determined that the proposed system achieved stable and good prediction after three days. However, after 15 days the accuracy of prediction decreased and an unstable status was achieved. This may have been caused by the change of the season. After Jun 20th 2017, the user habit for bathing changed, for example, the user changed the wash times each day from 1 to 2.

From 3.19 (d) & (e) & (f) it was determined that the proposed system did not achieved stable status after 21 days of training and performed with low accuracy of prediction. The reason for this may have been mentioned in the Section 'Evaluation of Improvement of the proposed method', the feature value of the activities were still not enough when only environmental features and the location of the resident were sensed.

From 3.19 (g) it was determined that the proposed system achieved a stable status later than the previous system, but the actual length of the training period for the proposed system was 3 days shorter than that of the previous system. The reason is that an error in the cooking log during the experiment using the proposed system, the resident forget to open the location application. Moreover, the application stopped for some program problems.

3.2.4 Summary

To address the limitations of the previous work, a new activity recognition system was proposed in the section. The proposed system recognizes a resident's activities by integrating non-intrusive environment sensing and zone-based indoor positioning. The environment sensing is implemented by a stationary IoT device, called Autonomous SensorBox. The indoor positioning uses RSSI of Bluetooth beacon signals. A resident first attaches activity labels to the sensor data. The system then constructs a recognition model based on supervised machine learning.

The proposed system has been applied to an actual one-person household. The combined use of environment sensing and zone-based indoor positioning well recognizes seven kinds of daily activities. Especially, the accuracy of Cooking, Bathing, Sleeping, Absence was over 80%. The proposed method outperformed the previous method (with environment sensing only) in the quality of activity recognition and the training period length. The experiment showed that the macro-average recall reached around 70% with just one-week training.

Chapter 4

Derivation of personalized assessment model for life rhythm

4.1 Introduction

Recently, due to the proliferation of smartphones and IoT technologies, there has been an increase in the number of studies and applications (e.g., [5] [6] [56]), which aim to support user health by capturing daily activity logs (e.g., sleeping, eating, cooking), or recognizing the pattern of a day (e.g., workday, hospital visit day etc,). This is very promising in the terms of the health management of individuals in OPHs. However, although most monitoring systems provide features for recording and visualizing activity logs, they do not facilitate the interpretation and assessment of the achievement of the acquired data. As a result, it is not easy for individuals to determine what their healthy life rhythm should be, and how to improve their current situation.

To address these limitations, a system is being developed that quantitatively assesses life rhythm based on the daily activity logs and the self-assessment of quality of life (QoL). The proposed system attempts to identify correlations between activities and the user's QoL, and then establishes a personalized model that explains the QoL based on the daily activities. Using the model, the user can more easily understand the state of their current life rhythm. Therefore, they are in a better position to modify habits to achieve a healthy life rhythm. It was eventually determined that the proposed system would be integrated with life monitoring system, so that the system automatically *intervenes* in OPHs to encourage the user to maintain or improve life rhythm.

In this chapter, the details of a study that proposes a method that establishes a personalized assessment model is presented. The method consists of the following three steps:

Step 1: Characterization of life rhythm using activities

Step 2: Measurement and recording of QoL

Step 3: Establishing an assessment model that maps the life rhythm onto the QoL

To implement the aforementioned steps, the corresponding technological challenges must also be addressed.

The first challenge is to collect activity data and to extract appropriate features to represent individual life rhythms. For this purpose, the sleeping and eating log data collected by a daily activity recognition system are used. This was developed in the second projects of my research (see chapter 3). From the log data, several statistical features can be calculated.

The second challenge is to measure the QoL. By its nature, the scale of QoL varies among individuals. Hence, a healthy life rhythm for one person is not necessarily healthy for another person. For example, for a patient, the scale is the health of his body. For a researcher, his scale may be quantity and quality of achievements. For this purpose, a system that requests that each user should perform a weekly survey was developed. In the survey, the user evaluates the fulfillment of daily living in last week. The degree of fulfillment is represented by a numerical value.

The third challenge is to derive a model for personal life rhythms. For this purpose, effective features were extracted from the statistics of daily activity logs and a regression model was derived that explains the QoL values based on the resulting statistics.

A preliminary experiment was conducted in an actual apartment, where activity logs for 224 days and self-assessment QoL logs for 32 weeks are collected. Based on the experimental results, the assessment model personalized for the resident was interpreted, and appropriate habits for maintaining a high QoL were identified.

4.2 Preliminary

4.2.1 Life Rhythm

The *life rhythm* is a cycle of life activities and biological functions, with a period of approximately one day [57]. Most of our biological functions (e.g., sleeping, waking) have daily periodic variations. We often perceive that our activities are controlled by a clock inside of our body. In the field of biology, this phenomenon is called the *circadian rhythm* (*dian* means day). The Nobel Prize for physiology and medicine in 2017 was awarded for proving that the circadian rhythm is controlled by molecular mechanisms.

According to [58], life rhythm is characterized by the following three properties:

1. Two basic states: Activities in the daytime and sleep in the nighttime.
2. Daily cycle: The two states are repeated periodically every day.
3. Diversity: The cycle is different among individuals. It adapts to *biology, environment, living society*, *individuality*, and *variability*.

4.2.2 Maintaining a Healthy Life Rhythm in OPH

A chaotic of life rhythm often leads to a deterioration in health. For instance, people with circadian rhythm disturbance have a higher risk of cardiovascular disease [4]. Sleep disturbances increase the risk of suffering from neutral fat [2]. Hence, maintaining a good life rhythm is very important for individuals.

However, people living in OPHs readily fail to manage their life rhythm, because there is often no one else to lend daily assistive care. For example, students living with families get up and go to beds earlier than those living alone [1]. With respect to the total number of meals skipping and breakfast skipping, people in OPHs have a significantly higher rate [2]. Thus, it is more difficult for individuals in OPHs to maintain a healthy life rhythm.

The life rhythm is a long-term variable consisting of many activities. Even manually recording the daily activities requires strong motivation. Thus, it is challenging to identify an optimal life rhythm for a person without any technological assistance. Hence, providing personalized assessment models of life rhythm is quite promising, since the model can explain the status of the current life rhythm. Moreover, based on the results of assessment, the user is able to easily find appropriate life rhythm and to understand how to improve his/her daily habit.

4.2.3 Related Work

There are many existing studies that address the issue of healthy life rhythm. Based on the type of approach used, they can be categorized into three groups as follows:

1. **Manual survey:** A study in the field of medical science measured life rhythms of patients based on a manual medical survey [59]. Life rhythm was determined based on physiological metrics including body temperature, the power of gripping etc. However, this approach cannot be applied in general households.
2. **Activity detection:** Many technologies are being studied to aid in the recognition of human daily activities at home. They are expected to be used in real-life and human-centric applications such as elderly care and health care. Some approaches (e.g., [5] [6]) attempt to capture daily living using cameras, or microphones directly. Other approaches use state-change sensors, and/or positioning systems to detect activities (e.g., [8] [9]). However, most of these systems only provide features for recording and visualizing activities. The interpretation and assessment are left to the individual users. Thus, it is not easy for the user to understand which patterns of activities leads to a healthy life rhythm.
3. **Life pattern recognition:** Several studies reported in [60] [56] attempt to detect the daily life pattern of residents by analyzing data from pyroelectric motion sensors. However, the result cannot facilitate the determination of the optimal life rhythm, since the patterns are characterized only by mo-

tions. Therefore, it does not contain more detailed but essential information related to life rhythm including sleeping and eating.

4.3 Research Goal and Approach

4.3.1 Research Goal

My research goal is to implement a life rhythm assessment system that quantitatively evaluates the quality of life. In this chapter, I focus on the most essential components of the system, considering the construction of a personalized assessment model. To construct this model, three steps were followed:

- **Step 1:** Characterization of individual life rhythms based on activities
- **Step 2:** Measuring and recording of the user's QoL data
- **Step 3:** Establishing of an assessment model that maps life rhythm onto the QoL.

Table 4.1. Daily Activities Log Data

username	date	startTime	endTime	DA[a]
niulong	2018/6/1	0:00:00	0:08:59	Others
niulong	2018/6/1	0:08:59	0:19:10	Bath
niulong	2018/6/1	0:19:10	0:33:52	Others
niulong	2018/6/1	0:33:52	8:05:34	Sleep
niulong	2018/6/1	8:05:34	8:12:18	Rise
niulong	2018/6/1	8:12:18	8:26:17	Others
niulong	2018/6/1	8:26:17	8:50:09	Eat
niulong	2018/6/1	8:50:09	9:41:00	Others
niulong	2018/6/1	9:41:00	23:59:59	GoOut
niulong	2018/6/2	0:00:00	18:09:16	GoOut

[a] 'DA' is an acronym for 'Daily Activity'.

4.3.2 Challenges and Approaches

To implement these three steps, there are technological challenges that should be addressed. In the following, we explain each challenge and the approach used to address them.

A1: How is the life rhythm characterized?

According to Section 4.2.1, the life rhythm is a cycle based on activities and sleep, and the cycle varies among individuals. Nevertheless, the life rhythm is constructed by a huge number of factors. Hence, it is difficult to establish a universal definition. It is thus challenging to *represent* the life rhythm based on specific features. Promising features include the bedtime, the period (duration) of sleep, and mealtimes. It is also challenging to collect such data for daily activities.

To address this challenge, a daily activity recognition system was used [61], which was previously developed in my second projects (see Chapter 3). Exploiting *SensorBox* [42] and indoor positioning beacons, the system is able to recognize seven kinds of daily activities (cooking, eating, cleaning, bathing, sleeping, going out, PC working). Table 4.1 shows log data for the daily activities recorded in an actual apartment.

This log data is exploited to represent life rhythm. According to [4], there are three relevant factors with respect to life rhythm: sleep, breakfast, and hormone levels. By managing these three factors, everyone can effectively adjust their life rhythm. However, based on the current technology, it is impossible to measure the hormone levels of a resident. Hence, it was decided that the log data related to sleeping and eating would be used. To represent life rhythm, the log data was aggregated in a certain period of days, and obtain relevant statistics. The statistics include the average waking time for a week and the number of days with breakfast skipping. In the proposed method, these statistics are called *features of life rhythm*.

A2: How is the life rhythm assessed?

The quality of the life rhythm is related to the QoL of the user. However, the sense of value of QoL varies among individuals. Therefore, it is considered that the assessment model should be built based on the user's subjective scale of QoL. To determine the user's subjective scale, the proposed method requires each user to conduct a self-assessment of QoL and to record the result.

For the self-assessment, the *fulfillment of daily living* was used. The aspect of fulfillment is determined by individual users, considering one's lifestyle and personal opinion on the meaning of life. A user may be happy if he accomplished many tasks. Another user may be satisfied if she is able to concentrate on her studies. The degree of fulfillment must be quantified by a numerical value, which can be either discrete or continuous. In the proposed method, the value of the self-assessment is refrred to as *the value of QoL*.

A3: How is the assessment model established?

Life rhythm varies among individuals, and the assessment scale of QoL is also different from one user to another. Therefore, it is unrealistic to define a common assessment model that fits all users. Therefore, the main idea, in this case, is to build a *personalized* model that can explain the personal value of QoL based on the personal features of life rhythm. To build the personalized model, correlation analysis is initially applied to identify strong features correlated to the QoL. Then, using these features, regression analysis was performed to derive equations that maps the life rhythm features onto the QoL value.

4.4 Proposed Method

4.4.1 Overview

Figure 4.1 shows an overview of the proposed system. As mentioned in Section 4.3.1, this method includes three steps. Step 1 represents the life rhythm using statistical values for the log data associated with sleeping and eating. Step 2 involves the collection of values of QoL by self-assessment, asking the user questions based on the fulfillment of daily living. Step 3 creates *labeled life rhythm* data by joining the life rhythm data and the QoL data and then derives a regression model that maps *features of life rhythm* onto the *value of QoL*.

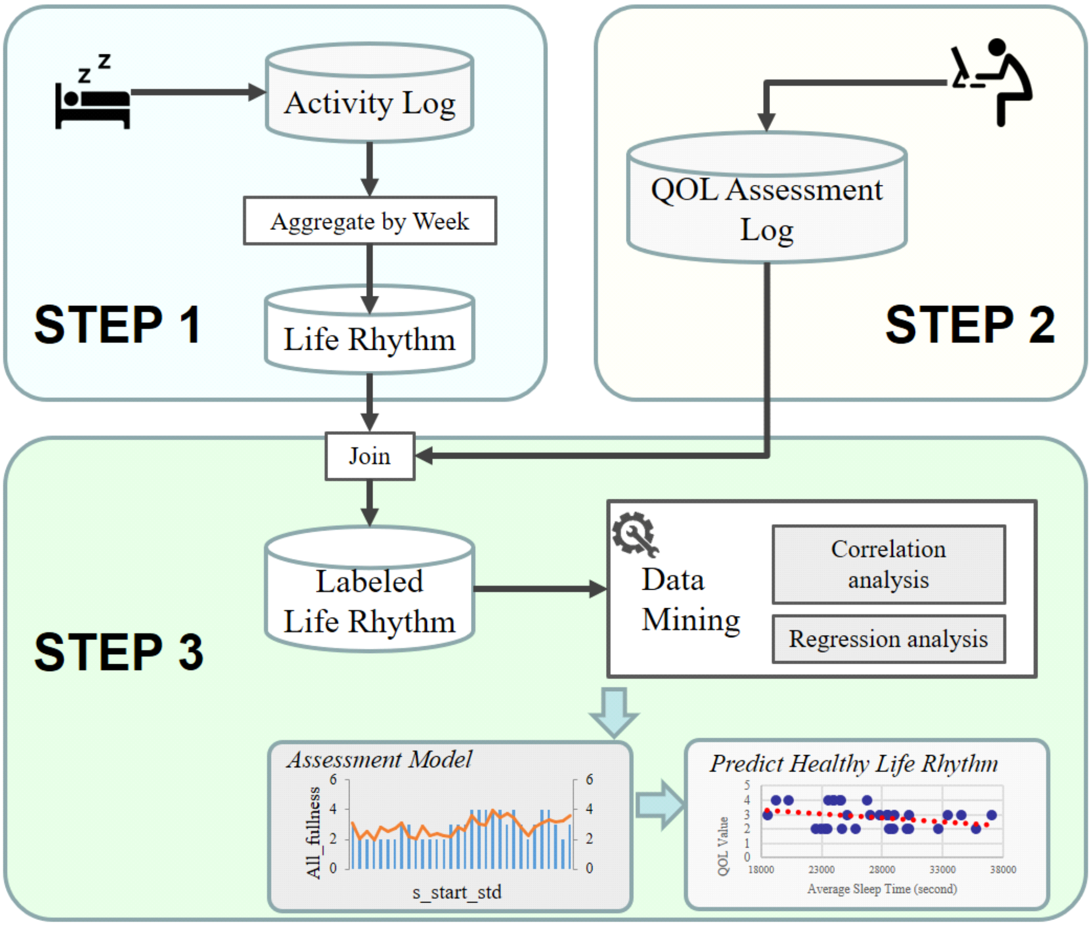


Fig. 4.1. Overview of Propose Method

4.4.2 Step1: Representing Life Rhythm

In this step, it is assumed that our previous system [61] continuously collects and records the daily activities of the user. The system is able to recognize seven types of activities including(sleeping, rising, watching, eating, bathing, going out, returning). For each activity recognized, the system records the activity in log data, as shown in Table 4.1. Each row represents a recognized daily activity, specified by username, date of occurrence, start time, end time, and the type of activity.

In A1 of Section 2.2.3, we saw that sleep and breakfast are relevant factors for the life rhythm. Hence, sleeping and eating activities are extracted from the log data, and the records are aggregated by *week*. The reason for the data aggregation by week is that this time period is a reasonable unit in which to observe the variation of life rhythm.

For the sleeping activities, the following statistics were calculated as relevant features of the life rhythm:

- **s_start_mean** (Average of start time of Sleep): Characterizes the time the user went to bed during the week.
- **s_start_std** (Standard deviation of start time of Sleep): Characterizes how regularly the user went to bed during the week.
- **s_end_mean** (Average of end time of Sleep): Characterizes the time the user got up during the week.
- **s_end_std** (Standard deviation of end time of Sleep): Characterizes how regularly the user got up during the week.
- **s_length_mean** (Average of length of Sleep): Characterizes the lenght of time the user slept every day during the week.
- **s_length_std** (Standard deviation of length of Sleep): Characterizes the regularity with which the user secured sleeping time.

For the eating activities, there was a focus on only the *first occurrence* of the day to capture breakfast activities. Then, the following statistics were calculated:

- **e_start_mean** (Average of start time of Eating Breakfast): Characterizes the time the user started eating breakfast during the week.
- **e_start_std** (Standard deviation of start time of Eating Breakfast): Characterizes how regularly the user started eating breakfast during the week. during the week.
- **e_end_mean** (Average of end time of Eating Breakfast): Characterizes the time the user finished eating breakfast during the week.
- **e_end_std** (Standard deviation of end time of Eating Breakfast): Characterizes how regularly the user finished eating breakfast during the week.
- **e_skip_count** (Number of days with breakfast skipping): Characterizes the number of days that the user skipped breakfast during the week.

Table 4.2 shows instances of [s_start_mean, s_start_std, e_start_mean] of an ac-

Table 4.2. A Part of the Features of Life Rhythm

WeekID (Period)	s_start_mean	s_start_std	e_start_mean
Week26 (2018/5/8-14)	27	5.318	7
Week27 (2018/5/17-23)	25.285	1.666	9.5
Week28 (2018/5/28-6/3)	25.714	2.312	8
Week29 (2018/6/5-11)	24.857	1.457	8.5

Table 4.3. Evaluation Scales of Fulfillment

Value	Description
1	not fulfilled at all
2	little fulfilled
3	fulfilled
4	very fulfilled
5	perfectly fulfilled

Table 4.4. A Part of the Results for QoL Assessment

WeekID (Period)	General	Research	PT–Job	Private
Week26 (2018/05/08-14)	2	2	4	4
Week27 (2018/05/17-23)	3	2	4	4
Week28 (2018/05/28-6/3)	4	4	3	3
Week29 (2018/06/05-11)	4	5	2	5

tual user calculated over four weeks, where each value is specified by an hour. For example, in Week27 (from May 17th to 23rd, 2018), the user went to bed at approximately 1:17 a.m. (=25.285) with a standard deviation of 1 h 40 min (=1.666). The user ate breakfast at approximately 9:30 am. Thus, it is possible to observe the life rhythm of the user from a certain perspective. It should be noted in Table 4.2 that the date between each week is not consecutive, since the user was out of home and data was missing for several days.

Table 4.5. Correlation Coefficient for All Features

Features of Life Rhythm	General	Research	PT - Job	Private
s_end_mean	-0.256741794	0.160209484	-0.270695775	-0.041061413
s_end_std	-0.046970989	-0.107007606	0.032590173	-0.033097385
s_start_mean	-0.102617434	-0.255107666	-0.014314429	-0.126324597
s_start_std	-0.40827074	-0.28103621	0.143160929	-0.220093272
s_length_mean	-0.311529873	0.089419633	-0.28006759	-0.187254745
s_length_std	-0.083191003	-0.058528694	-0.253880433	-0.062953223
e_end_mean	-0.422832323	-0.237803351	-0.420376469	-0.488726311
e_end_std	-0.038780787	-0.015032395	-0.190315633	-0.175506563
e_start_mean	-0.410194664	-0.242140335	-0.438639084	-0.460622496
e_start_std	-0.024196275	-0.06495214	-0.214491294	-0.147718547
e_skip_count	0.088386956	-0.173127965	-0.121837521	0.105884029

4.4.3 Step2: Measuring Quality of Life (QoL)

In this step, the proposed method requested that the user performed a self-assessment of QoL every week. The value of assessment requires that a numerical value is specified, which can be either discrete or continuous.

As mentioned in A2 of Section 2.2.3, the QoL is assessed by the fulfillment of daily living based on the user's perspective. In this preliminary study, the user was asked every week to answer the following four questions:

1. How was your last week in general?
2. How was your research work last week?
3. How was your part-time job last week?
4. How was your private time last week?

For each question, the user is instructed to indicate the degree of fulfillment based on a 5-level scale, as shown in Table 4.3.

Table 4.4 shows examples of the self-assessment of QoL. The columns General, Research, PT-Job and Private represent the answers to the aforementioned four questions (1)-(4), respectively. Each assessment was performed on the last day of each week. For instance, the user relayed that Week 27 (from May 17th to 23rd, 2018) was marginally fulfilled in general, little for research work, and very fulfilled for the part-time job and the private time.

4.4.4 Step3: Deriving Life-Rhythm Assessment Model

In this step, the proposed method combines the features of life rhythm and the value of the QoL Assessment, and then establishes an assessment model based on data mining. The data mining process is divided into two sub-steps.

Step3-1: Extraction of relevant features of life rhythm

In this sub-step, the features associated with life rhythms that are relevant to the QoL value of the user were identified. For this purpose, we applied correlation analysis to the combined data, and obtained a *correlation coefficient* for a pair of feature and QoL value.

Let X be a series of any feature of the life rhythm defined in Step 1 (See Section 4.4.2), and let Y be a series of any QoL value defined in Step 2 (See Section 4.4.3). Then, a correlation coefficient $\rho(X,Y)$ is defined by:

$$\rho(X,Y) = \frac{E[(X-E[X])(Y-E[Y])]}{(E[(X-E[X])^2]E[(Y-E[Y])^2])^{1/2}} \tag{4.1}$$

where E represents the mean operation.

When the absolute value of $\rho(X,Y)$ is large, this means that the feature X contributes to predicting the QoL Y. Therefore, using this correlation analysis, only features where $|\rho(X,Y)|$ is larger than a certain threshold τ. are considered.

The preceding correlation analysis is performed for pairs of feature X and the QoL values Y. However, the correlation analysis was also applied among the features. When two features X_1 and X_2 are highly correlated, choosing both features results in a decrease in the performance of the model. For instance, e_start_mean and e_end_mean are highly correlated, since the time of ending breakfast strongly depends on the time of starting breakfast. In such a case, either X_1 or X_2 is dropped from the feature selection, even if $|\rho(X_1,Y)|$ and $|\rho(X_2,Y)|$ are large.

Based on the aforementioned analysis, only relevant features of the life rhythm are selected, that have a relatively stronger correlation with the QoL value of the user.

Step 3-2: Derive assessment model

In this sub-step, a personalized model for the life rhythm assessment is established, which explains (predicts) the user's QoL value from given features of life rhythm. Let X_i $(i = 1,2,...,n)$ be a series of i-th relevant features, and let Y be a series of target QoL values. Then, the model is defined by a function f such that

$$Y = f(X_1, X_2, ..., X_n)$$

where f maps n-tuples of feature values $x_1, x_2, ..., x_n$ onto a QoL value y.

In this preliminary study, the linear regression model was adopted to derive f.

$$\begin{aligned} Y &= f(X_1, X_2, ..., X_n) \\ &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n \end{aligned} \tag{4.2}$$

Regression analysis is a set of statistical processes that estimate values of β_i's for a given n relevant features. In Section 4.5, I will explain how to build the assessment model using regression analysis tools.

4.5 Case Study

4.5.1 Experiment

A preliminary experiment was conducted to establish an assessment model of an actual user living in an OPH. The subject is a Ph.D. student in a university who lives alone in an apartment. He usually studies at the university, however, he also works part-time to make a living. Our previous system was already installed in his apartment.

In Step 1, his daily activity log was collected from May 1st, 2016 until June 31, 2018, and valid data for 32 weeks was retrieved from the period. In Step 2, the QoL self-assessment log data was collected for the same 32-weeks period. Given that the details of Step 1 and Step 2 are already outlined in Sections 4.4.2 and 4.4.3, respectively, I will review the details of Step 3.

Firstly, relevant features of the life rhythms were identified based on the correlation analysis. Table 4.5 shows the result of the analysis between the features and the four kinds of QoL values *1. As we focus *General* (the degree of QoL in

*1 The data used for the experiment is available at `http://ws.cs.kobe-u.ac.jp/~longniu/data/iiwas2018/`

Table 4.6. Detailed Results of Regression Analysis

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
Intercept	6.383265769	0.825	7.735	1.993E-08	4.693	8.073	4.693	8.073
s_start_std	-0.334647885	0.095	-3.524	0.001	-0.529	-0.14	-0.529	-0.14
s_length_mean	-8.1073E-05	2.6E-05	-3.095	0.004	-0.0001	-2.7E-05	-0.0001	-2.7E-05
e_end_mean	-0.062287619	0.032	-1.914	0.066	-0.129	0.0043	-0.128	0.004

Table 4.7. Regression Statistics

Multiple R	0.676199
R Square	0.457245
Adjusted R Square	0.399093
Standard Error	0.626037
Observation	32

general), it can be seen that the correlations coefficient of s_start_std (standard deviation of the time to go to bed), s_length_mean (average of sleeping time length) e_end_mean (average of the time of ending breakfast), e_start_mean (average of the time of beginning breakfast) are significant, which means that these features significantly contribute to the degree of *General*.

Because it was determined that e_end_mean and e_start_mean are strongly correlated with each other (i.e,. $\rho(e\_end\_mean, e\_start\_mean) = 0.983$), e_start_mean was droped from the feature selection. As a result, three features of s_start_std, s_length_mean, and e_end_mean were identified for the regression model of *General*.

Then, an assessment model was established using a regression analysis based on the three features, for predicting the value of QoL (*General*). Table 4.6 shows the detailed results of the regression analysis. The coefficients correspond to $\beta_i (i = 0, 1, 2, 3)$ for the equation (2). From the analyzed result, it is possible to derive a function of the assessment model of life rhythm:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \tag{4.3}$$

where

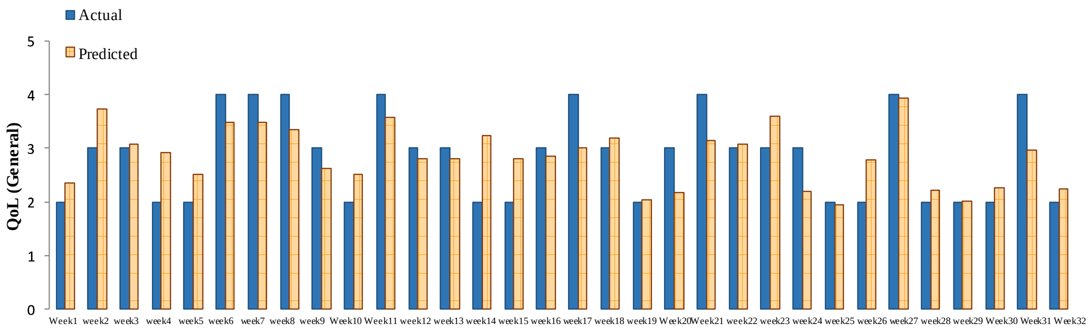


Fig. 4.2. Comparison of Actual and Predicted Values of QoL(General)

- Y: general fulfillment in daily living
- X_1: s_start_std, the standard deviation of the time to go to bed
- X_2: s_length_mean, the average of the sleeping time length
- X_3: e_end_mean, the average of the time to end breakfast
- β_0: 6.3832657 which is the constant called Y-Intercept
- β_1: -0.33464785 which is coefficient of s_start_std
- β_2: -0.000081073 which is coefficient of s_length_mean
- β_3: -0.06228761 which is coefficient of e_end_mean

Table 4.7 shows the overall statistics of the regression analysis of *General* with s_start_std, s_length_mean, e_end_mean. *R Square* is the *coefficient of determination*, which is a statistical measure of the closeness of the data to the fitted regression line. The result of *R Square* is approximately 0.457, which means the statistic measure is relatively close to the fitted regression line.

4.5.2 Interpreting Assessment Model

An attempt is made to interpret the derived assessment model. By assigning the obtained values in the model (4.3), the following formula is obtained:

$$QoL(General) = 6.38 - 0.33 \cdot s\_start\_std - 8.1 \times 10^{-5} \cdot s\_length\_mean - 0.062 \cdot e\_end\_mean \tag{4.4}$$

Firstly, the aim is to visualize how the model (4.4) is able to predict the actual values of QoL. Figure 4.2 depicts the actual QoL values (*General*) recorded during the self-assessment of the subject, and the predicted QoL values produced by the model (4.4). In the figure, the horizontal axis represents weeks, whereas the vertical axis represents the degree of general fulfillment. The solid blue line represents the actual value, and the shaded orange line shows the predicted value. Based on the results, it can be seen that the model is able to predict the actual values to a certain extent.

Next, the semantics of the model (4.4) are interpreted. The units of s_start_std, s_length_mean, and e_end_mean are hours, seconds, and hours, respectively. With respect to the life rhythm of the subject, the model (4.4) establishes the following facts:

- If the deviation of the time to go to bed increases by 1 hour, the QoL value of the subject decreases by 0.33 points.
- If the sleeping time increases by 1 hour (=3,600 seconds), the QoL value of the subject decreases by 0.29 points.
- If the time of breakfast is delayed for 1 hour, the QoL value decreases by 0.062 points.

Therefore, for this subject, maintaining a regular bedtime in addition to avoiding oversleeping, and late breakfasts are good habits for maintaining good QoL values.

4.5.3 Finding Personal Advice for Maintaining Healthy Life

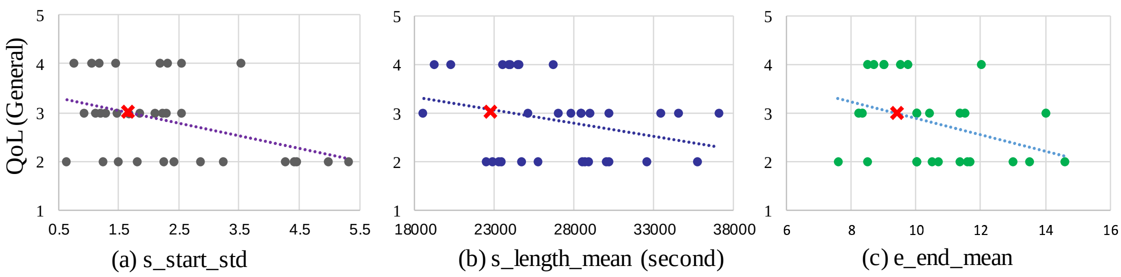


Fig. 4.3. Finding Correlations Between Each Feature and QoL

The derived model indicates that the statistical correlation between the relevant features of the life rhythm and QoL values. However, the user of the proposed method would like to know more details *personal Advice* on the daily activities. To identify such personal advice, regression analysis is applied to *each* of the relevant features.

More specifically, for each s_start_std, s_length_mean, and e_end_mean, a regression model is constructed for QoL(General). Figure 4.3 shows three scattered plots between QoL(General) and s_start_std, s_length_mean, or e_end_mean, respectively. In each sub-figure, the vertical axis represents the value of QoL(General), the horizontal axis represents the corresponding feature. The dotted line shows the regression line that fits the samples.

Suppose now that the subject wants personal advice for maintaining QoL values more than or equal to 3.0 (marginally fulfilled). Using Figure 4.3, three specific advice can be provided.

Firstly, from Figure 4.3(a), it is evident that the regression of QoL by s_start_std (depicted by a dotted line) is defined by:

$$QoL(General) = 3.44 - 0.26 \cdot s\_start\_std \tag{4.5}$$

From the line, it is known that the value of s_start_std should be smaller than 1.7 to achieve a value of QoL higher than 3.0. Based on this assessment, if the subject can control s_start_std to be less than 1.7, then the QoL will likely to be more than 3.0. Thus, we identify the personal advice that "the subject should keep the deviation of the time to go to bed within 1 hour 35 minutes, in order to be marginally or more fulfilled".

Secondly, from Figure 4.3(b), it is evident that the regression of QoL by s_length_mean (depicted by a dotted line) is defined by:

$$QoL(General) = 4.33 - 6 \times 10^{-5} \cdot s\_length\_mean \tag{4.6}$$

From the line, we know that the value of s_length_mean should be between 185,000 and 22,000 to achieve the value of the QoL higher than 3.0. Thus, the second personal advice is identified, i.e., "the subject should control the sleeping time to be between 5 hours 10 minutes and 6 hours 10 minutes, in order to be marginally or more fulfilled".

Finally, from Figure 4.3(c), it is evident that the regression of QoL by e_end_mean (depicted by a dotted line) is defined by:

$$QoL(General) = 4.6 - 0.17 \cdot e\_end\_mean \quad (4.7)$$

From the line, it is known that the value of s_length_mean should be less than 9.41 to achieve the value of QoL higher than 3.0. Thus, the third personal advice is identified, i.e., "the subject should control finish the breakfast before 9:25 every day, in order to be marginally or more fulfilled".

If the subject is able to adhere to the aforementioned three pieces of advice, it is highly probable that he will maintain fulfilled daily living.

4.6 Summary

In this chapter, a method that constructs a personalized assessment model of life rhythm was proposed. In the proposed method, the individual's life rhythms was characterized based on features related to sleeping and eating, which were extracted from the daily activity log. Data of the user's QoL was also collected based on the self-assessment survey. Using this data, the proposed method derives the assessment model by using regression analysis. A preliminary study was performed with an actual subject. Using the proposed method, it was possible to derive his personal assessment model based on bedtime, duration of sleeping, and the time to finish breakfast. Using the model, it was also possible to establish personal advice to maintain the subject's life rhythm as marginally or more fulfilled.

Chapter 5

Conclusion

In this chapter, the main conclusions of this dissertation are presented. In addition, a broad perspective on future work is considered.

5.1 Collection of Living Data of Individuals in OPHs

In this investigation, I presented a common data model for indoor location, called DM4InL, which prescribes a data schema to represent indoor location information without depending on any specific IPS or InL-App. By composing three data models (i.e., the location model, the building model and the object model), DM4InL represents location information of various objects inside a building in a standard format.

Next, I have proposed a Web-based integration framework called WIF4InL, in order to achieve data and operation integration for heterogeneous IPS. To achieve data integration, WIF4InL implements InL-Adapter, which provides different adaptation patterns for different system topology of IPS. InL-Query was also implemented, which provides fundamental API and composite API based on DM4InL. WIF4InL contributes to loose coupling of IPS and InL-App, which will significantly improve the efficiency and re-usability in InL-App development. The proposed framework was applied to integrate two existing IPS: RedPin and BluePin. In addition, I evaluated WIF4InL by investigating the sufficiency of the five capabilities for location-dependent queries.

With respect to the future work, we plan to conduct further evaluation of WIF4InL, with respect to performance and security for practical use cases. We are also interested in addressing more pragmatic issues that include *uncertainty* of location caused by unreliable devices, as well as *feature interactions* when integrating data and operations.

5.2 Recognition of Daily Activity

In this project, a system was proposed to improve the accuracy of daily activity recognition by the addition of a BluePin to an original system and integrating BLE-based location data with sensor data in the processing of feature engineering. To evaluate this system, I performed three experiments to compare the precision of activity recognition and averaged all daily activities.

Based on the results, It is evident that the Micro&Macro-average precisions and accuracy of some daily activity recognition improved significantly. However, for daily activities where overlap occur with other activities or the activity space crosses multi-zones, the accuracy recognition of the proposed system was not improved.

I then evaluated the activity recognition performance of 3 popular algorithms. Based on the results it was determined that the multi-classification decision forest is most suitable for the proposed system.

In addition, it was observed that there was a change of system performance for activity recognition with a change of the length of the training period. Based on the detailed comparisons, it was determined that the proposed system requires less time for training than the system only use Sensorbox.

In the future, the proposed system will be evaluated in several houses to determine how the learning process varies from one OPH to another. Moreover, we will validate whether the proposed seven types of daily activities are sufficient to capture the life rhythms in OPH.

5.3 Derivation of Personalized Assessment Model for Life Rhythm

In this project, a method was proposed that constructs a personalized assessment model for life rhythm. In the proposed approach, an individual life rhythm was characterized based on features related to sleeping and eating, which were extracted from daily activity logs. Data on the user's QoL were also collected using a self-assessment survey. Based on these data, the proposed method derives an assessment model by utilizing the regression analysis. A case study was conducted with an actual subject. The proposed method was able to derive his personal assessment model based on bedtime, the duration of sleeping, and the time to finish breakfast. Using the model, personal advise was also determined to maintain the subject life rhythm as marginally or more fulfilled.

In future works, the proposed method will be evaluated based on experiments with more subjects. It may also be interesting to investigate other algorithms apart from linear regression.

Acknowledgements

The research presented in this dissertation could not have been accomplished without the assistance of many other people.

Firstly, I am indebted to Professor Kuniaki Uehara, Professor Itsuo Hatono and Professor Mitsuo Yokokawa of Kobe University for their valuable guidance in my examination process. In addition, I also wish to thank Professor Zhiwei Luo of Kobe University for his insightful comments and advice in the interim report of my intensive course.

I am deeply grateful to Associate Professor Masahide Nakamura of Kobe University for providing me with the opportunity to study in his laboratory and for his excellent guidance during my research, which was instrumental in the completion of this thesis. I extend my sincerest thanks to him for his kindness and assistance in all aspects of my life. I would also like to express my graduate to Assistant Professor Sachio Saiki of Kobe University, who provided me with generous support on innumerable occasions. A special thank you goes out to Assistant Professor Shinsuke Matsumoto of Osaka University, who provided me with much-needed support during my semester of the doctoral program. I would also like to express appreciation to all members of the CS24 laboratory and the students who have graduated in the last 6 years. In particular, I am grateful to Sinan Chen and Haruhisa Maeda, for their cooperation as subjects in the final experiment in my research. It was a pleasant and fruitful experience to work with the laboratory, and I thank them for helping to create an enjoyable and friendly research environment.

I am so grateful to the Senshu International Student Scholarship Foundation for providing me with significant financial support during the 5 years of my intensive course. This funding allowed me to devote most of my attention and time to research without the distraction of part-time jobs. I am also grateful for the funding received through the Tateishi Science and Technology Foundation. I would also like to thank Editage (www.editage.jp) for English language editing.

A special thank you to my cousin, Hao Guo, who recommended me to my doctoral advisor. Thanks so much for your support while in Japan which assisted me in navigating through some difficult times during 2012 and 2013. In particular, I would like to thank my fiancée, Mingzi Sun. Thank you for supporting my decision to pursue my doctoral degree and the countless sacrifices you made that allowed me to get to this point. Last but not least, I would like to say a heartfelt thank you to my parents for always believing in me and encouraging me to study aboard. Without such support, I doubt that I would be at this point today.

Bibliography

[1] S.Asoka, K. Fukuda, and K. Yamazaki. Effects of sleep-wake pattern and residential status on psychological distress in university students. In *Sleep and Biological Rhythms*, volume 2, pages 192–198, 2004.

[2] E. Uchida, R. Kimoto, M. Tsukamoto, I. Kamibayashi, and S. Takeda. Influence of differences in living style in college students on sleep habits and eating habits. *MEMOIRS OF TAISHO UNIVERSITY Faculty of Buddhist Studies Faculty of Human Studies Faculty of LiteratureFaculty of Communication and Culture*, 100:331–340, mar 2015.

[3] E. Shiotani. *The Key of Health: Life Rhythm*. Kobe University, Kobe, Japan, 2009.

[4] Koji Maemura. Three elements (light, meal, melatonin) to adjust disorder of biorhythm. *Journal of Cardiology*, 43(2):154–158, 2011.

[5] Loren Fiore, Duc Fehr, Robot Bodor, Andrew Drenner, Guruprasad Somasundaram, and Nikolaos Papanikolopoulos. Multi-camera human activity monitoring. *Journal of Intelligent and Robotic Systems*, 52(1):5–43, 2008.

[6] Kazushige Ouchi and Miwako Doi. Smartphone-based monitoring system for activities of daily living for elderly people and their relatives etc. In *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, UbiComp'13 Adjunct, pages 103–106, New York, NY, USA, 2013. ACM.

[7] K. Kusano, H. Muro, T. Hayashi, F. Harada, and H. Shimakawa. Derivation of life rhythm from tracing elderly movement. In *The 10th Forum on Information Technology (FIT2011)*, volume 10, pages 891–892, sep 2011.

[8] E. Munguia-Tapia, S.S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Proceedings of PERVASIVE*,

pages 50–57, 2004.

[9] L. Pei, R. Guinness, R. Chen, J. Liu, H. Kuusniemi, Y. Chen, L. Chen, and J. Kaistinen. Human behavior cognition using smartphone sensors. In *Sensors*, volume 13, pages 1402–1424, 2013.

[10] Navizon Inc. Navizon indoor triangulation system, 2014. Retrieved September 8, 2014 from http://www.navizon.com/product-navizon-indoor-triangulation-system.

[11] NEC Corp. Smartlocator (in Japanese), 2014. Retrieved September 7, 2014 from http://jpn.nec.com/engsl/pro/smartlocator/index.html.

[12] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom'00, pages 32–43, New York, USA, 2000. ACM.

[13] Dinesh Manandhar, Seiya Kawaguchi, and Hideyuki Torimoto. Results of imes (indoor messaging system) implementation for seamless indoor navigation and social infrastructure platform. In *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation*, ION GNSS 2010, pages 1184–1191, September 2010.

[14] AR. Pratama, Widyawan, and R. Hidayat. Smartphone-based pedestrian dead reckoning as an indoor positioning system. In *System Engineering and Technology (ICSET), 2012 International Conference on*, pages 1–6, Sept 2012.

[15] Koozyt Inc. About placeengine, 2011. Retrieved September 7, 2014 from http://www.placeengine.com/showe/about.

[16] Josh Sookman. Guardly releases industry-first integrated indoor positioning system to provide responders with floor and room-level accuracy of mobile emergency callers, February 2013. Retrieved September 7, 2014 from http://blog.guardly.com/guardblog/2013/02/26.

[17] Kevin Curran, Eoghan Furey, Tom Lunney, Jose Santos, Derek Woods, and Aiden McCaughey. An evaluation of indoor location determination technologies. *J. Location Based Services*, 5(2):61–78, Jun 2011.

[18] Kiyohiko Hattori, Tetsuya Fujii, Youiti Kado, and Bing Zhang. Verification of the indoor user position and direction presumption system by using the two dimensional marker (in Japanese). *J. Information Processing Society of Japan. Ubiquitous Computing Systems (IPSJ. UBI)*, 2008(18):203–207, feb 2008.

[19] Hyun-Seung Kim, Deok-Rae Kim, Se-Hoon Yang, Yong-Hwan Son, and Sang-Kook Han. An indoor visible light communication positioning system using a RF carrier allocation technique. *J. Lightwave Technology*, 31(1):134–144, Jan 2013.

[20] S.L. Ting, S.K. Kwok, Albert H.C. Tsang, and George T.S. Ho. The study on using passive RFID tags for indoor positioning. *J. Engineering Business Management*, 3(1):9–15, 2011.

[21] Takahiro Ogasawara, Hiroshi Igaki, Akifumi Inoue, and Tohru Hoshi. Velsy: Development support system for indoor location platform (in Japanese). *J. Human Interface Society*, 15(2):131–150, 2013.

[22] NTT Data MSE Corp. Shoplat (in Japanese), 2014. Retrieved September 7, 2014 from https://www.nttdocomo.co.jp/service/convenience/shoplat.

[23] Shashi Shekhar and Hui Xiong. *Encyclopedia of GIS*. Springer Publishing Company, Incorporated, 1st edition, 2007.

[24] May Yuan. Wildfire conceptual modeling for building gis space-time models. In *Proceedings of GIS/LIS '94, Annual Conference and Expo Phoenix, Arizona*, GIS/LIS '94, pages 860–869, October 1994.

[25] Imad Afyouni, Cyril Ray, and Christophe Claramunt. Spatial models for context-aware indoor navigation systems: A survey. *J. Spatial Information Science*, 4(1):85–123, 2012.

[26] Kozo Watanabe. *Introduction to Data Modeling for Database Designing (in Japanese)*. Nippon Jitsugyo, Tokyo, 2003.

[27] Ulf Leonhardt and Jeff Magee. Multi-sensor location tracking. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom'98, pages 203–214, New York, USA, 1998. ACM.

[28] Y.J. Kim, H.Y. Kang, and J. Lee. Development of indoor spatial data model using CityGML ADE. *J. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-2/W2:41–45, 2013.

[29] Philipp Bolliger. Redpin - adaptive, zero-configuration indoor localization through user collaboration. In *Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, MELT '08, pages 55–60, September 2008.

[30] Jean-Christophe Zufferey, A. Klaptocz, A. Beyeler, J.-D. Nicoud, and D. Floreano. A 10-gram microflyer for vision-based indoor navigation. In *Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3267–3272, October 2006.

[31] Mohamed Kara Mohamed, Sourav Patra, and Alexander Lanzon. Designing simple indoor navigation system for uavs. In *Proceedings of 19th Mediterranean Conference on Control Automation (MED)*, pages 1223–1228, June 2011.

[32] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '99, pages 59–68, 1999.

[33] Stephen P. Tarzia, Peter A. Dinda, Robert P. Dick, and Gokhan Memik. Indoor localization without infrastructure using the acoustic background spectrum. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 155–168, June 2011.

[34] Nisarg Kothari, Balajee Kannan, Evan D. Glasgwow, and M. Bernardine Dias. Robust indoor localization on a commercial smart phone. *Procedia Computer Science*, 10:1114–1120, August 2012.

[35] Jo Agila Bitsch Link, Paul Smith, Nicolai Viol, and Klaus Wehrle. Footpath: Accurate map-based indoor navigation using smartphones. In *Proceedings of 2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, September 2011.

[36] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067–1080, November 2007.

[37] Yuki Kashio, Shinsuke Matsumoto, Sachio Saiki, and Masahide Nakamura. Design and implementation of service framework for presence sensing in home network system. In *The Third International Conference on Digital Information, Networking, and Wireless Communications*, DINWC2015, pages 109–114, February 2015.

[38] Serigio Ilarri, Eduardo Mena, and Arantza Illarramendi. Location-dependent query processing: Where we are and where we are heading. *ACM Comput. Surv.*, 42(3):12:1–12:73, March 2010.

[39] Ferenc Brachmann. A multi-platform software framework for the analysis of multiple sensor techniques in hybrid positioning systems. In *Proceedings of 10th Conference on Telematics Engineering*, JITEL 2011, September 2011.

[40] Kurt Gubi, Rainer Wasinger, Michael Fry, Judy Kay, Tsvi Kuflik, and Robert Kummerfeld. Towards a generic platform for indoor localisation using existing infrastructure and symbolic maps. In *Proceedings of 18th International Conference on User Modelling, Adaptation and Personalisation*, June 2010.

[41] INSITEO Inc. Platform of insiteo, 2014. Retrieved September 7, 2015 from http://www.insiteo.com/joomla/index.php/en/plateform.

[42] Seiji Sakakibara, Sachio Saiki, Masahide Nakamura, and Shinsuke Matsumoto. Indoor environment sensing service in smart city using autonomous sensor box. In *15th IEEE/ACIS International Conference on Computer and Information Science*, pages 885–890, June 2016. Okayama, Japan.

[43] Oliver Brdiczka, Patrick Reignier, and James L. Crowley. *Detecting Individual Activities from Video in a Smart Home*, pages 363–370. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[44] B.K. Hensel, G. Demiris, and K.L. Courtney. Defining obtrusiveness in home telehealth technologies: A conceptual framework. In *Journal of the American Medical Informatics Association*, volume 13, pages 428–431,

2006.

[45] Matthai Philipose, Kenneth P. Fishkin, Mike Perkowitz, Donald J. Patterson, Dieter Fox, Henry Kautz, and Dirk Hahnel. Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 3(4):50–57, October 2004.

[46] Christopher R. Wren and Emmanuel M. Tapia. *Toward Scalable Activity Recognition for Sensor Networks*, pages 168–185. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[47] Y. Fujino. A prospective cohort study of shift work and risk of ischemic heart disease in japanese male workers. *Journal of University of Occupational and Environmenatal Health*, 30(1):104, mar 2008.

[48] Pairwise independent combinatorial testing tool (pict). `http://balasegu.weebly.com/pict.html`, 2017.

[49] Long Niu, Sachio Saiki, and Masahide Nakamura. Recognizing adls of one person household based on non-intrusive environmental sensing. In *18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2017)*, pages 477–482. IEEE Computer Society and International Association for Computer and Information Science (ACIS), June 2017. Kanazawa, Japan.

[50] Vincent Van Asch. Macro-and micro-averaged evaluation measures [[basic draft]]. 2013.

[51] Long Niu, Sachio Saiki, Lydie du Bousquet, and Masahide Nakamura. Recognizing adls based on non-intrusive environmental sensing and ble beacons. In *Proceedings of Eighth International Conference on INDOOR POSITIONING AND INDOOR NAVIGATION (IPIN2017)*, September 2017.

[52] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1):69–90, Apr 1999.

[53] M. Bihis and S. Roychowdhury. A generalized flow for multi-class and binary classification tasks: An azure ml approach. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 1728–1737, Oct 2015.

[54] Z. Jianyong, L. Haiyong, C. Zili, and L. Zhaohui. Rssi based bluetooth

low energy indoor positioning. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 526–533, Oct 2014.

[55] Microsoft. Machine learning - initialize model - classification. `https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/machine-learning-initialize-model-classification`, 2018.

[56] K. Kusano, H. Muro, T. Hayashi, F. Harada, and H. Shimakawa. Derivation of life rhythm from tracing elderly movement. In *The 10th Forum on Information Technology (FIT2011)*, volume 10, pages 891–892, sep 2011.

[57] Taeko Yashima. *A Study on Life Rhythm of Elderly*. PhD thesis, Graduate School of Oriental University, 2014.

[58] Kumiko Ohashi. ”daily rhythms” in nursing: A concept analysis. *J-GLOBAL*, 14(2):1–9, 2010.

[59] Marie-Christine Mormont, James Waterhouse, Pascal Bleuzen, Sylvie Giacchetti, Alain Jami, André Bogdan, Joseph Lellouch, Jean-Louis Misset, Yvan Touitou, and Francis Lévi. Marked 24-h rest/activity rhythms are associated with better quality of life, better response, and longer survival in patients with metastatic colorectal cancer and good performance status. *Clinical Cancer Research*, 6(8):3038–3045, 2000.

[60] Masamichi Shimosaka. Analysis life rhythm based on human activity sensing data. *Journal of the Society of Instrument and Control Engineers*, 53(7):611–616, 2014.

[61] Long Niu, Sachio Saiki, and Masahide Nakamura. Integrating environmental sensing and BLE-based location for improving ADL recognition. In *The 19th International Conference on Information Integration and Web-based Applications & Services (iiWAS2017)*, December 2017. Salzburg, Austria.

A

List of Publications

A.1 Presentations in International Conferences

1. Long Niu, Sachio Saiki, and Masahide Nakamura, "A Preliminary Study for Quantitative Assessment of Life Rhythm Based on Sleeping and Eating Log Data," In 20th International Conference on Information Integration and Web-based Applications & Services (iiWAS2018), November 2018.
2. Long Niu, Sachio Saiki, and Masahide Nakamura, "Integrating Environmental Sensing and BLE-Based Location for Improving ADL Recognition,' ' In The 19th International Conference on Information Integration and Web-based Applications & Services (iiWAS2017), December 2017.
3. Long Niu, Sachio Saiki, Lydie du Bousquet, and Masahide Nakamura, "Recognizing ADLs Based on Non-Intrusive Environmental Sensing and BLE Beacons," In Proceedings of Eighth International Conference on Indoor Positioning and Indoor Navigation (IPIN2017), September 2017.
4. Long Niu, Sachio Saiki, and Masahide Nakamura, "Recognizing Adls of One Person Household Based on Non-Intrusive Environmental Sensing," In 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2017), no.CFP1779A-USB, pp.477-482, June 2017. (Kanazawa, Japan)
5. Long Niu, Sachio Saiki, and Masahide Nakamura, "Analyzing Indoor Environment Sensing Data for Recognizing Adls of One Person Household," In 2017 6th International Conference on Advanced Materials and Computer Science (ICAMCS 2017), vol.3, pp.323-329, May 2017. (Shenzhen, China)
6. Long Niu, Sachio Saiki, Shinsuke Matsumoto, and Masahide Nakamura,

"Implementation and Evaluation of Cloud-Based Integration Framework for Indoor Location," In The 17th International Conference on Information Integration and Web-based Applications & Services, pp.548-557, December 2015. (Brussels, Belgium)

7. Long Niu, Shinsuke Matsumoto, Sachio Saiki, and Masahide Nakamura, "Considering Common Data Model for Indoor Location-Aware Services," In 4th International Workshop on Location and the Web (LocWeb2014), pp.25-32, November 2014. (Shanghai, China)

A.2 Journal Papers

1. Long Niu, Sachio Saiki, Shinsuke Matsumoto, and Masahide Nakamura, "WIF4InL: Web-Based Integration Framework for Indoor Location," International Journal of Pervasive Computing and Communications, vol.12, no.1, p.49 - 65, May 2016.
2. Long Niu, Sachio Saiki, and Masahide Nakamura, " Using Non-Intrusive Environmental Sensing for ADLs Recognition in One-Person Household" International Journal of Software Innovation, vol.6, 2018.

A.3 Presentations in Domestic Conferences

1. 鈕 龍, 佐伯 幸郎, 中村 匡秀, "睡眠と食事のログデータに基づく生活リズムの定量的評価手法の検討," 電子情報通信学会技術研究報告, no.SC2018-22, pp.45-50, August 2018. (東京・法政大学)
2. 鈕 龍, 佐伯幸郎, 中村匡秀, "屋内環境センシングデータを用いた独居者の生活行動の検知," SC 研究会, vol.117, no.75, pp.41-46, June 2017. (University of Aizu(UBIC 3D))
3. Long Niu, Seiji Sakakibara, Seiki Tokunaga, Sachio Saiki, Takashi Matsubara, Masahide Nakamura, and Kuniaki Uehara, "Reasoning Daily Activities of Single Life Using Environment Sensing and Indoor Location," In 電子情報通信学会技術研究報告, pp.7-8, October 2016.
4. 鈕 龍, まつ本 真佑, 佐伯 幸郎, 中村 匡秀, "屋内ロケーションアウェアサー

ビスのための問い合わせ API の考察," 電子情報通信学会技術研究報告, vol.114, no.110,IN2014-28, pp.73-78, June 2014.

5. 鈕 龍, まつ本 真佑, 佐伯 幸郎, 中村 匡秀, "屋内ロケーションアウェアサービスに向けた位置表現データモデルの提案," 電子情報通信学会技術研究報告, vol.113, no.479, LOIS2013-71, pp.101-106, March 2014.

Doctor Thesis, Kobe University

“Achieving Healthy and Quality Life of One-person Households Using IoT and Machine Learning”, 127 pages

Submitted on January, 18, 2019

The date of publication is printed in cover of repository version published in Kobe University Repository Kernel.