February 2019
Master's Degree Thesis

Design of Geomagnetic Field Based Indoor Landmark Classifier using a Deep Learning

Graduate School of Chosun University

Department of Information and Communication Engineering

Bimal Bhattarai

Design of Geomagnetic Field Based Indoor Landmark Classifier using a Deep Learning

딥 러닝을 활용한 지자기장 기반 실내 랜드마크 분류기 설계

February 25, 2019

Graduate School of Chosun University

Department of Information and Communication Engineering

Bimal Bhattarai

Design of Geomagnetic Field Based Indoor Landmark Classifier using a Deep Learning

Advisor: Prof. Jae-Young Pyun

A thesis submitted in partial fulfillment of the requirements for a master's degree in engineering

October 2018

Graduate School of Chosun University

Department of Information and Communication Engineering

Bimal Bhattarai

This is to certify that the master's thesis of

Bimal Bhattarai

has been approved by examining committee for the thesis requirement

for the master's degree in engineering.

Committee Chairperson	**Prof. Goo-Rak Kwon**	_____
Committee Member	**Prof. Bumshik Lee**	_____
Committee Member	**Prof. Jae-Young Pyun**	_____

November 2018

Graduate School of Chosun University

Acknowledgement

First, I would like to express my sincere gratitude to my advisor Prof. Jae-Young Pyun for his invaluable support, encouragement, supervision, personal guidance, and useful suggestions throughout the course of my research work. I have been lucky to have a supervisor who cared about my work and who responded to my questions and queries so promptly.

Beside my advisor, I would like to thank the committee members, Prof. Goo-Rak Kwon and Prof. Bumshik Lee, for their encouragement and insightful comments.

I have great pleasure in acknowledging my fellow lab mates of Wireless and Mobile Computing Systems Lab, for the stimulating discussions and for all the fun we have had in the last two years.

Finally, I must express my profound gratitude to the biggest source of my strength, my family. This accomplishment would not been possible without their love, support, and continuous encouragement throughout my years of study. I will be grateful forever for your love.

Table of Contents

List of Figures

List of Tables

Acronyms

RF	Radio Frequency
BLE	Bluetooth Low Energy
DRNN	Deep Recurrent Neural Network
LSTM	Long Short-Term Memory
SVM	Support Vector Machine
KNN	K-Nearest Neighbors
GNSS	Global Navigation Satellite System
LOS	Line-Of-Sight
RSS	Received Signal Strength
CSI	Channel State Information
WAP	Wireless Access Points
MFS	Magnetic Field Strength
RNN	Recurrent Neural Network
IPS	Indoor Positioning System
RFID	Radio-Frequency Identification
IMU	Inertial Measurement Unit
FNN	Feed-Forward Neural Network
SLAM	Simultaneous Localization and Mapping
PDR	Pedestrian Dead Reckoning
NB	Naïve Bayes
AoA	Angle of Arrival

Abstract

Design of Geomagnetic Field Based Indoor Landmark Classifier using a Deep Learning

Bimal Bhattarai
Advisor: Prof. Jae-Young Pyun
Department of Information and
Communication Engineering
Graduate School of Chosun University

The unstable nature of RF signals and the need for external infrastructure inside buildings have limited the use of positioning techniques, such as Wi-Fi and Bluetooth fingerprinting. Compared to these techniques, the geomagnetic field exhibits a stable signal strength in the time domain. However, existing magnetic positioning methods cannot perform well in a wide space because the magnetic signal is not always discernible. In this thesis, deep recurrent neural networks (DRNNs) is proposed to build a model that is capable of capturing long-range dependencies in variable-length input sequences. The used of DRNNs is brought from the idea that the spatial/temporal sequence of magnetic field values around a given area will create a unique pattern over time; despite multiple locations having the same magnetic field value. Therefore, the indoor space can be divided into landmarks with magnetic field values and find the position of the user in a

particular area inside the building. A long short-term memory (LSTM) DRNNs is presented for spatial/temporal sequence learning of magnetic patterns and evaluate their positioning performance on our testbed datasets. Experimental results show that our proposed models outperform other traditional positioning approaches with machine learning methods, such as k-nearest neighbors (KNNs) and support vector machine (SVM).

한글요약

딥 러닝을 활용한 지자기장 기반 실내 랜드마크 분류기 설계

비말 바하타라이
지도교수 :변재영
조선대학교대학원,
정보통신공학과

GPS 신호의 수신이 제한되는 실내 환경에서 위치 기반 서비스 제공을 위해 Wi-Fi 및 블루투스와 같은 무선 기술 기반의 핑거프린팅 방식이 사용되고 있지만, 이러한 기술들은 실내 환경에 인프라를 구축해야 한다는 점과 실내 환경에 따라 무선 신호의 세기가 불안정하게 관측된다는 한계가 있다. 이러한 무선 기술과 달리, 지자기장은 시간 도메인 상에서 비교적 안정적인 자기장 세기를 보여주고 있어 지자기장 기반 측위 기술에 대한 연구 및 활용 사례가 증가하고 있다. 그러나, 자기장 세기가 언제 어디서나 식별 가능한 고유한 값을 나타내지는 않기 때문에 기존의 지자기장 기반 측위 방식은 넓은

공간에서 활용하기에 무리가 있다. 본 논문에서는 가변 길이 입력 시퀀스에서 긴 범위의 종속성을 획득할 수 있는 모델을 구축하기 위한 Deep Recurrent Neural Networks (DRNN)을 제안한다. DRNN 의 사용은 동일한 세기의 자기장 값이 여러 장소에서 관측될 수 있지만 주어진 영역 주위의 자기장 값의 공간적/시간적 시퀀스는 시간의 흐름에 따라 고유한 패턴을 생성한다는 점에 착안하여 제안되었다. 따라서, 실내 공간은 자기장 값을 가진 랜드마크로 나뉘어질 수 있으며 이를 통해 건물 내 특정 영역에서 사용자의 위치를 찾을 수 있다. 본 논문에서는 자기장 패턴의 공간적/시간적 시퀀스 학습을 위해 장단기 메모리 (LSTM) DRNN 을 제시하고 본 연구에서 사용된 테스트 환경에 대한 데이터세트에서 측위 성능을 평가하였다. 마지막으로, 본 논문의 실험 결과를 통해 제안 모델이 Support Vector Machine (SVM) 및 K-nearest Neighbors (KNN)과 같은 기존의 머신러닝 기반 측위 방식들보다 우수한 성능을 제공함을 확인하였다.

Chapter 1: Introduction

1.1 Motivation

The demand for indoor location-based service (LBS) is fueling the decade-long research into indoor positioning technology. In an outdoor environment, the global navigation satellite system (GNSS) uses line-of-sight (LOS) transmission to position the user [1]. However, it cannot be applied in an indoor environment due to multipath effect, signal fading, shadowing, and delay distortion in a radio propagation environment [2]. With the proliferation of smartphones and other mobile devices, an array of embedded sensors can be used for the indoor positioning system. Many studies have been performed on Wi-Fi or Bluetooth based fingerprinting indoor localization using received signal strength (RSS) and channel state information (CSI) [3][4][5][6][7][8][9]. Although these methods can achieve the desired accuracy at an acceptable cost, they cannot work effectively when a radio frequency (RF) signal is weak. Also, these methods need an expensive external device such as wireless access points (WAP) or Bluetooth beacon all over the building to transmit the RF signal.

In contrast, the geomagnetic field is ubiquitous and does not need any additional infrastructure. The magnetic field strength (MFS) is non-uniform inside a building due to building materials such as steels, iron, and reinforced concrete [10][11]. Due to these anomalies in the MFS, it can be used by an indoor positioning system. Magnetic signatures have previously been used for robot tracking and navigation [12][13]. Specifically, the fingerprint-based approach is widely accepted for magnetic signature recognition due to low

complexity and a real-time testing process [14][15]. This fingerprint method is usually divided into two phases: the training and the testing. In the training phase, the dataset is prepared by collecting the MFS data at all reference points and stored in the positioning server. Then, the proper deep learning model is used to train those data. In the testing phase, the real-time MFS data are collected and given to the trained model to find out the current location. The performance can be evaluated by performance metrics like accuracy, precision, recall, and F1 score.

Although many other magnetic-based positioning methods have been proposed, the ambiguity of magnetic data in a wide space may converge a positioning result in the wrong way at some cases. Here, a deep learning model, which can perform effectively to classify landmarks based on the magnetic signal variations is adopted. In the area of deep learning, there is an increasing interest in the recurrent neural network (RNN), which has been used for many sequence modeling tasks. RNN has been used in many technical applications, such as language modeling, speech recognition, video processing, and many other sequence labeling tasks [16][17][18][19]. The reason behind its promising performance is its ability to learn the temporal dependencies by exploiting the contextual information and in variable-length data.

1.2 Objectives

This research introduces an indoor classification system based on a deep learning models and compare its effectiveness with other state-of-art machine learning methods. We also aim to suggest and evaluate other deep learning

techniques that can increase the overall accuracy and performance of indoor positioning system using landmarks classification.

1.3 Contributions

In this thesis, we utilize the unique signal features of the magnetic field gathered from a smartphone for fingerprinting based classification. We use long short-term memory (LSTM)-based deep RNNs (DRNNs) for classifying the location mapped from variable-length input sequences of MFS, and we develop a positioning estimation architecture based on deep layers of unidirectional and bidirectional RNNs, as well as a cascaded architecture [20]. Moreover, we test these deep learning models with different testbeds to validate their performance at classifying various landmarks. The major contributions of our research work are as follows:

1. We experimentally validate the feasibility of using MFS for landmark classification. In addition, we show that the MFS data is stable over a period of time.
2. We show the success of using unidirectional and bidirectional DRNNs for landmark classification without any additional data preprocessing and validate its performance in two typical indoor environments.
3. We introduce the implementation of bidirectional DRNNs for magnetic landmark classification. As far as we know, this is the first work to do so.

1.4 Thesis Layout

The structure of this thesis is organized as follows. After the Introduction in Chapter 1, we provide a background overview of the Indoor positioning

system (IPS), Magnetic field preliminaries, RNN, LSTM, and performance metrics commonly used for performance evaluation in Chapter 2. Chapter 3 discusses relevant previous works in the field of IPS. The architectural structure of our proposed system and the three DRNN models are described in Chapter 4. Chapter 5 provides an overview of the experimental methodology and setup, which is used to train and evaluate our models. Experimental results are discussed and analyzed in Chapter 6. Finally, chapter 7 concludes the thesis and explore suggestions for further research based on the presented work.

Chapter 2: Background

2.1 Indoor Positioning System (IPS)

In an indoor environment, due to signal attenuation and multipath fading, GNSS cannot be used to track user location. An IPS is a system to locate a user or an object inside a building. An IPS is implemented based on environment and technologies available. Several non-radio and wireless technologies have been studied in the last decades. Most of the studies use Wi-Fi or Bluetooth low energy (BLE) signals, taking advantages of existing infrastructures deployed inside the building. However, non-radio technology such as geomagnetic field does not require any additional infrastructure.

2.1.1 Radio technologies

Most of the research in positioning system has been focused on using radio frequency (RF) signals in an indoor environment. The technologies generally used are Wi-Fi and Bluetooth, which utilizes Received signal strength (RSS) to determine the user's location inside a building. Since the inverse-square law applies to radio wave propagation, the determination of distance is possible based on RSS.

Radio-Frequency Identification (RFID) tags are also used for indoor positioning systems [21]. The location is estimated using fixed RFID readers placed in the building. The RFID tag or a smartphone attached to user is scanned by RFID readers to give the positon of a user inside a building. However, the limitation is that the user should be close enough to the RFID reader for scanning the tag.

The operation principle for wireless indoor positioning technologies using Wi-Fi or Bluetooth is similar. The RF signal is received using a different RF receiver in a smartphone. The signal received provides an information on Angle of Arrival (AoA), Time of Arrival (ToA) or RSS. The RSS and ToA can be used for distance estimation, which can be deployed in different location estimating algorithms. The most commonly used technique is the fingerprinting technique. It requires the collection of RSS at many reference points to build a "radio map". The radio map is stored in a database and it is used to position the user by matching the RSS in real time. The fingerprinting process can be divided into an offline phase and an online phase as shown in Figure 1.

2.1.2 Non-radio technologies

The technologies using infrared signals, geomagnetic field, acoustic/sound signals, inertial measurement, vision-based data etc. do not require RF signal to position a user. Moreover, the technologies such as the geomagnetic field and inertial measurement do not even need any expensive external infrastructure inside a building for positioning task. The infrared signals are the active badge system introduced by R. Want et al. in 1992 [22]. In this system, the receivers are placed inside the specific places in the building.

The user with a wearable badge, which emits an infrared signal, can know its position using receivers. However, the system is not efficient because infrared signals need to be in line-of-sight and have a short-range transmission. Similarly, acoustic signal based positioning is based on sound signal transmission by a beacon.

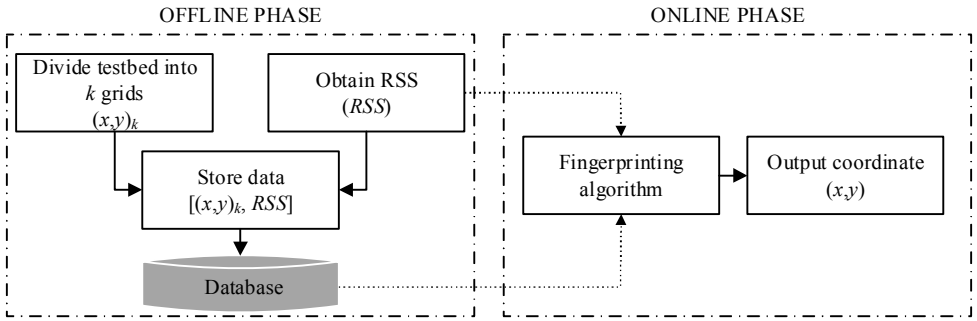


Figure 1. Fingerprinting based wireless indoor positioning system using RSS.

The inertial measurement unit (IMU) is used to track the object or person and it is mostly applied in pedestrian dead reckoning system [23][24][25]. The sensor data from the modern smartphones can be given to proper algorithms to track the movement of a person. The user can be localized based on step detection, step length and heading angle in an indoor environment using the IMU unit of the smartphone, assuming that the starting position of the user is already known.

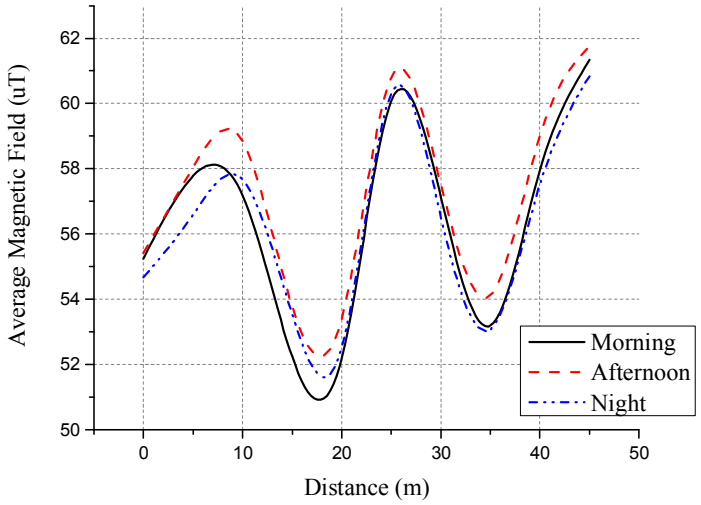
In recent years, many researches have been done in geomagnetic filed based positioning system [26][27]. The dedicated chips inside smartphone called as magnetometer can sense the magnetic field variations in the Earth's magnetic field. These variations can be used to build a radio map and provide localization [28]. Magnetic based positioning system is further discussed in other chapters.

2.2 Magnetic Field Preliminaries

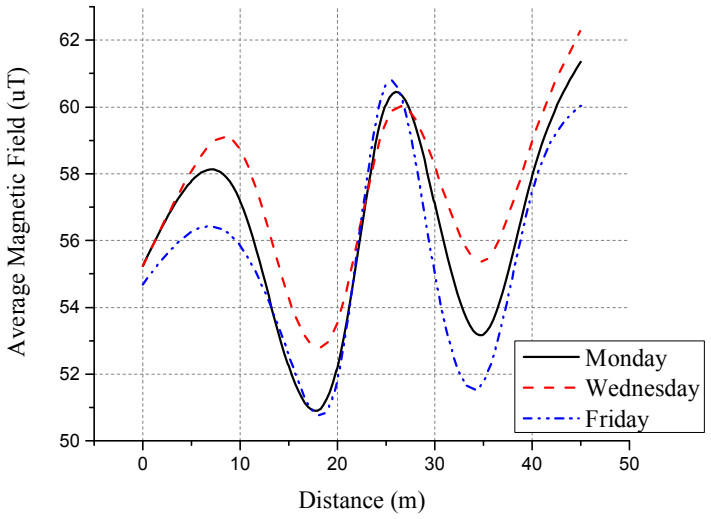
The geomagnetic field is present on the surface of the earth with a magnitude from 0.22 to 0.65 Gauss (25 to 65 μT). By using a magnetometer, a smartphone can measure the magnetic field in the form of a vector with three

components (m_x, m_y, m_z). The geomagnetic field is found to be stable in the absence of any interference from other external magnetic elements. We conducted an experiment to find the stability of the magnetic field in a corridor located on the eighth floor of an IT department building, Chosun university. Magnetic field data were collected along the corridor of length $100\,m$ at different times for a day, a week, and a month as shown in Figure 2. Later, the data were analyzed to see a statistical significance of magnetic field over a period of time.

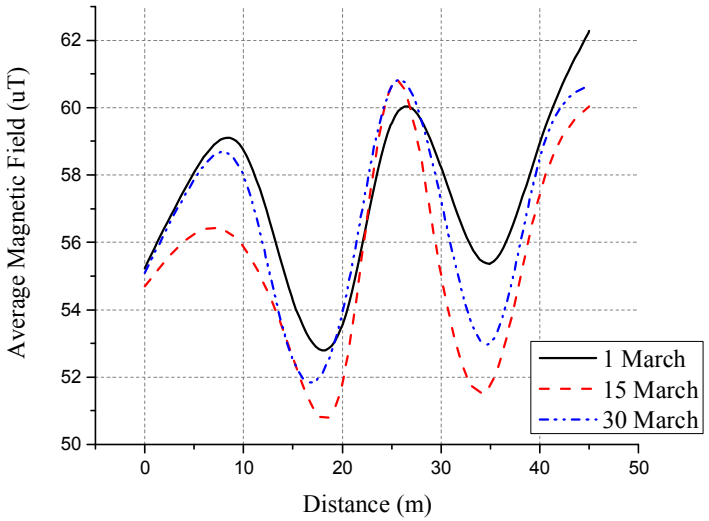
There were no statistically significant differences between magnetic field data from different time of day $(F = 0.29, P = 0.74)$, week $(F = 0.90, P = 0.41)$, and month $(F = 0.96, P = 0.39)$ as determined by a one-way analysis of variance (ANOVA) [29]. Here, F is the ratio of variation between MFS sample means to variation within the samples and P is the probability to determine how common or rare an F-value is on the presumption that the null hypothesis is true. The null hypothesis is usually rejected if $P < 0.05$. Therefore, we used the magnetic field, which is stable over a period of time for the indoor landmark classification.



(a)



(b)



(c)

Figure 2. Stability of magnetic field value over time: (a) variation in a day; (b) variation over a week; (c) variation over a month.

2.2.1 Magnetic Data Acquisition

The magnetic data for a landmark or reference location is obtained from the magnetic sensor of a smartphone. The data structure can be formed as $D = [m_x, m_y, m_z, M_A, m_{xrot}, m_{yrot}, m_{zrot}]$, where $m_x, m_y,$ and m_z represent magnetic field intensity from the three-axis magnetic sensor of a smartphone in space relative to the orientation of the phone, M_A represents average magnetic field intensity, and $m_{xrot}, m_{yrot},$ and m_{zrot} represent the magnetic field intensity after it has been converted to global frame system. Since the orientation of the smartphone plays an important role in positioning, we had also gathered magnetic field data with a quaternion-derived rotation matrix. Thus, we have to obtain the 3×3 rotation matrix R, which is the change from the device coordinate system to global coordinate system. Given any vector m

in the device coordinate system, the corresponding vector m_{rot} in global coordinate system can be obtained by multiplying m with R.

$$M_A = \sqrt{m_x^2 + m_y^2 + m_z^2} \tag{1}$$

$$[m_{xrot}, m_{yrot}, m_{zrot}] = R \times [m_x, m_y, m_z]^T. \tag{2}$$

The first feature m_{xrot} in (2) is a very small value close to zero, thus m_{yrot} and m_{zrot} usually retain the variation of magnetic field at different location points.

2.3 Recurrent Neural Network (RNN)

A recurrent neural network (RNN) is a class of deep neural network that contains cyclic connections that allow it to learn the temporal changes of a sequential data. Unlike traditional feed-forward neural networks (FNNs), a RNN has the characteristics of memorizing the previous information and applying it to the current input. RNNs have been successfully applied to sequential nature datasets such as natural language processing, due to their capability to model highly non-linear features. As shown in Figure 3, each RNN node generates the output y_t and current hidden state h_t by using the current input x_t and previous hidden state h_{t-1} based on the following equations:

$$h_t = \sigma_H(W_{HH} h_{t-1} + W_{IH} x_t + b_h) \tag{3}$$

$$y_t = \sigma_O(W_{HO} h_t + b_y), \tag{4}$$

where σ_H and σ_O are the hidden layer and output layer activation functions, respectively. W_{HH}, W_{HO}, and W_{IH} are the weights for the hidden-to-hidden recurrent connection, hidden-to-output recurrent connection, and input-to-hidden connection, respectively. b_y and b_h are the bias terms for the output

and hidden states, respectively. The activation functions are element-wise and non-linear and are commonly selected from various existing functions such as sigmoid, hyperbolic tangent, or rectified linear unit (ReLU).

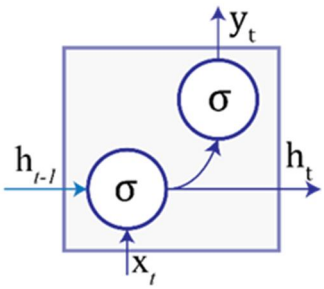


Figure 3. Schematic diagram of an RNN node.

2.4 Long Short-Term Memory (LSTM) based RNN

The traditional RNN is unable to handle long sequences of data. In addition, training RNN can be challenging due to vanishing and exploding gradient, which creates a problem when backpropagating through long-range temporal intervals [29]. In order to handle the long-range dependencies of learning data, a new class of network architecture with learnable gates has been used which is known as LSTM. LSTM contains memory blocks with memory cells called gates in the recurrent hidden layer as shown in Figure 4.

These learnable gates modulate the flow of information and control when to forget previous hidden states. Also, the new information is updates by the gates. The function of each memory cell is as follows:

- Input gate i_t controls input activation into the memory cell
- Output gate o_t controls memory cell outflow of activation to output

- Forget gate f_t determines when to forget memory cell content in the internal state
- Input modulation gate g_t provides the input to the memory cell
- Internal state I_t controls cell internal recurrence
- Hidden state h_t controls information from previous samples within the context window

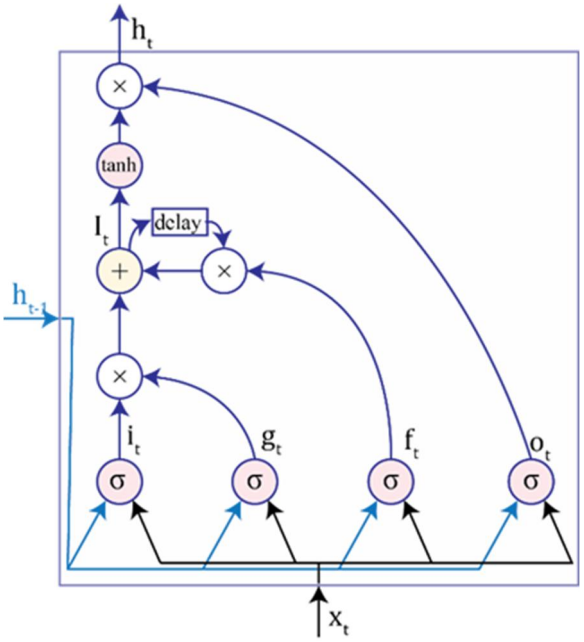


Figure 4. Schematic diagram of LSTM memory cell structure with an inner recurrence c_t and outer recurrence h_t.

$$i_t = \sigma(b_i + U_i x_t + W_i h_{t-1}) \tag{5}$$

$$f_t = \sigma(b_f + U_f x_t + W_f h_{t-1}) \tag{6}$$

$$o_t = \sigma(b_o + U_o x_t + W_o h_{t-1}) \tag{7}$$

$$g_t = \sigma(b_g + U_g x_t + W_g h_{t-1}) \tag{8}$$

$$I_t = f_t I_{t-1} + g_t i_t \tag{9}$$

$$h_t = tanh(I_t) o_t, \tag{10}$$

where σ is the activation function, the U and W terms represent weight matrices (e.g., U_i is the weight matrix for the input data x_t given to input gate i_t and W_i is the weight matrix for h_{t-1} data given to input gate i_t), and the b term denotes the bias vector (e.g., b_i is the input gate bias vector). The training process of LSTM-RNNs is mostly centered on learning when to let an activation into the internal states of its cell and when to let an activation of the outputs. In addition, the network needs to learn the parameters $b, U,$ and W of the memory cell, as shown in Equations (5) – (10).

2.5 Performance Metrics

The performance of the proposed model is verified using the following evaluation metrics [30]:

a) Precision: Measuring the number of true landmarks out of those classified as positive. The overall precision is calculated by averaging the precision of each individual class:

$$Per - class\ Precision_c = \frac{t_{p_c}}{t_{p_c} + f_{p_c}} \tag{11}$$

$$Overall\ Precision = \frac{1}{k}\left(\sum_{c=1}^{k} \frac{t_{p_c}}{t_{p_c} + f_{p_c}}\right), \tag{12}$$

where t_{p_c} is the true positive rate of landmark c, f_{p_c} is the false positive rate, and k is the number of landmarks in the dataset.

b) Recall (Sensitivity): Measuring the number of landmarks that are correctly classified out of the total samples in a class. The overall recall can be obtained by averaging of the recalls for each individual class:

$$Per - class\ Recall_c = \frac{t_{p_c}}{t_{p_c} + f_{n_c}} \tag{13}$$

$$Overall\ Recall = \frac{1}{k}\left(\sum_{c=1}^{k} \frac{t_{p_c}}{t_{p_c} + f_{n_c}}\right), \tag{14}$$

where f_{n_c} is the false negative rate of a class c.

c) Accuracy: Measuring the proportion of correctly predicted classes over all predictions:

$$Overall\ Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \tag{15}$$

where $TP = \sum_{c=1}^{k} t_{p_c}$ is the overall true positive for a classifier on all classes, $TN = \sum_{c=1}^{k} t_{n_c}$ is the overall true negative rate, $FP = \sum_{c=1}^{k} f_{p_c}$ is the overall false positive rate, and $FN = \sum_{c=1}^{k} f_{n_c}$ is the overall false negative rate.

d) F1-score: The weighted mean of precision and recall:

$$F1\ Score = \sum_{c=1}^{k}\left(2\left(\frac{n_c}{N}\right) * \frac{precision_c * recall_c}{precision_c + recall_c}\right), \tag{16}$$

where n_c is the number of samples in a class c and $N = \sum_{c=1}^{k} n_c$ is the total number of individual examples in a set of k classes. The F1 Score provides a measure of a test accuracy.

Chapter 3: Related Works

3.1 Traditional Approaches

Many approaches have been used to obtain the desired accuracy in indoor positioning. Most studies use Wi-Fi signal strength or radio frequency identification (RFID) to measure the user position [19][31]. Recent literature has reported that the MFS can be used instead of RF signal. Some studies have shown a navigation system for robots using a magnetic field. [32] and [33] showed simultaneous localization and mapping (SLAM) for geomagnetic field based robot positioning. They used a particle filter, which utilizes odometers to achieve a maximum positioning error of 10 cm. The odometer gave accurate distance and rotation information, and thus re-sampling particles based on rotation and moving distance was precise.

Haverinen et al. in [26] implemented the same SLAM with a person by switching the odometers with pedestrian dead-reckoning (PDR). However, due to lack of proper odometric information, the performance was not as good as when using of odometers.

In another work, Navarro and Benet in [34] used a two-dimensional magnetic map to determine the local heading of the robot. They considered the magnetic field as a continuous function and used bilinear interpolation to determine the MFS at un-sampled points. In [35] mobile phones were used to measure MFS and used it as magnetic signatures for identifying rooms. Since this system depends heavily on pillars in the building, it only achieved room-level accuracy.

Gozick et al. in [26][27] attempted magnetic landmark localization with the MFS created from pillars of a building. They measured the sequence of magnetic field values and matched these values to pre-recorded landmark's MFS. However, in their research, the magnetic landmarks are defined with prior knowledge that columns are ferromagnetic objects.

3.2 Machine Learning Approaches

3.2.1 K-nearest Neighbor (KNN)

The KNN is a non-parametric algorithm, which assumes that the similar inputs give similar outputs. It has a minimal training phase and is used for both classification and regression [36]. The user location at every instance is predicted using test data by matching the features to find the closest locations in the radio map. To find the k closest points to the target, KNN match the signal vector measured by the target device against n location fingerprints saved in the database. We use Euclidean distance $D\ (a, b)$ between the target and the pre-stored location points in the database as shown in Equation 17.

$$D\ (a, b) = \sqrt{\sum_{i=1}^{n} (b_i\ - a_i)^2}, \tag{17}$$

where b_i and a_i are the pre-stored signal value and obtained signal value in the i^{th} location point. The final location is determined by averaging the x and y coordinates of the k closest fingerprints obtained.

3.2.2 Support Vector Machines (SVM)

The SVM, introduced in 1995 by Cortes and Vapnik [37] is a machine learning algorithm used for classification. It is used in many different tasks including pattern recognition and face detection [38][39]. An SVM generally works by finding a separating hyperplane between data of two different classes. The algorithm searches the optimal hyperplane, which maximizes the minimum distance from the plane to the nearest data point of each class. The hyperplane is called optimal when the distance from the nearest data point, known as the margin, to the hyperplane is maximized.

3.2.3 Decision Trees

The algorithm separates the data set based on criterion resulting in a tree-like structure [40]. It breaks down a data set into smaller subsets, also increasing the associated decision tree. The decision tree outputs decision nodes and leaf nodes. A decision node has more than two branches while a leaf node represents a classification. The criterion mostly used is information gain, which means that at each split, there is maximization of the decrease in entropy due to the split. The ratio of n class elements over all elements of the leaf node containing data item m is given as $P\ (^{n}/_{m})$.

3.2.4 Naïve Bayes (NB)

The NB classifier is based on the Bayes Theorem [41] that examines the maximum likelihood of features present in the predicted class. It assumes that the presence of particular feature in a class is not dependent to any other features. The Bayes's rule of conditional probability is given as:

$$P(K_i/A) = \frac{P(K_i).P(A/K_i)}{\sum_{j=1}^{n} P(K_j).P(A/K_j)}, \quad (18)$$

where K_i denotes the classes, A denotes a set of conditions on the features and n denotes the total number of classes. In above equation, the denominator can be regarded as a constant. Since the features are considered to be independent with each other, the probability that an observation is of class K_i, given feature values A_j, is

$$P(K_i/A) = \frac{1}{Z}.P(K_i) \prod_{j=1}^{n} P(A_j/K_i). \quad (19)$$

With the help of the log-likelihood ratio, we can determine the class, where the observed features belong.

3.2.5 Logistic Regression

Logistic regression (LR) analyzes the relation between multiple independent variables and the dependent variables. The output of LR is the probability that the given input feature belongs to a certain class. The parameters to estimate the probability of classes can be found using different optimization methods. The optimization method is selected based on the data set's size and a number of input features. Most commonly used methods are coordinate descent and stochastic average gradient (SAG).

The LR models are probability based so it does not have requirements on the distribution of the input feature variables [42][43]. If P is the probability that a binary output variable $Y = 1$ when input feature variable $X = x$ then the logistic response function can be shown as:

$$P(Y = 1/X = x) = \frac{e^{\beta_1 + \beta_0 x}}{1 + e^{\beta_1 + \beta_0 x}}, \tag{20}$$

where β is the coefficient of the input feature x used in the above equation. For multiple input features i.e., $X = (x_1, \ldots, x_n)$, where n is total number of input features, the logistic regression function can be written as:

$$P(Y = 1/X = x_1, \ldots, X_n = x_n) = \frac{e^{\beta_0 + \beta_1 x + \cdots + \beta_n x_n}}{1 + e^{\beta_0 + \beta_1 x + \cdots + \beta_n x_n}}. \tag{21}$$

This equation calculates the probability of the output variable to be one, given many input features.

Chapter 4: Proposed System

The architecture of the proposed DRNN based landmark classification system is shown in Figure 5. It does not require any additional infrastructure except a smartphone device with a magnetic sensor to classify the landmark locations. The system consists of two steps: an offline training phase and an online testing phase. First, the raw magnetic data are collected at various reference locations known as landmarks. Then, in the preprocessing procedure, the data are divided into different segments according to the length of our DRNN input.

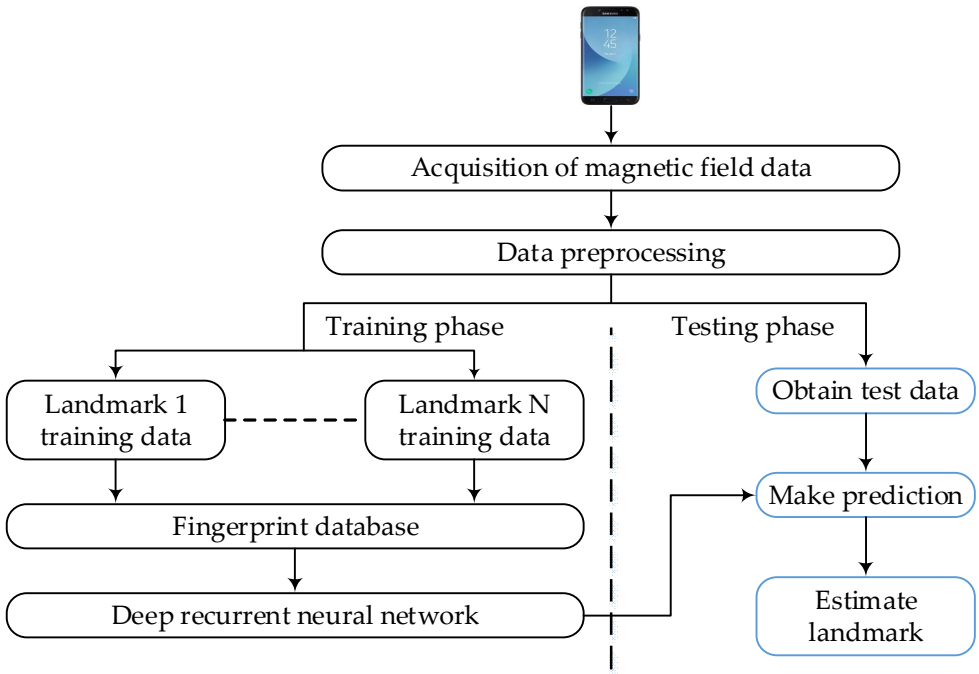


Figure 5. Architecture of the proposed LSTM-DRNN positioning system representing training and testing phase.

Finally, the preprocessed data from the landmarks are combined to generate a fingerprint database corresponding to each location, which consists of training and testing sets.

The training set is used to train our proposed LSTM-DRNN, whereas the testing set is used to validate the model. In the testing phase, the trained model uses the test data to classify the estimated landmark. The accuracy for landmark's classification can be dependent on the training model of the proposed LSTM-DRNN.

4.1 DRNN Architecture

Figure 6 presents a schematic diagram of the proposed DRNN indoor positioning system. It performs the direct mapping from magnetometer inputs to different landmarks. A specific time window is used to classify the landmark position. The input contains a discrete sequence of equally spaced samples $(x_1, x_2, \dots, x_T)$, where each data x_t is a vector of individual data examples D observed by the magnetic sensor at time t. These samples are passed to an LSTM-based DRNN model after being segmented into windows of maximum time index T. For the output, we get a sequence of scores denoting the landmark label prediction for each time step $(y_1^L, y_2^L, \dots, y_T^L)$, where $y_t^L \in R^k$ is a vector of prediction scores for given input sample x_t, L is the number of DRNN layers or top layer and k is the number of landmark positions. The score is assigned at each time-step for the label of landmark occurring at time t. Later, the prediction for the window length T is obtained by combining the scores into single prediction. Equation (22) shows the "sum rule" that is used as the fusion scheme for better results, which is theoretically superior to other schemes used in [44]. We applied softmax layer over Y to convert predictions into probabilities:

$$Y = \frac{1}{T}\sum_{t=1}^{T} y_t^L. \tag{22}$$

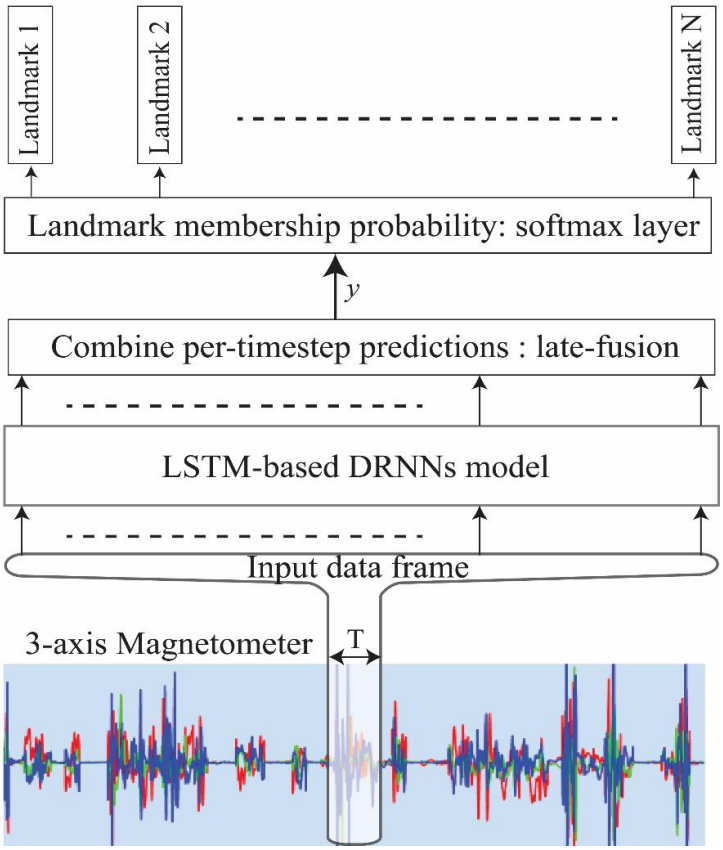


Figure 6. Proposed DRNN indoor positioning architecture. The inputs are raw signals collected from magnetometer, segmented them with windows of length T given to LSTM-based DRNN model. The output is class membership probability obtained from output prediction score for each timestamp, merged via late-fusion.

4.2 Unidirectional LSTM-Based DRNN Model

Figure 7 shows a unidirectional LSTM-based DRNN, which is used in the proposed system. A higher number of DRNN layers can help to transform raw data into a more abstract representation for learning temporal dependencies [16]. The input is the MFS, which is a discrete sequence of samples $(x_1, x_2, \dots, x_T)$ that are passed into the first layer at time t $(t = 1, 2, \dots, T)$.

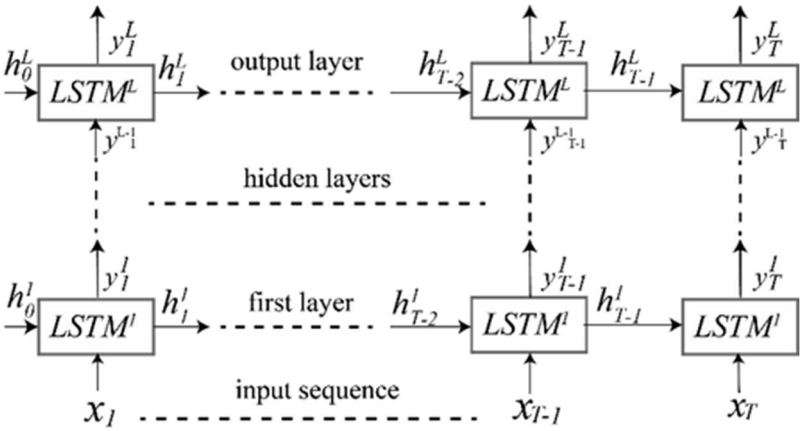


Figure 7. Unidirectional LSTM-based DRNN model.

Initially, the internal state I_0^l and the hidden state h_0^l are both set to zeros. The first layer output y_t^1 is obtained using the input sample x_t at time t, previous internal hidden state I_{t-1}^1 and previous hidden state h_{t-1}^1 given its parameter θ^1 as follows:

$$y_t^1, h_t^1, I_t^1 = LSTM^1(I_{t-1}^1, h_{t-1}^1, x_t;\ \theta^1), \tag{23}$$

The lower layers y_t^{l-1} is used as input by any upper layer l. If θ^l represents the parameter (b, U, W) of the LSTM cells for layer l, then (12) can be written as:

$$y_t^l, h_t^l, I_t^l = LSTM^l(I_{t-1}^l, h_{t-1}^l, y_t^{l-1};\ \theta^l). \tag{24}$$

The prediction at every time step in the window T is given by the outputs $(y_1^L, y_2^L, \ldots, y_T^L)$ from the top layer L.

4.3 Bidirectional LSTM-Based DRNN Model

This architecture uses a bidirectional LSTM-based DRNN, as shown in Figure 8. It includes two parallel LSTM tracks: forward and backward loops to exploit context from past and future to predict its label [20][45]. In the first

layer, the forward track ($LSTM^{f1}$) and backward track ($LSTM^{b1}$) reads input window T from left to right, and right to left respectively:

$$y_t^{f1}, h_t^{f1}, I_t^{f1} = LSTM^{f1}(I_{t-1}^{f1}, h_{t-1}^{f1}, x_t ; W^{f1}) \quad (25)$$

$$y_t^{b1}, h_t^{b1}, I_t^{b1} = LSTM^{b1}(I_{t-1}^{b1}, h_{t-1}^{b1}, x_t ; W^{b1}). \quad (26)$$

At each time step, the top layer L outputs a sequence of scores from both forward LSTM($y_1^{fL}, y_2^{fL}, \dots, y_T^{fL}$)and backward LSTM ($y_1^{bL}, y_2^{bL}, \dots, y_T^{bL}$).

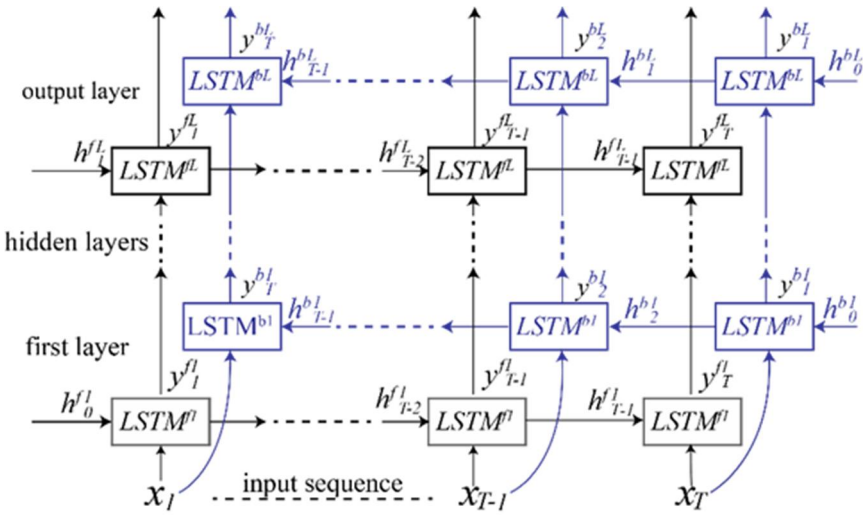


Figure 8. Bidirectional LSTM-based DRNN model with forward $LSTM^{fl}$ and backward $LSTM^{bl}$ track.

The combined scores $Y \in R^k$ represents landmark labels prediction for the window segment T . The late-fusion is the resulting results from both forward and backward tracks, which are merged together as follows:

$$Y = \frac{1}{T}\sum_{t=1}^{T}\left(y_t^{fL} + y_t^{bL}\right). \quad (27)$$

4.4 Cascaded Bidirectional and Unidirectional LSTM-based DRNN Model

The model architecture, shown in Figure 9, is inspired from [30][20]. In this architecture, the first layer is designed with a bidirectional RNN, whereas the upper layers are unidirectional.

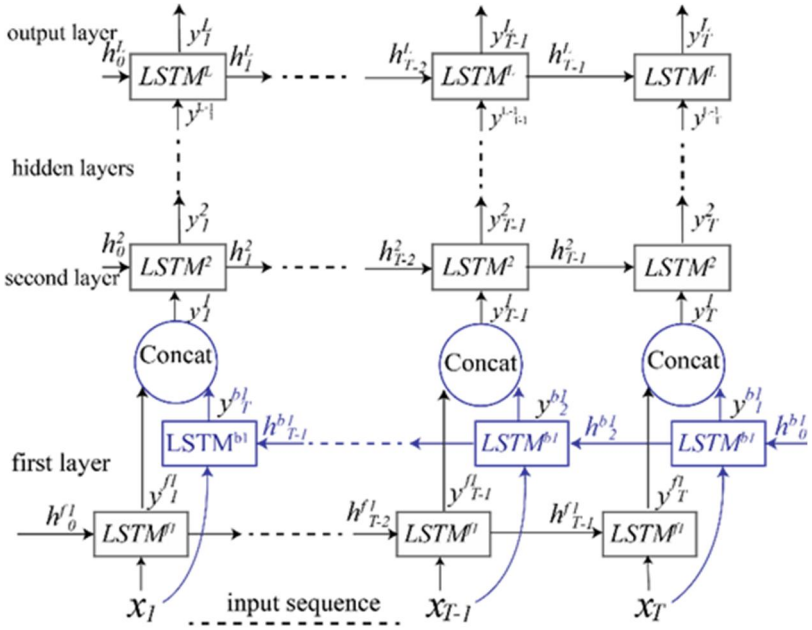


Figure 9. Cascaded unidirectional and bidirectional LSTM-based DRNN. The upper unidirectional layer is concatenated with the bidirectional first layer.

The first layer with a forward LSTM track $LSTM^{f1}$ generates an output $(y_1^{fL}, y_2^{fL}, \dots, y_T^{fL})$ and a backward track $LSTM^{b1}$ generates an output $(y_1^{bL}, y_2^{bL}, \dots, y_T^{bL})$. The two types of outputs are combined and fed into the second unidirectional layer to form a new output $(y_1^1, y_2^1, \dots, y_T^1)$:

$$y_t^1 = y_t^{f1} + y_{T-t+1}^{b1}. \tag{28}$$

The operation of the upper layers is with the unidirectional model explained in Section 5.2.1.

Chapter 5: Experimental Procedure

5.1 Setup and Methodology

The experiments were conducted in a corridor and a lab on the eighth floor of an IT department building, Chosun university in Korea with dimensions of 100m × 2.5m and 7m × 7m, respectively. The corridor contains magnetic

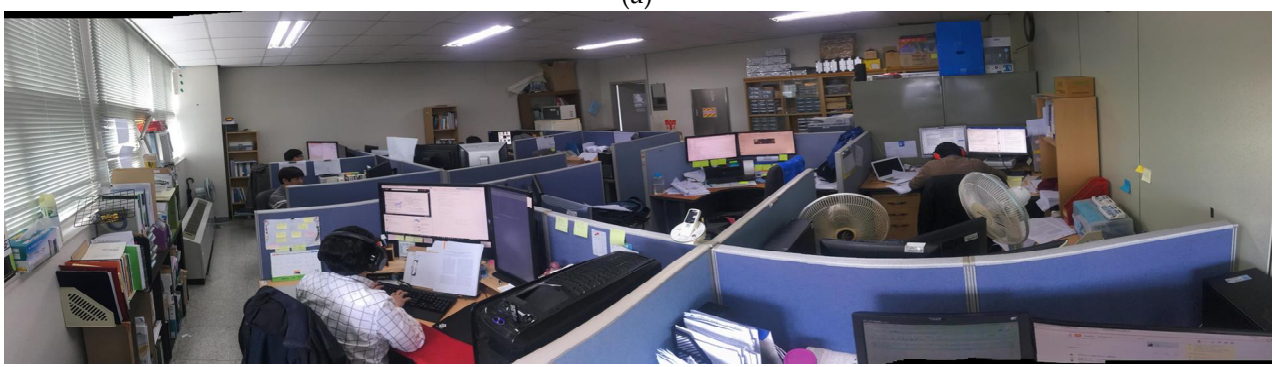
elements like iron doors, bearing column, and other building materials, which creates fluctuation to the MFS.

On the other hand, the lab has other factors such as computers, microwaves, and lab equipment that cause further significant magnetic field fluctuation. A pictorial representation of the test environment is shown in Figure 10.

The proposed model was evaluated through experiments. The layouts of the testbeds are shown in Figures 11 (c) and (d). The magnetic field data were collected by using an android smartphone, which has a Yamaha MS-3E magnetometer sensor. To make this process convenient, we developed an android application for the smartphone that sensed geomagnetic field.

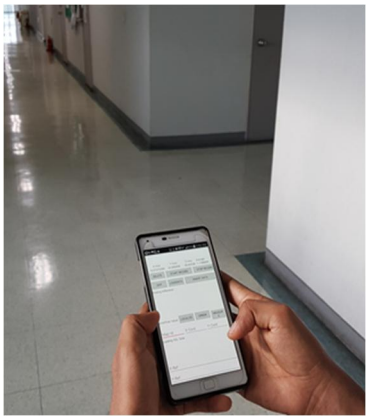
In the corridor, we marked 25 landmarks and measured magnetic signals by moving around the point in all possible directions as shown by the red eclipse dots in Figure 11 (d). While recording the data in the corridor, the orientation of the phone was held by a walking user as shown in Figure 11 (a) to avoid errors due to soft iron distortion.

(a)



(b)

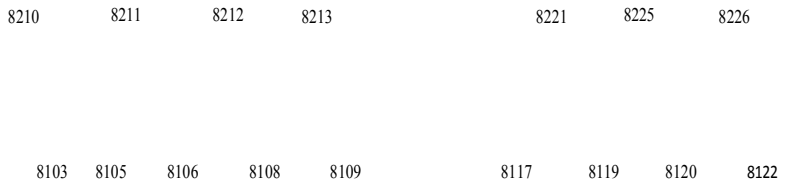
Figure 10. Experiment environment: (a) first testbed in the lab; (b) second testbed in the corrido



(a)

(b)

(c)



(d)

Figure 11. Data gathering methods: (a) orientation of smartphone while collecting data in corridor at each grid; (b) stand used in lab to gather data at each landmark; (c) layout of lab containing working desks surrounded with landmark denoted as red dots, where data are taken; (d) layout of corridor with landmarks denoted with red eclipse, where data are gathered.

However, in the lab, due to limited space, we used a movable stand as shown in Figure 11 (b). Also in the lab, we marked 17 landmarks and gathered magnetic field data at each point denoted by a red circular dot in Figure 11 (c). Finally, our dataset consisted of 25 landmark positions in the corridor and 17 landmark positions inside the lab. The size of training data in the corridor is about 71,300 MFS samples for the 25 landmarks and in the lab, it was about 19,500 MFS samples for 17 landmarks

5.2 Evaluation

We used Google TensorFlow as a deep learning framework, as it allowed us to design more detailed neural network model. Also, NVIDIA cuDNN and CUDA Toolkit, which provide parallel processing were used to drastically improve our training performance [31].

The proposed LSTM-DRNN positioning system uses the configuration and framework shown in Table 1.

Table 1. System configuration and framework for deep learning network

Category	Tools
CPU	Intel i5-7600 @3.50 GHz
GPU	NVIDEA GeForce GTX 1050 @ 2GB
RAM	DDR4 @8GB
Operating System	Windows 10 Enterprise
Language	Python 3.6
Library	Google Tensorflow 1.2, CUDA Toolkit 9.0, NVIDEA cuDNN v7.0

5.2.1 Network Training and Testing

We trained our DRNN model with the preprocessed geomagnetic data stored in a fingerprinting database or dataset. The dataset was divided such that 80% of the total data is used for training a model and the rest 20% is for testing process. The hyperparameters such as the number of hidden nodes, mini-batch size, number of iterations, learning rate etc. were chosen for the optimized model. Also, the biases and weights were initialized by using a standard normal distribution. The cost function $\mathcal{L}$ in (18) was obtained by using the mean cross-entropy between the true landmark labels and the predicted output labels. The ground truth labels indicate the true landmark labels for the segmented windows and were given in the dataset. They are arranged as a one-hot vector $O \in R^k$ with a value o_c associated with each landmark label c. The predicted classes $\hat{O} \in R^k$ contains the probability of each individual class ρ_c generated by our model:

$$\mathcal{L}(O, \hat{O}) = -\sum_{c=1}^{k} o_c \log \rho_c. \tag{29}$$

The Adam optimization algorithm was used to minimize the cost function $\mathcal{L}(O, \hat{O})$ by backpropagating its gradient and updating the model parameters [46]. The weight and other parameters were optimized with forward and backpropagation. In our work, we provide parameters for forward propagation and Google TensorFlow calculated all required back propagation steps. We used dropout technique as regularization to avoid overfitting in our model [44]. During a training iteration, the node is dropped out based on the dropout probability $\wp$, which represents the percentage of units to drop. The output of the final layer's hidden state is passed as an input to a fully connected layer, which uses a simplified hidden layer neural network to train the output

data using a softmax classifier. Also, the datasets were divided with different window lengths, as shown in Table 2. The optimal window length was selected based on their performance results by a "trial-and-error" method. The fixed length windows were used for training and testing, but during real-time data acquisition scenarios, we were able to use variable-length windows.

Table 2. Summary of DRNN input data to evaluate the proposed models. Training window length denotes the number of data samples in a window that was found to give the best results for each testbed.

Testbed	No. of Landmarks	Sensor	No. of samples per second	Training window length	No. of training samples	No. of testing samples
Corridor	25	Magnetometer	25	16	56982	14246
Lab	17	Magnetometer	25	4	15636	3909

We used two separate sets of hyperparameters due to a different number of data samples in the different testbeds, i.e., the corridor and lab. Figure 12 shows the training and testing accuracy along with a cost for the unidirectional DRNN model on our testbed. The training and testing accuracies increases with training epoch as the model generalizes to new data. Similarly, the training and testing cost decreases with each epoch as the model learns the data and reach an optimal value. In addition, the testing accuracy and cost follows the training accuracy and cost graph closely, which indicates the effectiveness of the dropout technique in the model for avoiding overfitting.

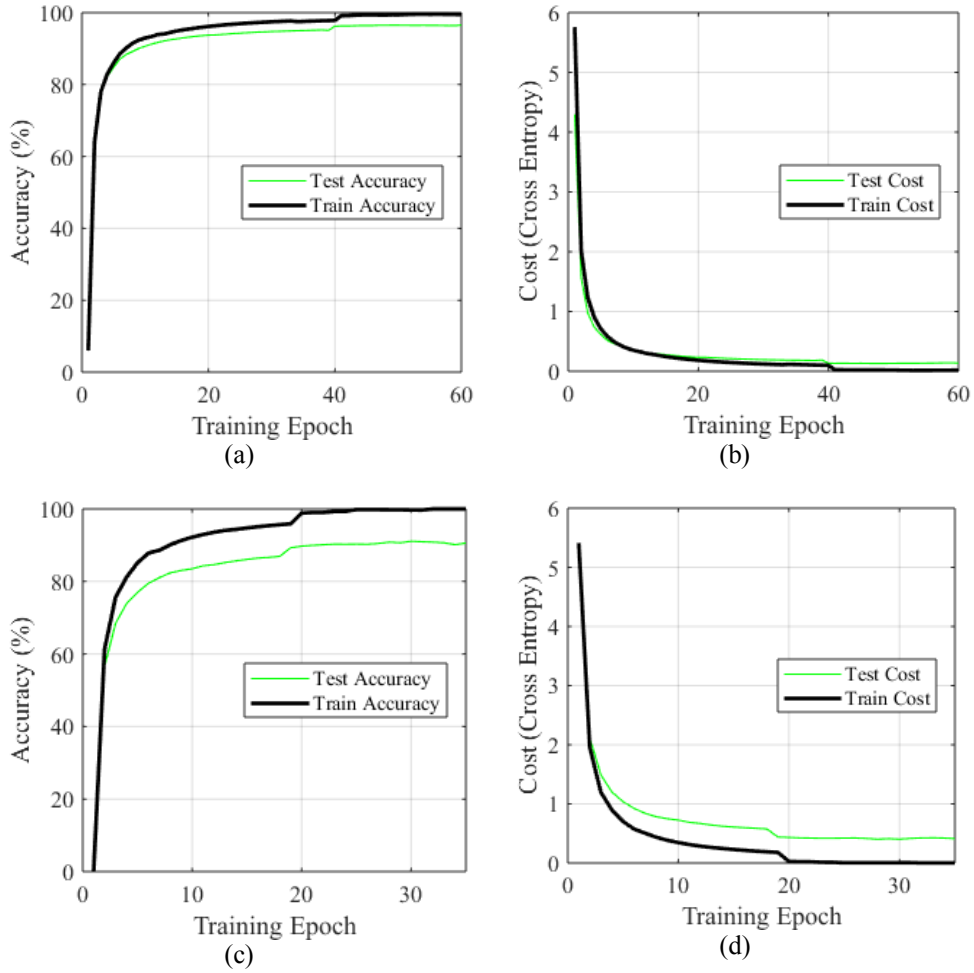


Figure 12. The accuracy and cost of our DRNN model for the testbed dataset over training iterations: (a) training and testing accuracies in lab; (b) cross-entropy cost between true landmarks and predicted landmarks for both training and testing in lab; (c) training and testing accuracy in corridor; (d) cross-entropy cost in corridor.

We divided our dataset into mini-batch for efficient memory usage and the problem of gradient explosion caused when the dataset is used as a single batch. When we used the small batch size, the training time was generally increased. This could be due to smaller step-sizes taken by the smaller mini-batch to

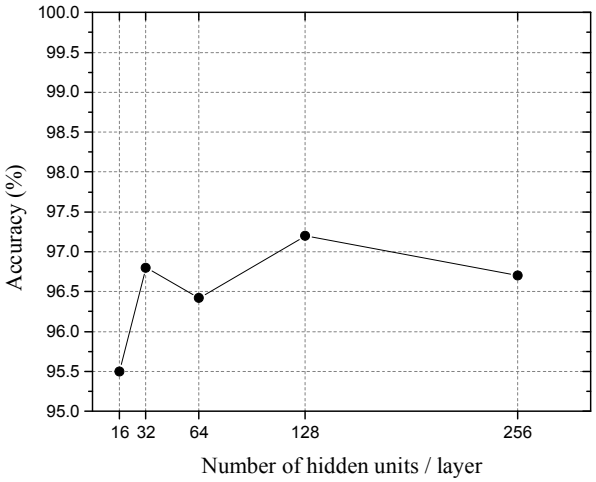
reduce the variances of gradient updates. For example, a batch size of 128 or 256 can process more data per mini-batch than that of 16 or 32. However, using a small mini-batch helped us to increase the accuracy. The configuration of the proposed DRNN system the was found to be best for our testbeds is listed in Table 3.

Table 3. Summary of hyperparameters used in two different testbeds

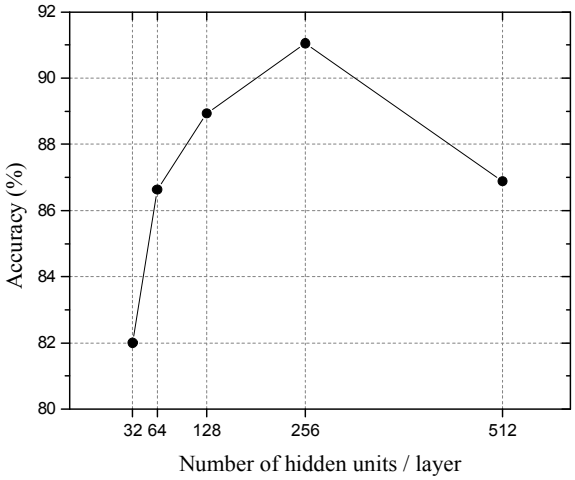
Parameters	**Corridor**	**Lab**
Neural Model	DRNN	DRNN
Loss Function	Cross Entropy	Cross Entropy
Optimizer	Adam	Adam
Learning Rate	0.001	0.001
No. of hidden node	256	128
Mini-batch size	100	50
No. of Landmark	25	17

The trained DRNN models were evaluated with test dataset. We found that the testing accuracy was greatly affected by a number of hidden nodes per layer and the mini-batch size. It was observed that the accuracy increased as the number of hidden nodes per layer increases. However, if we increased the number of hidden layers, the performance of the model was not necessarily good. This might be due to the difficulty in gradient propagation when we increase the number of layers. Figure 13 shows the test accuracy when we

changed the number of hidden nodes in each layer using a “trial-and-error” method. It can be seen that the best test accuracies for the lab and corridor were obtained with hidden units of 128 and 256, respectively.



(a)



(b)

Figure 13. Accuracy measurement with increasing number of hidden units per layer: (a) test accuracy for a different number of hidden units per layer in the lab; (b) test accuracy for a different number of hidden units per layer in the corridor.

The flow chart for the implementation of the training phase is shown in Figure 14.

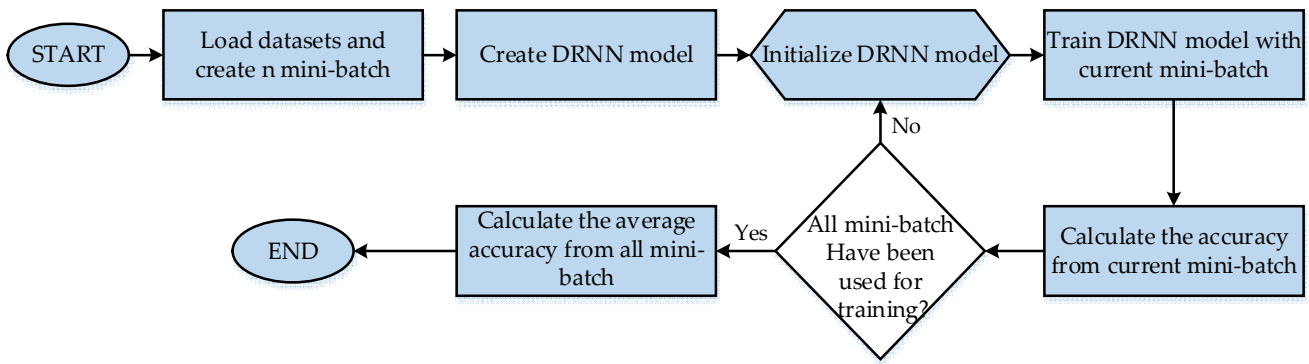
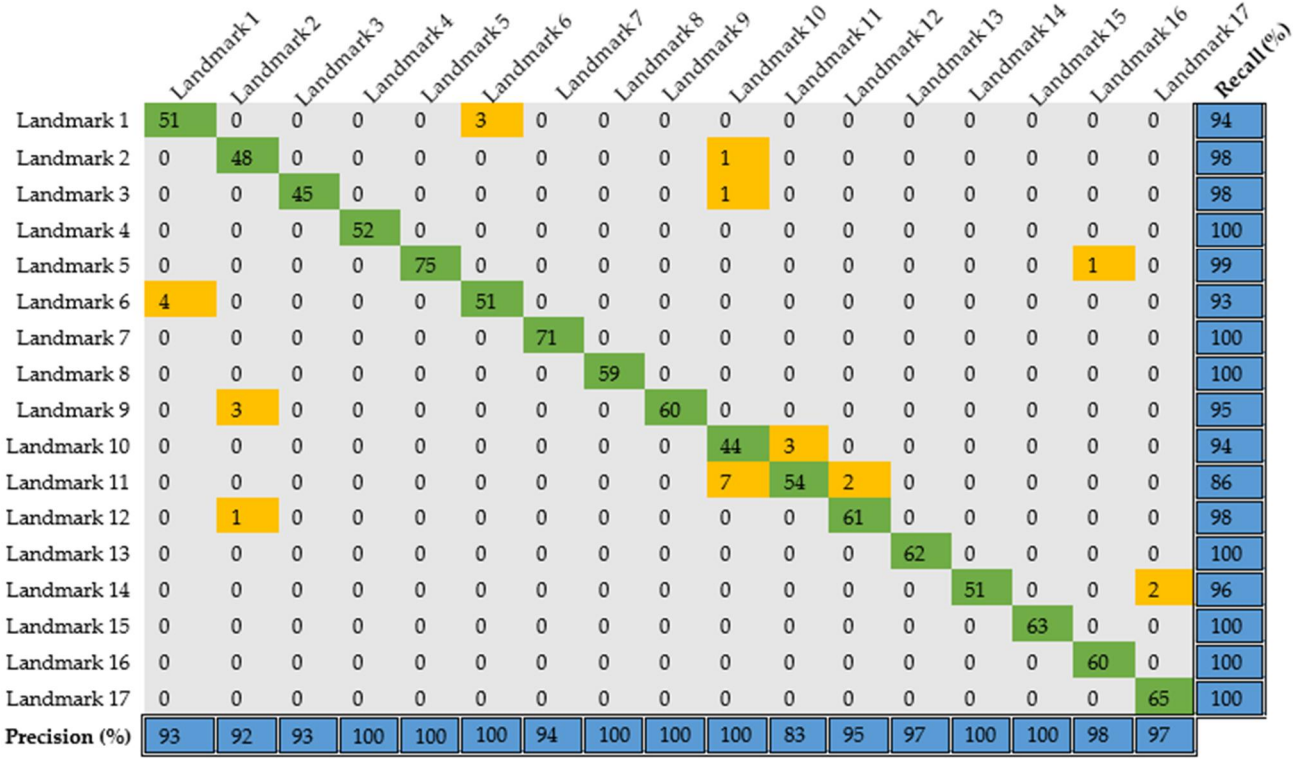


Figure 14. Flowchart representation for the training of DRNN model using mini-batch from dataset.

Chapter 6: Experimental Results and Discussion

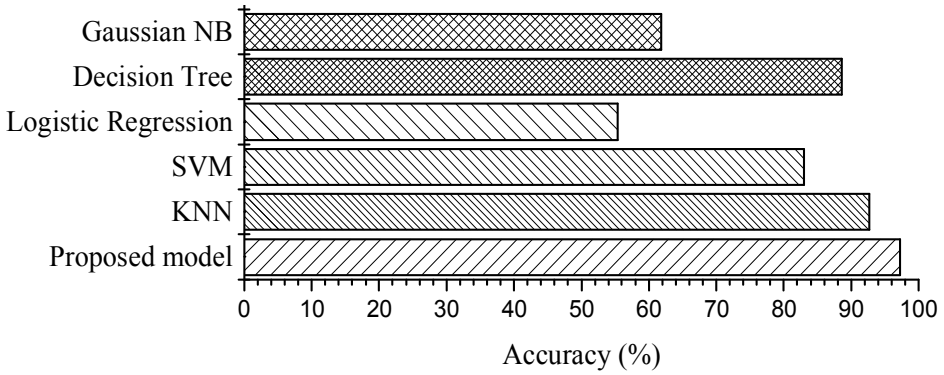
Our proposed LSTM-DRNN based positioning system was compared with other previously introduced machine learning methods tested on both testbeds. In the lab, a bidirectional DRNN model with three-layers yielded the best performance results with an overall classification accuracy of 97.20 %. The confusion matrix in Figure 15 (a) gives an overview of the classification results for the proposed model including the per-class precision and recall results in the test set. Figure 15 (b) and 15 (c) show a performance comparison of the proposed system with other machine learning methods such as k-nearest neighbor (KNN) [45], support vector machine (SVM), logistic regression, decision tree and Gaussian Naïve Bayes (GNB). However, in the corridor, we found that four layers of a unidirectional DRNN yields the best performance results including per-class precision and recall, as shown by the confusion matrix in Figure 16 (a). Here, the overall classification accuracy is 91.1 %. Also, we compared the performances with other machine learning algorithms. A comparison of accuracy and F1-score between our model and these algorithms can be seen in Figure 16 (b) and 16 (c) respectively.

The performance results clearly shows that all of the architectures performed very well with a dataset on both testbeds. The corridor is a wide space with MFS fluctuations mostly due to pillars and columns. However, the lab is a small space with a cluttered environment that has MFS fluctuations due to equipment, such as computers, microwave ovens, printers etc. Therefore, we included more landmarks in the lab compared to the corridor in terms of space density.

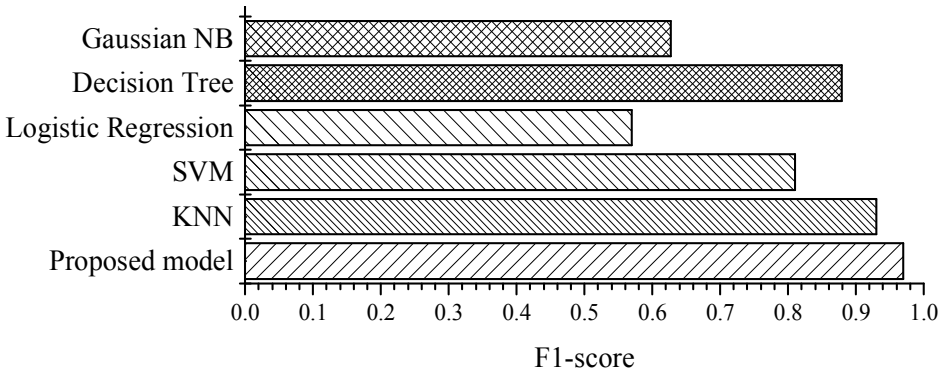


	Landmark 1	Landmark 2	Landmark 3	Landmark 4	Landmark 5	Landmark 6	Landmark 7	Landmark 8	Landmark 9	Landmark 10	Landmark 11	Landmark 12	Landmark 13	Landmark 14	Landmark 15	Landmark 16	Landmark 17	**Recall(%)**
Landmark 1	51	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	94
Landmark 2	0	48	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	98
Landmark 3	0	0	45	0	0	0	0	0	0	1	0	0	0	0	0	0	0	98
Landmark 4	0	0	0	52	0	0	0	0	0	0	0	0	0	0	0	0	0	100
Landmark 5	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	1	0	99
Landmark 6	4	0	0	0	0	51	0	0	0	0	0	0	0	0	0	0	0	93
Landmark 7	0	0	0	0	0	0	71	0	0	0	0	0	0	0	0	0	0	100
Landmark 8	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	0	100
Landmark 9	0	3	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	95
Landmark 10	0	0	0	0	0	0	0	0	0	44	3	0	0	0	0	0	0	94
Landmark 11	0	0	0	0	0	0	0	0	0	7	54	2	0	0	0	0	0	86
Landmark 12	0	1	0	0	0	0	0	0	0	0	0	61	0	0	0	0	0	98
Landmark 13	0	0	0	0	0	0	0	0	0	0	0	0	62	0	0	0	0	100
Landmark 14	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	2	96
Landmark 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	63	0	0	100
Landmark 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	100
Landmark 17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65	100
Precision (%)	93	92	93	100	100	100	94	100	100	100	83	95	97	100	100	98	97	

(a)



(b)



(c)

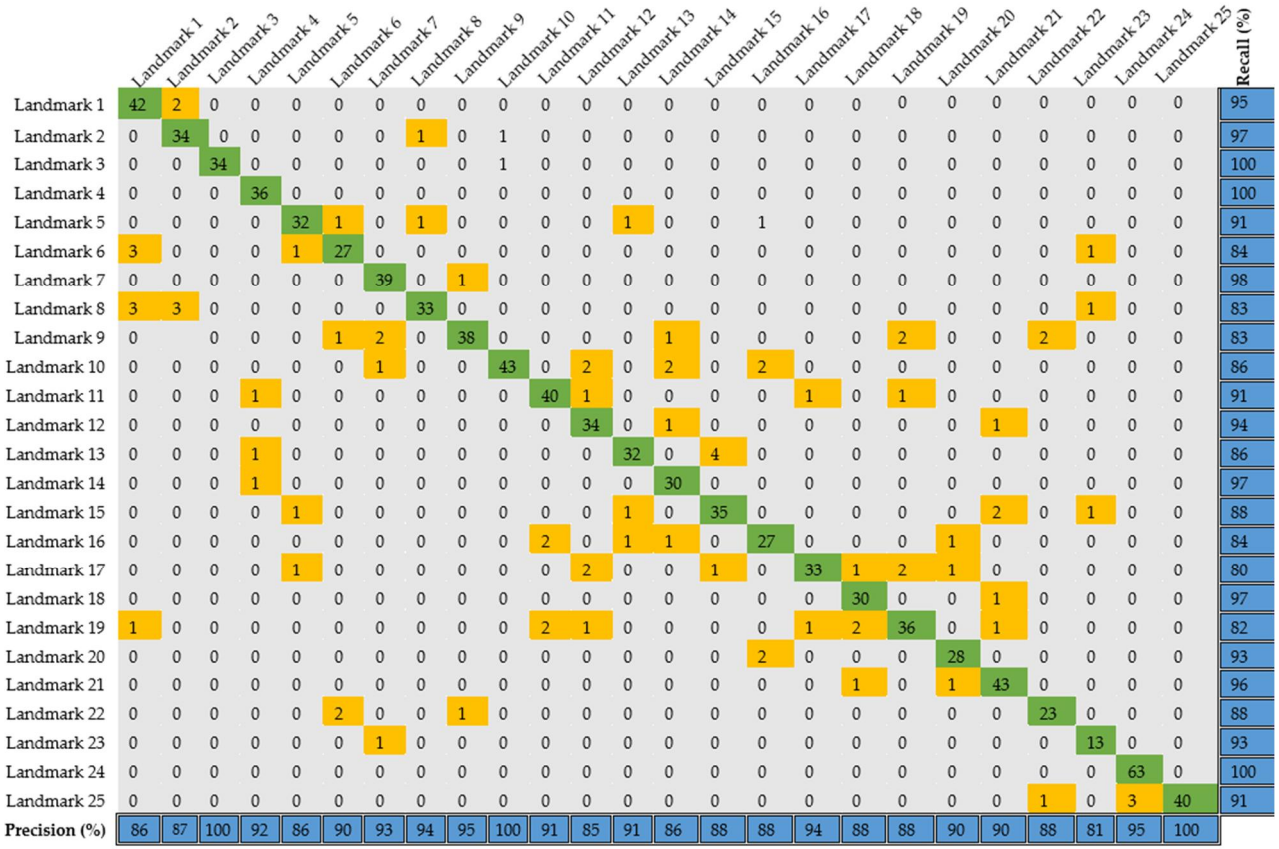
Figure 15. Performance results of the proposed bidirectional DRNN model in lab (a) confusion matrix for the landmark classification in the lab with per-class precision and recall; (b) Accuracy comparison of the proposed model with other methods; (c) F1-score comparison of the proposed model with other methods.

It proves that our models can be effective for a broad range of landmark classification in various indoor environments. Table 4 contains a performance summary of our models in the two test environments.

Table 4. Performance summary of our model on dataset of two testbeds

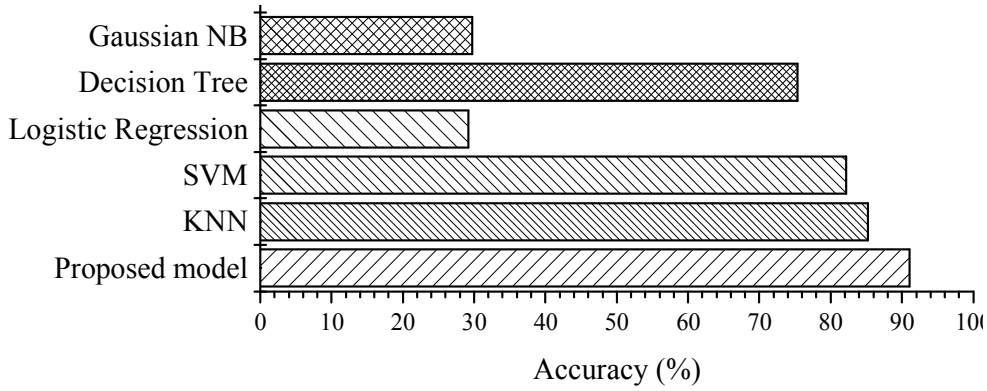
Model	Dataset	Overall Accuracy	Average Precision	Average Recall	F1 Score
Unidirectional DRNN	Corridor	91.1	90.6	91.1	0.90
Bidirectional DRNN	Corridor	88.5	88.3	87.7	0.88
Cascaded DRNN	Corridor	90.1	89.5	89.6	0.89
Unidirectional DRNN	Lab	95.6	95.6	95.8	0.95
Bidirectional DRNN	Lab	97.2	97	97.1	0.97
Cascaded DRNN	Lab	96	96.2	96.1	0.96

As the sample size grew, the conventional shallow-structured methods like KNN, SVM, and logistic regression have limited modeling capability and cannot extract reliable features from a large dataset of fluctuating MFS. However, including more layers in the DRNN helped the model to extract discriminative features. These features were exploited for effective learning and distinguishing more complex pattern formed by MFS at landmarks. In addition, using DRNNs to capture sequential and temporal dependencies provided a significant improvement in performance.

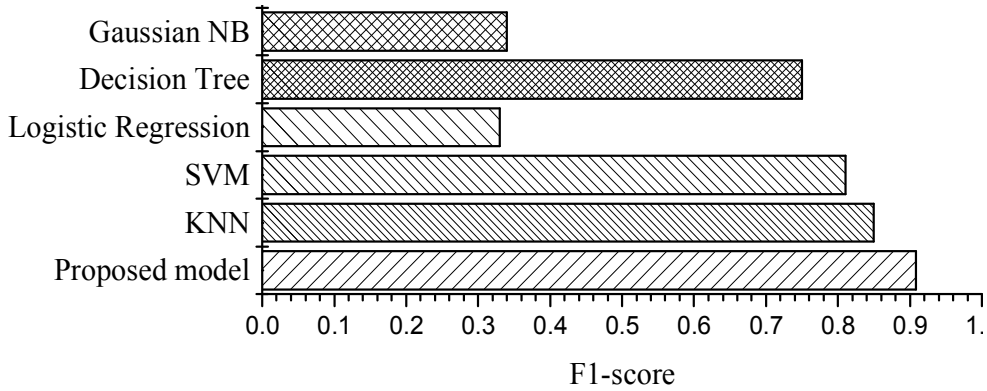


	Landmark 1	Landmark 2	Landmark 3	Landmark 4	Landmark 5	Landmark 6	Landmark 7	Landmark 8	Landmark 9	Landmark 10	Landmark 11	Landmark 12	Landmark 13	Landmark 14	Landmark 15	Landmark 16	Landmark 17	Landmark 18	Landmark 19	Landmark 20	Landmark 21	Landmark 22	Landmark 23	Landmark 24	Landmark 25	Recall (%)
Landmark 1	42	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95
Landmark 2	0	34	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97
Landmark 3	0	0	34	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
Landmark 4	0	0	0	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
Landmark 5	0	0	0	0	32	1	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	91
Landmark 6	3	0	0	0	1	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	84
Landmark 7	0	0	0	0	0	0	39	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98
Landmark 8	3	3	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	83
Landmark 9	0		0	0	0	1	2	0	38	0	0	0	1	0	0	0	0	2	0	0	2	0	0	0	0	83
Landmark 10	0	0	0	0	0	0	1	0	0	43	0	2	0	2	0	2	0	0	0	0	0	0	0	0	0	86
Landmark 11	0	0	0	1	0	0	0	0	0	0	40	1	0	0	0	0	1	0	1	0	0	0	0	0	0	91
Landmark 12	0	0	0	0	0	0	0	0	0	0	0	34	0	1	0	0	0	0	0	0	1	0	0	0	0	94
Landmark 13	0	0	0	1	0	0	0	0	0	0	0	0	32	0	4	0	0	0	0	0	0	0	0	0	0	86
Landmark 14	0	0	0	1	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	97
Landmark 15	0	0	0	0	1	0	0	0	0	0	0	0	1	0	35	0	0	0	0	0	2	0	1	0	0	88
Landmark 16	0	0	0	0	0	0	0	0	0	0	2	0	1	1	0	27	0	0	0	1	0	0	0	0	0	84
Landmark 17	0	0	0	0	1	0	0	0	0	0	0	2	0	0	1	0	33	1	2	1	0	0	0	0	0	80
Landmark 18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	1	0	0	0	0	97
Landmark 19	1	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	1	2	36	0	1	0	0	0	0	82
Landmark 20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	28	0	0	0	0	0	93
Landmark 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	43	0	0	0	0	96
Landmark 22	0	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	88
Landmark 23	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	93
Landmark 24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	63	0	100
Landmark 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	3	40	91	
Precision (%)	86	87	100	92	86	90	93	94	95	100	91	85	91	86	88	88	94	88	88	90	90	88	81	95	100	

(a)



(b)



(c)

Figure 16. Performance results of the proposed unidirectional DRNN model in corridor (a) confusion matrix for the test in a corridor along with per-class precision and recall; (b) Comparison of accuracy among proposed model and other methods; (c) Comparison of F1-score among proposed model and other methods.

Chapter 7: Conclusion

In this thesis, we have implemented three unique LSTM-based DRNN architectures for indoor landmark classification using MFS. The performance of three LSTM-based DRNN models: unidirectional, bidirectional and cascaded, was evaluated on two testbed's data set. The use of DRNN based architectures allows us to classify variable-length window.

Moreover, we first verified experimentally the feasibility of using MFS for landmark classification and empirically evaluated our models using experiments on datasets from two testbeds. Although the training phase was computationally demanding, the test phase is fast and suitable for real-time indoor landmark classification. Experimental results show that the proposed models outperform other state-of-the-art methods. The performance improvement was mainly due to the ability of our models to extract more discriminative features by using deep layers in various landmark positions. Furthermore, by the functionality of DRNNs, our models were able to capture temporal dependencies between input magnetic field data.

In the future, researchers can extend our work by including odometry and Radio field (RF) data for a large-scale commercial indoor environment. The big improvement would be to develop a way to calibrate the magnetic sensor and to predict its rotation matrix accurately with respect to the smartphone rotation. More work can be done to develop an efficient way to collect training data in an offline phase and build greater training set.

Bibliography

[1] B. P. Misra and P. Enge, "*Global Positioning System : Signals, Measurements and Performance*," Second Edition, Lincoln, MA: Ganga-Jamuna Press, 2005.

[2] X. Wang, S. Mao, S. Pandey, and P. Agrawal, "CA2T: Cooperative Antenna Arrays Technique for pinpoint indoor localization," *Procedia Comput. Sci.*, vol. 34, pp. 392–399, 2014.

[3] M. X. Gong, B. Hart, and S. Mao, "Advanced Wireless LAN Technologies: IEEE 802.11AC and Beyond," *GetMobile Mob. Comp. Comm.*, vol. 18, no. 4, pp. 48–52, Jan. 2015.

[4] S. Subedi and J.-Y. Pyun, "Practical Fingerprinting Localization for Indoor Positioning System by Using Beacons," *J. Sensors*, vol. 2017, pp. 1–16, Dec. 2017.

[5] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, 2007.

[6] P. B. and V. N. Padmanabhan, "RADAR: An in-building RF based user location and tracking system," *Proc. IEEE INFOCOM 2000. Conf. Comput. Commun. Ninet. Annu. Jt. Conf. IEEE Comput. Commun. Soc. (Cat. No.00CH37064)*, vol. 2, no. c, pp. 775–784, 2000.

[7] M. Youssef and A. Agrawala, "The Horus WLAN Location Determination System," *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, pp. 205–218, 2005.

[8] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-Based Fingerprinting

for Indoor Localization: A Deep Learning Approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, 2017.

[9] X. Wang, L. Gao, and S. Mao, "BiLoc: Bi-Modal Deep Learning for Indoor Localization with Commodity 5GHz WiFi," *IEEE Access*, vol. 5, pp. 4209–4220, 2017.

[10] K. Yamazaki *et al.*, "Analysis of Magnetic Disturbance Due to Buildings," *IEEE Trans. Magn.*, vol. 39, no. 5 II, pp. 3226–3228, 2003.

[11] G. Casinovi, A. Geri, and G. M. Veca, "Magnetic Field Near a Concrete Wall During a Lightning Stroke," *IEEE Trans. Magn.*, vol. 25, no. 5, pp. 4006–4008, 1989.

[12] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman, "Indoor location sensing using geo-magnetism," *Proceedings of the 9th international conference on Mobile systems, applications, and services - MobiSys '11*, p. 141, 2011.

[13] W. Storms, J. Shockley, and J. Raquet, "Magnetic field navigation in an indoor environment," *Ubiquitous Position. Indoor Navig. Locat. Based Serv. UPINLBS 2010*, 2010.

[14] X. Wang, L. Gao, and S. Mao, "CSI Phase Fingerprinting for Indoor Localization with a Deep Learning Approach," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1113–1123, 2016.

[15] B. Bimal, S. Hwang, and J. Pyun, "An Efficient Geomagnetic Indoor Positioning System Using Smartphones," *The 3rd International Conference on Next Generation Computing (ICNGC2017b)*, pp. 3-6, July 2018.

[16] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," *2013 IEEE Int. Conf. Acoust. Speech*

Signal Process., no. 6, pp. 6645–6649, 2013.

[17] M. Sundermeyer, R. Schl, and H. Ney, "LSTM Neural Networks for Language Modeling," *Proc. Interspeech*, pp. 194–197, 2012.

[18] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 4507–4515, 2015.

[19] Z. Chen and C. Wang, "Modeling RFID signal distribution based on neural network combined with continuous ant colony optimization," *Neurocomputing*, vol. 123, pp. 354–361, 2014.

[20] A. Murad and J.-Y. Pyun, "Deep Recurrent Neural Networks for Human Activity Recognition," *Sensors*, vol. 17, no. 11, p. 2556, 2017.

[21] J. Korhonen, T. Ojala, M. Klemola, and P. Vaananen, "mTag - Architecture for Discovering Location Specific Mobile Web Services Using RFID and Its Evaluation with Two Case Studies," *Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services (AICT-ICIW'06)*, pp. 191–191, 2006.

[22] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Trans. Inf. Syst.*, vol. 10, no. 1, pp. 91–102, Jan. 1992.

[23] Z. Li, C. Liu, J. Gao, and X. Li, "An Improved WiFi/PDR Integrated System Using an Adaptive and Robust Filter for Indoor Localization," *ISPRS Int. J. Geo-Information*, vol. 5, no. 12, p. 224, Nov. 2016.

[24] Q. Chang, S. V. Velde, W. Wang, Q. Li, H. Hou, and S. Heidi, "Wi-Fi Fingerprint Positioning Updated by Pedestrian Dead Reckoning for Mobile Phone Indoor Localization," *Lect. Notes Electr. Eng.*, vol. 342,

2015.

[25] V. Radu and M. K. Marina, “HiMLoc: Indoor smartphone localization via activity aware Pedestrian Dead Reckoning with selective crowdsourced WiFi fingerprinting,” *International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–10, 2013.

[26] J. Haverinen and A. Kemppainen, “Global indoor self-localization based on the ambient magnetic field,” *Rob. Auton. Syst.*, vol. 57, no. 10, pp. 1028–1035, 2009.

[27] B. Gozick, K. P. Subbu, R. Dantu, and T. Maeshiro, “Magnetic maps for indoor navigation,” *IEEE Trans. Instrum. Meas.*, vol. 60, no. 12, pp. 3883–3891, 2011.

[28] H. Xie, T. Gu, X. Tao, H. Ye, and J. Lv, “MaLoc,” *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp ’14 Adjunct*, pp. 243–253, 2014.

[29] M. Myllymäki, T. Mrkvička, P. Grabarnik, H. Seijo, and U. Hahn, “Global envelope tests for spatial processes,” *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 79, no. 2, pp. 381–404, 2017.

[30] Y. Wu *et al.*, “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,” *arXiv:1609.08144*, pp. 1–23, 2016.

[31] M. Abadi *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” *arXiv:1603.04467,* 2016.

[32] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Röning, “Magnetic field-based SLAM method for solving the localization problem in mobile robot floor-cleaning task,” *IEEE 15th Int. Conf. Adv. Robot. New Boundaries Robot. ICAR 2011*, pp. 198–203, 2011.

[33] J. Jung, T. Oh, and H. Myung, "Magnetic field constraints and sequence-based matching for indoor pose graph SLAM," *Rob. Auton. Syst.*, vol. 70, pp. 92–105, 2015.

[34] D. Navarro and G. Benet, "Magnetic map building for mobile robot localization purpose," *ETFA 2009 - 2009 IEEE Conf. Emerg. Technol. Fact. Autom.*, pp. 4–7, 2009.

[35] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, 2017.

[36] J. Ma, X. Li, X. Tao, and J. Lu, "Cluster filtered KNN: A WLAN-based indoor positioning scheme," *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 1–8, 2008.

[37] C. Cortes, and V. Vapnik, "Support-Vector Networks," *Machine learning*, vol. 20, no. 3, pp. 273-297, 1995.

[38] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery,* vol. 2, no. 2, pp. 121-167, 1998.

[39] E. Osuna, R. Freund, and F. Girosit, "Training support vector machines: an application to face detection," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 130–136, 1997.

[40] Breiman, J. Friedman, R. Olshen, and C. Stone, "*Classification and regression trees*," Wadsworth International Group, Belmont, CA, 1984.

[41] B. Efron, "Title: Bayes' Theorem in the Twenty-First Century," *Science 2013,* vol. 340, pp. 1177-1178, 2013.

[42] F. E. Harrell, "Binary Logistic Regression," *Regression Modeling*

Strategies, pp. 215–267, Springer□Verlag, New York, 2001.

[43] F. E. Harrell, "Ordinal Logistic Regression," *Regression Modeling Strategies*, pp. 331–343, Springer□Verlag, New York, 2001.

[44] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout Improves Recurrent Neural Networks for Handwriting Recognition," *Proc. Int. Conf. Front. Handwrit. Recognition, ICFHR*, vol. 2014–Decem, pp. 285–290, 2014.

[45] M. Yao and B. Vocational, "Research on Learning Evidence Improvement for k NN Based Classification Algorithm," *Int. J. Database Theory Appl.*, vol. 7, no. 1, pp. 103–110, 2014.

[46] I. Goodfellow, Y. Bengio, and A. Courville, "Book Review: Deep Learning," *Deep Learn.*, vol. 22, no. 4, pp. 351–354, 2016.