SPECIAL SECTION ON AI-DRIVEN BIG DATA PROCESSING:

THEORY, METHODOLOGY, AND APPLICATIONS

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 15, 2019, accepted February 24, 2019, date of publication March 5, 2019, date of current version March 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2902573

Geomagnetic Field Based Indoor Landmark Classification Using Deep Learning

BIMAL BHATTARAI, ROHAN KUMAR YADAV, HUI-SEON GANG, AND JAE-YOUNG PYUN

Department of Information Communication Engineering, Chosun University, Gwangju 501-759, South Korea

Corresponding author: Jae-Young Pyun (jypyun@chosun.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology under Grant NRF-2016R1D1A1B03935889.

ABSTRACT The unstable nature of radio frequency 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 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 paper, we introduce deep recurrent neural networks (DRNNs) to build a model that is capable of capturing long-range dependencies in variable-length input sequences. The use 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, we can divide the indoor space into landmarks with magnetic field values and find the position of the user in a particular area inside the building. We present long short-term memory DRNNs for spatial/temporal sequence learning of magnetic patterns and evaluate their positioning performance on our testbed datasets. The experimental results show that our proposed models outperform other traditional positioning approaches with machine learning methods, such as support vector machine and k-nearest neighbors.

INDEX TERMS Deep recurrent neural network (DRNN), fingerprinting, geomagnetic field, long short-term memory (LSTM).

I. INTRODUCTION

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 indoor localization. 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]–[9]. Although these methods can achieve the desired accuracy at an acceptable cost, they cannot work effectively when a RF signal is weak. Also, these methods need expensive external devices, such as wireless access points (WAP) or Bluetooth beacons 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 steel, 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]. This fingerprinting positioning method is usually divided into two phases: training and testing. In the training phase, the dataset is prepared by collecting the MFS at all reference points and stored in the positioning server. In the testing phase, the real-time MFS data are collected and given to the positioning server to find out the current location. The performance can be evaluated by performance metrics, such as accuracy, precision, recall, and F1 score.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhanyu Ma.

We utilize the unique signal features of the magnetic field gathered from a smartphone for fingerprinting-based classification. 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 direction in some cases. Here, we adopt a deep learning model, which can perform effectively to classify landmarks based on the magnetic signal variations. In the area of deep learning, there is growing interest in the recurrent neural network (RNN), which has been used in many technical applications, such as speech recognition, language modeling, video processing, and many other sequence labeling tasks [15]–[18]. The reason behind its promising performance is its ability to exploit contextual information and learn the temporal dependencies in variable-length data.

In this paper, we use long short-term memory (LSTM)-based deep RNNs (DRNNs) to classify 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 advancing to unidirectional from bidirectional RNNs [19]. Moreover, we test these deep learning models with different testbeds to validate their performance at classifying various landmarks. The major contributions of our 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. To the best of our knowledge, this is the first work to do so.

The rest of the paper is organized as follows: Section II provides a brief review of previous works using the magnetic field for indoor positioning. Section III and IV present a preliminary analysis of magnetic field data and a background overview of RNNs, respectively. Then, the proposed architecture and experimental setup are explained in Section V and VI, respectively. Additionally, experimental results and analysis are presented in Section VII. Finally, Section VIII contains a conclusion of our work.

II. RELATED WORK

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 [20]. 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. [21] and [22] 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 depending upon moving distance and rotation was accurate.

Haverinen and Kemppainen [23] implemented the same SLAM with a human by replacing the odometers with pedestrian dead-reckoning (PDR). However, due to lack of proper odometric information, the performance was not as good as when using odometers.

In another work, Navarro and Benet [24] 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 [25] mobile phones were used to measure MFS and interpreted 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.* [26] attempted magnetic landmark localization with the MFS created from pillars of a building. They measured the sequence of peak values and matched these values to pre-obtained landmarks' MFS. However, in their research, the magnetic landmarks are defined with a prior knowledge that columns are ferromagnetic objects.

Recently, the use of deep neural networks (DNNs) is making a big impact in various research fields. In [27], the five-layer DNN classifier has been used with dynamic acoustic features. They proposed a scoring method using human log-likelihoods (HLLs) along with mathematical verification suggesting that their method can overcome other classical log-likelihood ratio (LLR) scoring methods. Also, the use of a LSTM neural network has been found to be more efficient in natural language processing. In [28], it has been used for speech recognition based on short utterance.

In our work, we advance the work of [25] and [26] by using DRNNs in a two-dimensional space. In terms of use of DRNNs, we used an LSTM network, which is a popular RNN for dealing with long-range dependencies [29], to train our magnetic field data. This LSTM network model is more flexible for classifying variable-length windows, in opposition to the fixed-length windows used by convolutional neural networks (CNNs).

III. MAGNETIC FIELD PRELIMINARIES

The geomagnetic field is present on the surface of the earth with a magnitude from 0.22 to 0.65 Gauss (22 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 Fig. 1. Later, the data were analyzed to see the statistical significance of the magnetic field over a period of time.

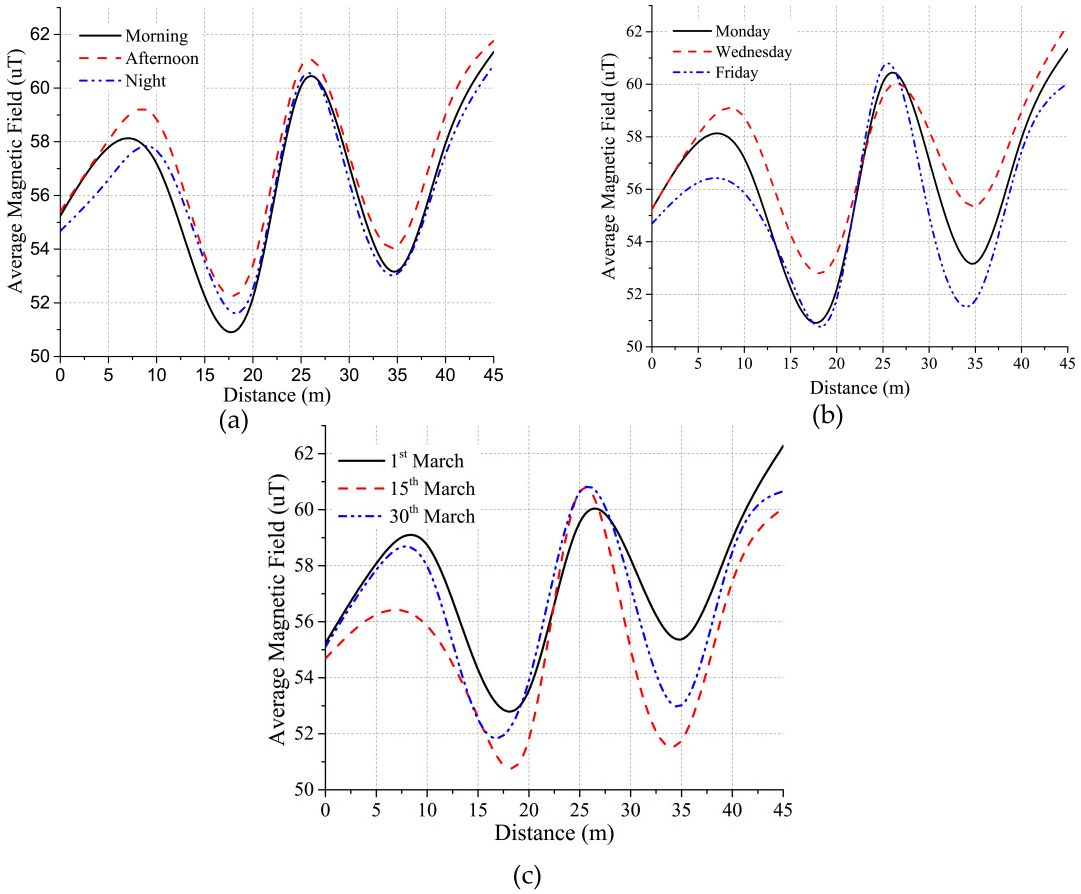


FIGURE 1. Stability of magnetic field value over time. (a) Variation over a day. (b) Variation over a week. (c) Variation over a month.

There were no statistically significant differences between magnetic field data from different times 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) [30]. 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 under the assumption 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. 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 a global frame system. Since the orientation of the smartphone plays an important role in positioning, we have also gathered magnetic field data with a quaternion-derived rotation matrix. Thus, we had to obtain the 3×3 rotation matrix R, which is the change from the device coordinate system to the global coordinate system. Given any vector m in the device coordinate system, the corresponding vector m_{rot} in the global coordinate system can be obtained by multiplying m with R.

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

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

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

IV. BACKGROUND: RECURRENT NEURAL NETWORK

A. RECURRENT NEURAL NETWORK

A recurrent neural network (RNN) is a class of deep neural network that contains cyclic connections that allow it to learn the temporal dynamics of 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

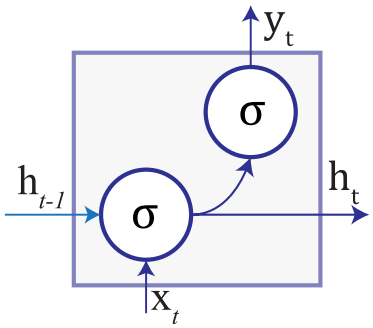


FIGURE 2. Schematic diagram of an RNN node.

features. As shown in Fig. 2, each RNN node generates the output y_t and current hidden state h_t by using the current input x_t and previous hidden stateh_{t-1} based on the following equations:

$$h_t = \sigma_H \left(W_{HH} h_{t-1} + W_{IH} x_t + b_h\right) \tag{3}$$
$$y_t = \sigma_O \left(W_{HO} h_t + b_y\right), \tag{4}$$

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

B. LONG SHORT-TERM MEMORY (LSTM)

The traditional RNN is unable to handle long sequences of data. In addition, training RNNs can be challenging due to vanishing and exploding gradients, which create a problem when backpropagating through long-range temporal intervals [31]. 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 Fig. 3.

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

- Input gatei_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 content regarding the internal state.
- Input modulation gate g_t provides the input to the memory cell.
- Internal state I_t controls cell internal recurrence.

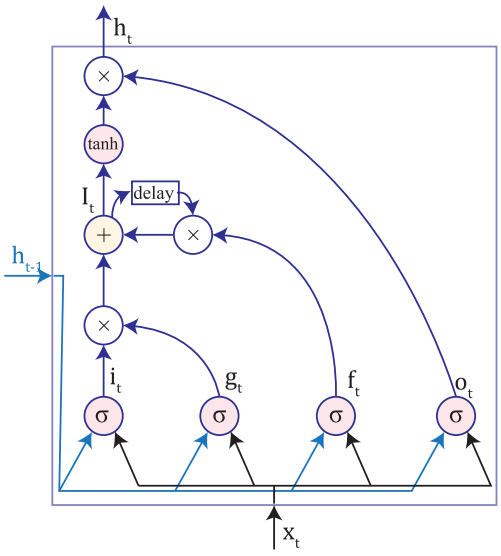


FIGURE 3. Schematic diagram of a LSTM cell structure with an internal recurrence $\mathbf{c_t}$ and outer recurrence $\mathbf{h_t}$.

- Hidden state h_t contains information from previous samples within the context window.

$$i_t = \sigma \left(b_i + U_i x_t + W_i h_{t-1}\right) \tag{5}$$
$$f_t = \sigma \left(b_f + U_f x_t + W_f h_{t-1}\right) \tag{6}$$
$$o_t = \sigma \left(b_o + U_o x_t + W_o h_{t-1}\right) \tag{7}$$
$$g_t = \sigma \left(b_g + U_g x_t + W_g h_{t-1}\right) \tag{8}$$
$$I_t = f_t I_{t-1} + g_t i_t \tag{9}$$
$$h_t = \tanh\left(I_t\right) o_t, \tag{10}$$

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 essentially focused 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 cell gates, as shown in (5) – (10).

V. PROPOSED ARCHITECTURES

A. SYSTEM ARCHITECTURE

The architecture of the proposed DRNN-based landmark classification system is shown in Fig. 4. 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. 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.

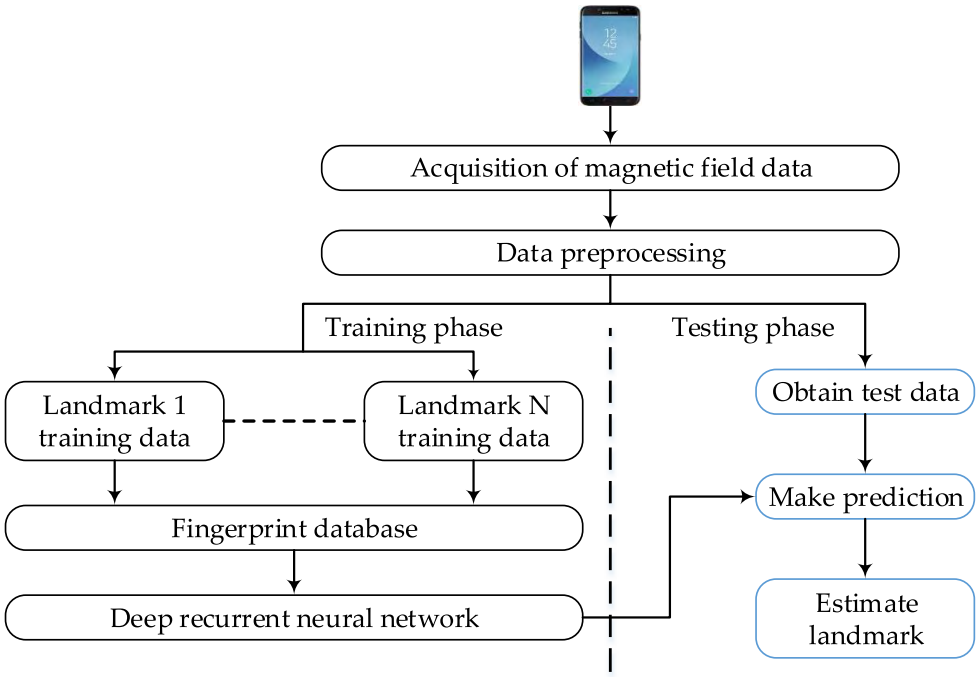


FIGURE 4. Architecture of the proposed LSTM-DRNN positioning system representing the training and testing phases.

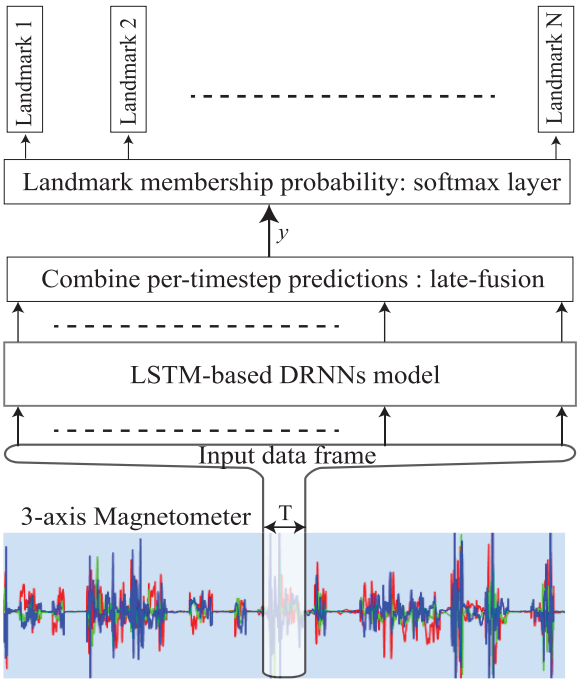


FIGURE 5. Proposed DRNN indoor positioning architecture. The inputs are raw signals obtained from a magnetometer that are then segmented with windows of length *T* given to the LSTM-based DRNN model. The output is the class membership probability obtained from the output prediction score for each timestamp, merged via late-fusion.

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 of the landmark's classification can be dependent on the training model of the proposed LSTM-DRNN.

B. DRNN ARCHITECTURE

Fig. 5 presents a schematic diagram of the proposed DRNN indoor positioning system. It performs direct mapping from magnetometer inputs to different landmarks. A specific time window is used to classify the landmark's position. The input contains a discrete sequence of equally spaced samples $(x_1, x_2, \ldots, x_T)$, where each data point x_t is a vector of individual MFS samples D observed by the magnetic sensor at time$t$. These samples are passed to a 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 $(y_1^L, y_2^L, \ldots, y_T^L)$, where $y_t^L \in R^k$ is a vector of prediction scores for a given input sample x_t, L is the number of DRNN layers or top layer, and k is the number of landmark positions. A score is assigned at each time-step for the label of a landmark occurring at time$t$. Later, the prediction for the entire window T is obtained by adding the scores into a single prediction. Equation (11) shows the "sum rule" that is used as the fusion scheme for better results, which is theoretically superior to other schemes used in [32]. We applied a softmax layer over Y to convert predictions into probabilities:

$$Y = \frac{1}{T} \sum_{t=1}^{T} y_t^L. \tag{11}$$

1) UNIDIRECTIONAL LSTM-BASED DRNN MODEL

We used a unidirectional LSTM-based DRNN, as shown in Fig. 6. A higher number of DRNN layers can help

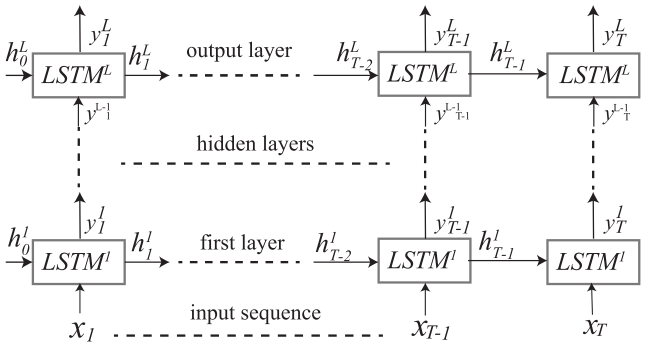


FIGURE 6. Unidirectional LSTM-based DRNN model.

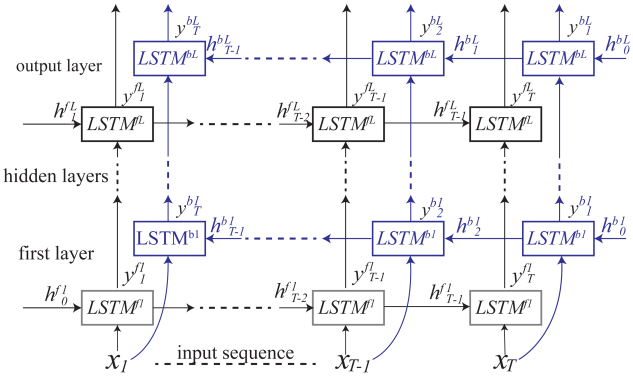


FIGURE 7. Bidirectional LSTM-based DRNN model with forward $LSTM^{fl}$ and backward $LSTM^{bl}$ tracks.

in transforming raw data into a more abstract representation, as well as for learning spatial dependencies [15]. The input is the MFS, which is a discrete sequence of samples $(x_1, x_2, \ldots, x_T)$ that are passed into the first layer at time t $(t = 1, 2, \ldots, T)$.

Initially, the internal state I_0^l and the hidden state h_0^l are both set to zero. 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:

$$y_t^1, h_t^1, I_t^1 = LSTM^1\left(I_{t-1}^1, h_{t-1}^1, x_t; \theta^1\right). \tag{12}$$

Any layer l in the upper layers uses the lower layers y_t^{l-1} as its input. If θ^l represents the parameter (b, U, W) of the LSTM cells for layer l, then (12) can be written as:

$$y_t^l, h_t^l, I_t^l = LSTM^l(I_{t-1}^l, h_{t-1}^l, y_t^{l-1}; \theta^l). \tag{13}$$

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.

2) BIDIRECTIONAL LSTM-BASED DRNN MODEL

This architecture uses a bidirectional LSTM-based DRNN, as shown in Fig. 7. It includes two parallel LSTM tracks: forward and backward loops to exploit context from past and future to predict its label [19], [34]. In the first layer, the forward track ($LSTM^{f1}$) and backward track ($LSTM^{b1}$) read

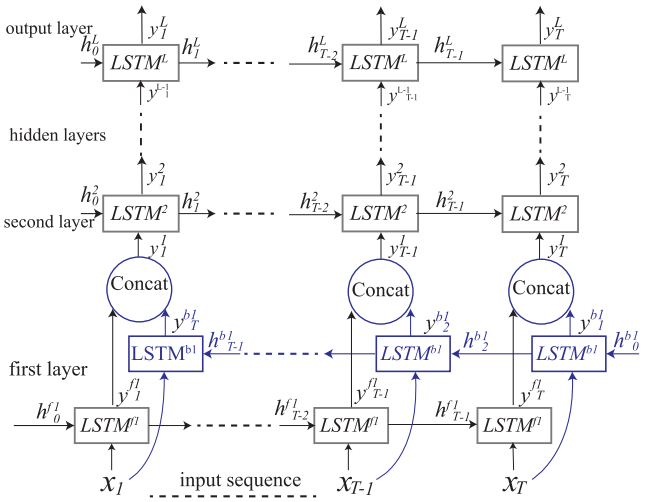


FIGURE 8. Cascaded unidirectional and bidirectional LSTM-based DRNN model. The upper unidirectional layer is concatenated with the bidirectional first layer.

input window T from left-to-right and right-to-left, respectively:

$$y_t^{f1}, h_t^{f1}, I_t^{f1} = LSTM^{f1}(I_{t-1}^{f1}, h_{t-1}^{f1}, x_t; W^{f1}) \tag{14}$$

$$y_t^{b1}, h_t^{b1}, I_t^{b1} = LSTM^{b1}(I_{t-1}^{b1}, h_{t-1}^{b1}, x_t; W^{b1}). \tag{15}$$

At each time step, the top layer L outputs a sequence of scores from both forward LSTM $(y_1^{fL}, y_2^{fL}, \ldots, y_T^{fL})$ and backward LSTM $(y_1^{bL}, \ldots y_2^{bL}, \ldots, y_T^{bL})$. The combined scores $Y \in R^k$ represent landmark label prediction for the window segment T. The late-fusion is the resulting outputs from both forward and backward tracks, which are combined as follows:

$$Y = \frac{1}{T}\sum_{t=1}^{T}\left(y_t^{fL} + y_t^{bL}\right). \tag{16}$$

3) CASCADED BIDIRECTIONAL AND UNIDIRECTIONAL LSTM-BASED DRNN MODEL

The model architecture, shown in Fig. 8, is inspired from [19] and [32]. In this architecture, the first layer is designed with a bidirectional RNN, whereas the upper layers are unidirectional. The first layer has a forward LSTM track $LSTM^{f1}$ generating an output $(y_1^{fL}, y_2^{fL}, \ldots, y_T^{fL})$ and a backward track $LSTM^{b1}$ generating an output$(y_1^{bL}, y_2^{bL}, \ldots, y_T^{bL})$. The two types of outputs are combined and fed into the second unidirectional layer to form a new output $(y_1^1, y_2^1, \ldots, y_T^1)$:

$$y_t^1 = y_t^{f1} + y_{T-t+1}^{b1}. \tag{17}$$

The operation of the upper layers is same as in the unidirectional model described earlier.

VI. EXPERIMENTAL PRELIMINARIES

A. EXPERIMENT SETUP

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 100 m × 2.5 m and 7 m × 7 m, respectively. The corridor contains magnetic elements like

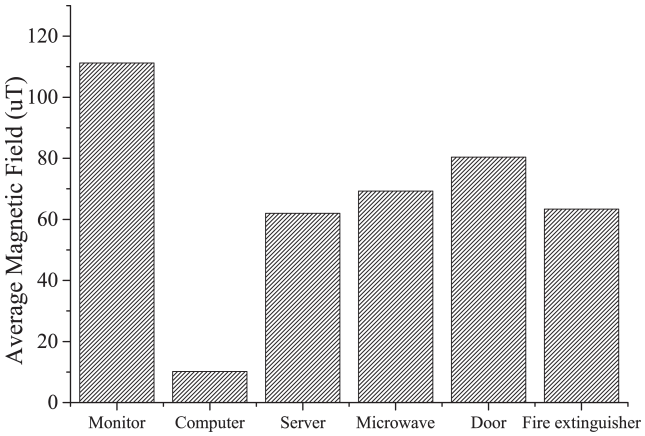


FIGURE 9. Magnetic field strength variation measured near various external factors.

iron doors, bearing columns, and other building materials, which create fluctuations 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 fluctuations. We studied the effect of these external factors in MFS fluctuation by collecting a sample of MFS using a smartphone. We found that the factors such as computers, microwaves, doors and other equipment significantly distorts magnetic field near them as shown in Fig. 9. This can be due to the presence of a magnetic element or an electrical circuit inside them. A pictorial representation of the test environment is shown in Fig. 10.

B. EXPERIMENT METHODOLOGY

The proposed model was evaluated through experiments. The layouts of the testbeds are shown in Figs. 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 the 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 Fig. 11 (d). While recording the data in the corridor, the orientation of the phone was held by a walking user, as shown in Fig. 11 (a) to avoid errors due to soft iron distortion.

However, in the lab, due to limited space, we used a movable stand, as shown in Fig. 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 Fig. 11 (c). Finally, our dataset consisted of 25 landmark positions in the corridor and 17 landmark positions inside the lab. The size of the training data in the corridor was about 71,300 MFS samples for the 25 landmarks, and in the lab, it was about 19,500 MFS samples for 17 landmarks.

C. EXPERIMENT EVALUATION

We used Google TensorFlow as a deep learning framework as it allowed us to design a more detailed neural network

TABLE 1. Server system configuration and framework for deep learning network.

CATEGORY	TOOLS
CPU	Intel i5-7600 @ 3.50 GHz
GPU	NVIDIA 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
	NVIDIA cuDNN v7.0

model. Also, NVIDIA cuDNN and CUDA Toolkit, which provide parallel processing, were used to drastically improve our training performance [33].

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

1) NETWORK TRAINING AND TESTING

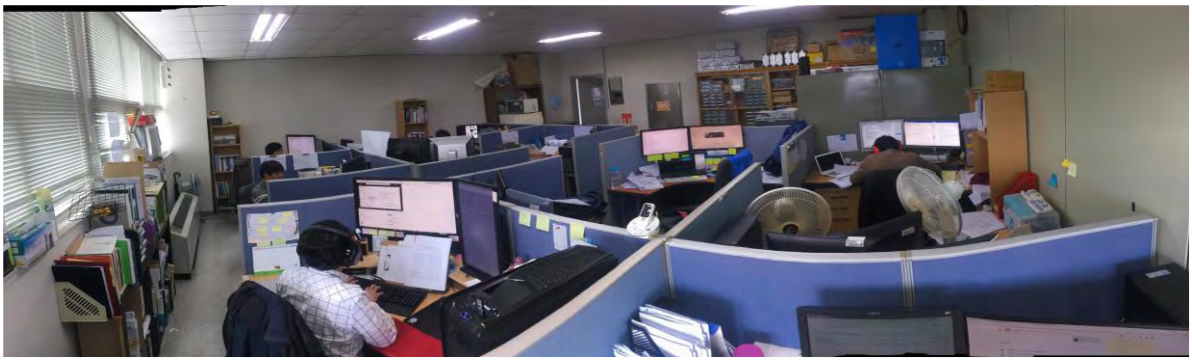
We trained our DRNN model with the preprocessed geomagnetic data stored in a fingerprinting database or dataset. The dataset was divided such that 80% of the data is used for training and the remaining 20% is for the testing process. The hyperparameters, such as the number of hidden nodes, mini-batch 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 ground truth 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 provided as a one-hot vector $O \in R^k$ with a value o_c associated with each landmark label c.

The predicted classes $\hat{O} \in R^k$ include the probability of every class ρ_c generated by our model:

$$\mathcal{L}\left(O, \hat{O}\right) = -\sum_{c=1}^{k} o_c \log \rho_c. \tag{18}$$

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 [34]. The weight and other parameters were optimized with forward and backpropagation. In our work, we provided parameters for forward propagation and Google TensorFlow calculated all required back propagation steps. We used a dropout technique as regularization to avoid overfitting in our model [35]. 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 segmented

(a)



(b)

FIGURE 10. Experiment environments. (a) First testbed in the corridor. (b) Second testbed in the lab.

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.

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. Fig. 12 shows the accuracy and cost of the training and testing processes for the unidirectional DRNN model on our testbed. The training and testing accuracies increase with training epoch as the model generalizes to new data. Similarly, the training and testing costs decrease with each epoch as the model learns the data and reaches an optimal value. In addition, the testing accuracy and cost follow the training accuracy and cost graphs closely, which indicates the effectiveness of the dropout technique in the model for avoiding overfitting.

We divided our dataset into a mini-batch for efficient memory usage and to prevent 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 that was found to be best for our testbeds is listed in Table 3.

TABLE 2. Summary of DRNN input data to evaluate the proposed deep learning models. Training window length denotes the number of samples in a window that we found to yield the best results for each testbed. Each dataset was divided into 80% for training and 20% for testing.

TESTBED	NO. OF LANDMARKS	TRAINING WINDOW LENGTH	TRAINING SAMPLES COUNT	TESTING SAMPLES COUNT
Corridor	25	16	56982	14246
Lab	17	4	15636	3909

TABLE 3. Summary of hyperparameters used in the two different testbeds.

PARAMETERS	NO. OF CLASSES	TRAINING WINDOW LENGTH
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 Classes	25	17

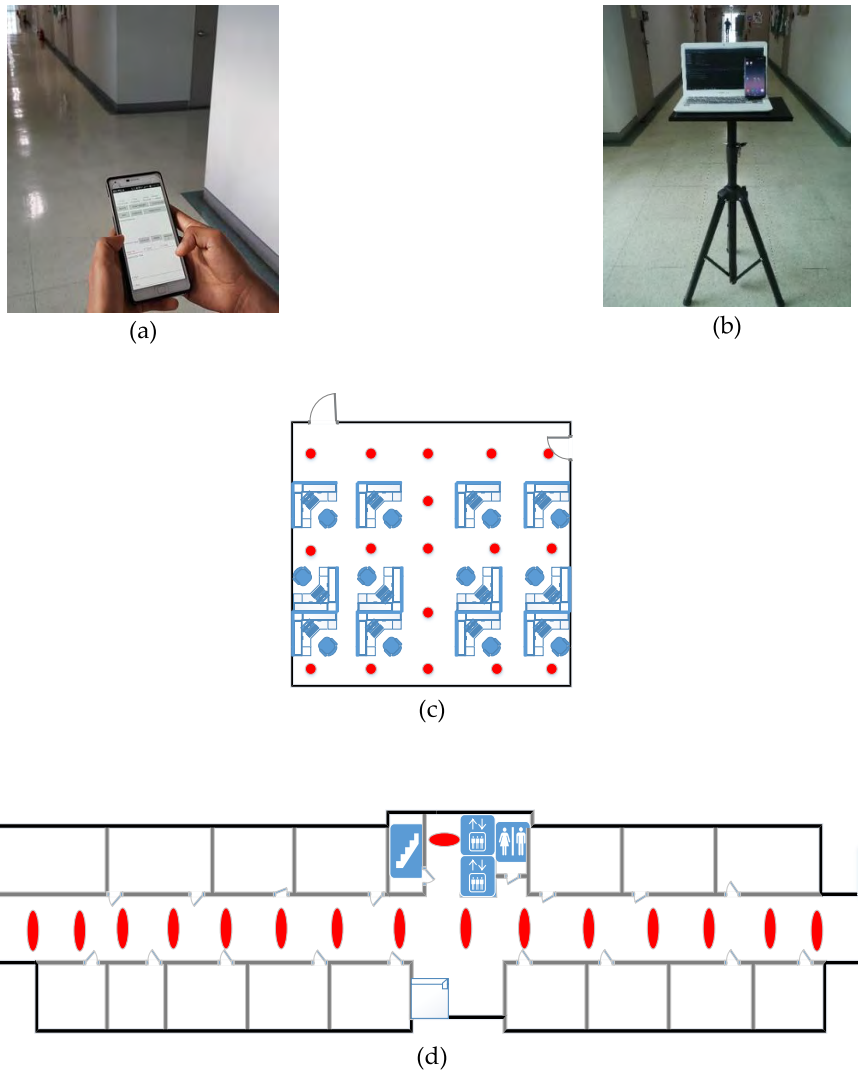


FIGURE 11. Data gathering methods. (a) Orientation of the smartphone while collecting data in the corridor at each grid. (b) Stand used in the lab to gather data at each landmark. (c) Layout of the lab containing working desks surrounded with landmarks denoted by the red dots, where the data are gathered. (d) Layout of the corridor with the landmarks denoted by red eclipse, where the data are gathered.

The trained DRNN models were evaluated with the test dataset. We found that the testing accuracy was greatly affected by the 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 increased. However, if we increased the number of hidden layers, the performance of the model was not necessarily good. This could be due to the difficulty in gradient propagation when we increase the number of layers. Fig. 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.

2) PERFORMANCE METRICS

The performance of the proposed model was verified using the following evaluation metrics [36]:

1. Precision: measuring the number of true samples out of those classified as positive. The overall precision was calculated by averaging the precision of each class:

$$Per-class\ Precision_c = \frac{t_{p_c}}{t_{p_c} + f_{p_c}} \tag{19}$$

$$Overall\ Precision = \frac{1}{k}\left(\sum_{c=1}^{k} \frac{t_{p_c}}{t_{p_c} + f_{p_c}}\right), \tag{20}$$

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.

2. Recall (Sensitivity): measuring the number of samples that are correctly classified out of the total samples in a class. The overall recall is the average of the recalls

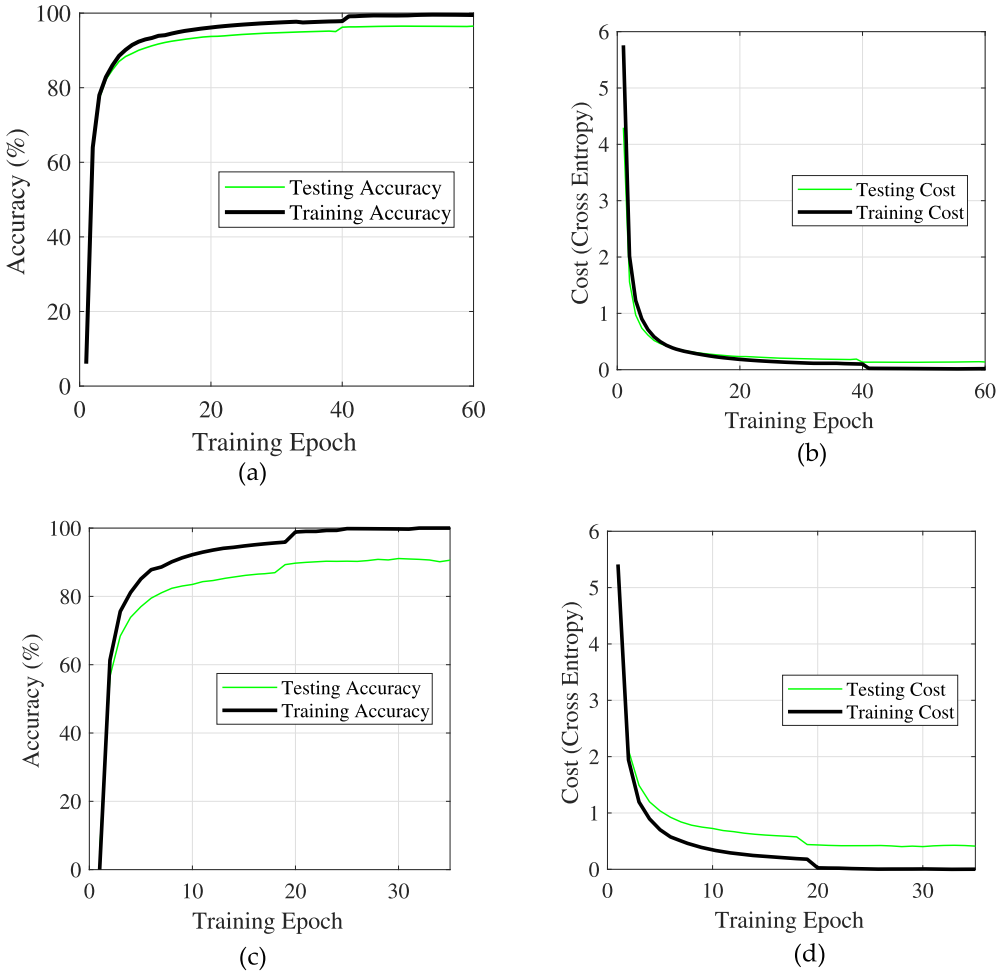


FIGURE 12. The accuracy and cost of our DRNN model for the testbed dataset over mini-batch training iterations. (a) Training and testing accuracies in the lab. (b) Cross-entropy cost between the ground truth labels and predicted labels for both training and testing in the lab. (c) Training and testing accuracy in the corridor. (d) Cross-entropy cost in the corridor.

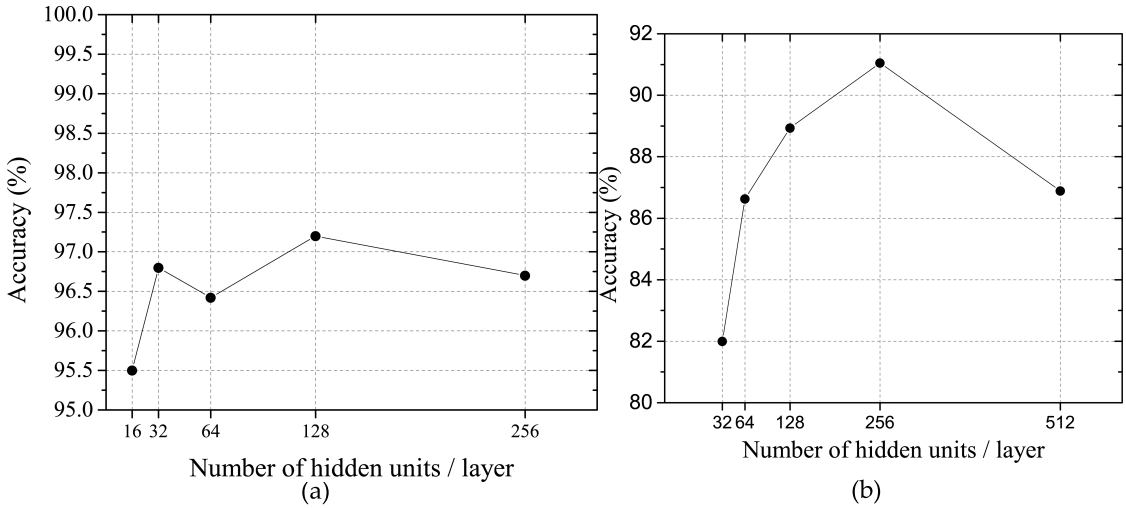


FIGURE 13. Accuracy measurements with increasing number of hidden units per layer. (a) Test accuracy for a different number of hidden units per layer in the lab. (b) Test accuracy for a different number of hidden units per layer in the corridor.

for each class:

$$Per-class\ Recall_c = \frac{t_{p_c}}{t_{p_c} + f_{n_c}} \tag{21}$$

$$Overall\ Recall = \frac{1}{k}\left(\sum_{c=1}^{k} \frac{t_{p_c}}{t_{p_c} + f_{n_c}}\right), \tag{22}$$

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

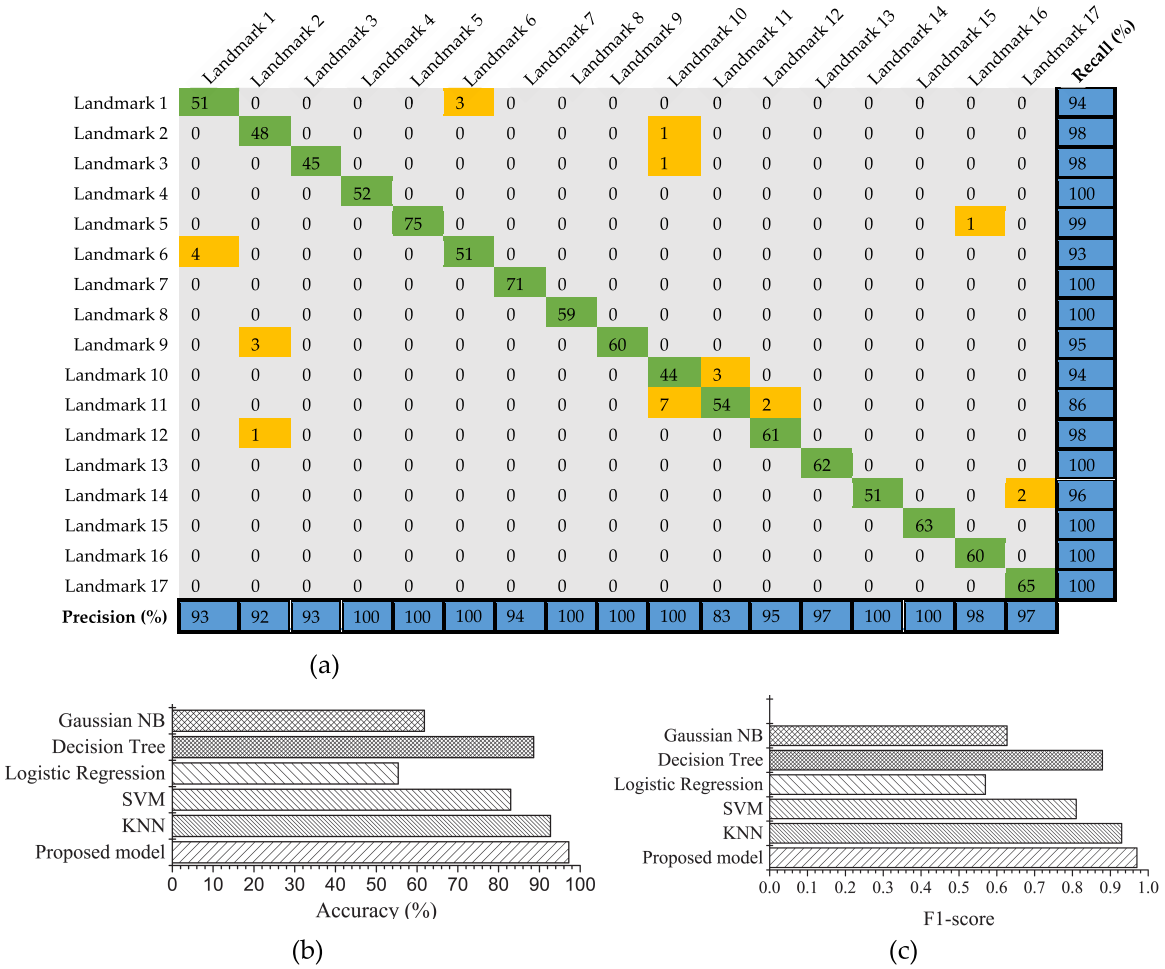


FIGURE 14. Performance results for the proposed bidirectional DRNN model in the lab. (a) Confusion matrix for the landmark classification in the lab with per-class precision and recall. (b) Accuracy comparison of the proposed model with other methods. (c) F1-score comparison of the proposed model with other methods.

3. Accuracy: measuring the proportion of correctly predicted classes overall predictions:

$$Overall\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (23)$$

where $TP = \sum_{c=1}^{k} t_{p_c}$ is the overall true positive for a classifier of 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.

4. F1-score: the weighted harmonic mean of precision and recall:

$$F1Score = \sum_{c=1}^{k} \left(2 \left(\frac{n_c}{N} \right) \frac{precision_c * recall_c}{precision_c + recall_c} \right), \quad (24)$$

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 samples in a set of k classes. The F1-score provides a measure of a test's accuracy.

VII. EXPERIMENTAL RESULTS AND ANALYSIS

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 Fig. 14 (a) gives an overview of the classification results for the proposed model in the test set, along with the per-class precision and recall results. Figs. 14 (b) and (c) show a performance comparison of the proposed system with other machine learning methods, such as k-nearest neighbor (KNN) [37], 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 yield the best performance results in term of per-class precision and recall, as shown by the confusion matrix in Fig. 15 (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

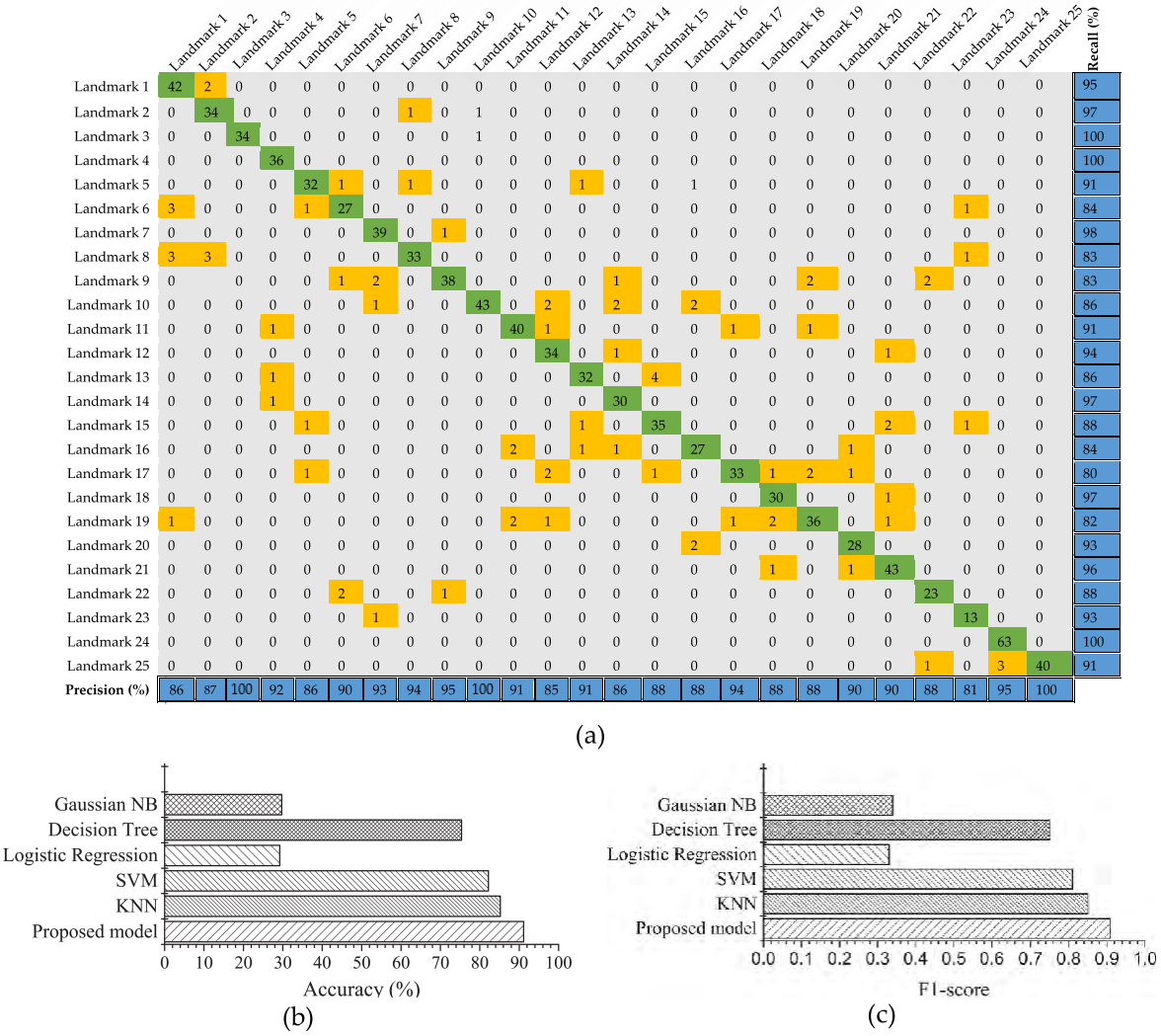


FIGURE 15. Performance results of the proposed unidirectional DRNN model in the corridor. (a) Confusion matrix for the test in the corridor along with per-class precision and recall. (b) Accuracy comparison of the proposed model with other methods. (c) F1-score comparison of the proposed model with other methods.

and these algorithms can be seen in Figs. 15 (b) and (c), respectively.

The performance results of the proposed models clearly show 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. Also, we studied the MFS at different landmarks from two testbed to find the overall changes brought by the aforementioned external factors. We found that the difference in fluctuations of MFS observed at different landmark positions inside the lab is more significant compared to the corridor as shown in Fig. 16. It allows our model to extract the reliable features of the magnetic pattern inside the lab. Hence, the performance of our model is found to be better in lab. Since the range of fluctuation in MFS around landmarks in lab is greater, we have included more landmarks in the lab in terms of space density. It proves that our models can be effective for a broad range of landmark classifications in various indoor environments. Table 4 contains a performance summary of our models in the two test environments.

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. To show the efficiency of our model with more amount of data samples, we trained the model with a total of 42 landmarks, obtained from combining the data samples from both testbeds. The result in Fig. 17 shows that our model outperform other methods drastically when the amount of samples and number of landmark increases. Including more layers in the DRNN helped the model to extract the discriminative features. These features were exploited for effective learning and distinguishing more complex patterns formed by MFS at landmarks.

TABLE 4. performance summary of our model with the datasets on the 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

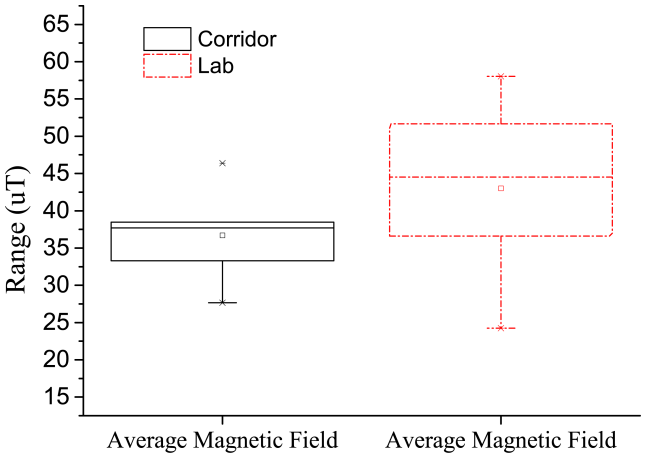


FIGURE 16. Distribution of MFS fluctuation measured around different landmarks in two testbeds.

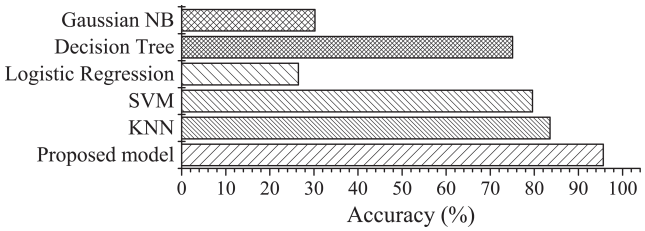


FIGURE 17. Performance result of the proposed model in a combined dataset from the lab and the corridor.

In addition, using DRNNs to capture sequential and temporal dependencies provided a significant improvement in performance.

VIII. CONCLUSION

This paper presented three novel LSTM-based DRNN architectures for indoor landmark classification using MFS. Moreover, we first verified experimentally the feasibility of using MFS for landmark classification and empirically evaluated our models using experiments with datasets on two testbeds. Although the training phase was computationally demanding, the test phase was fast and suitable for real-time indoor landmark classification. Experimental results showed 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 at various landmark positions. Furthermore, by exploiting the functionality of DRNNs, our models were able to capture temporal dependencies between input magnetic field data.

REFERENCES

[1] B. P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, 2nd ed. Lincoln, MA, USA: 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, Jan. 2014.
[3] M. X. Gong, B. Hart, and S. Mao, "Advanced wireless LAN technologies: IEEE 802.11ac and beyond," *GetMobile, Mobile Comput. Commun.*, 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. C, Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.
[6] P. B. and V. N. Padmanabhan, "RADAR: An in-building RF based user location and tracking system," in *Proc. IEEE Conf. Comput. Commun. 90th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, vol. 2, Mar. 2000, pp. 775–784.
[7] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. 3rd Int. Conf. Mobile Syst., Appl., Services*, 2005, pp. 205–218.
[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, Jan. 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, pp. 3226–3228, Sep. 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, Sep. 1989.
[12] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman, "Indoor location sensing using geo-magnetism," in *Proc. 9th Int. Conf. Mobile Syst., Appl., Services*, 2011, pp. 141–154.
[13] W. Storms, J. Shockley, and J. Raquet, "Magnetic field navigation in an indoor environment," in *Proc. Ubiquitous Position Indoor Navigat. Location Service (UPINLBS)*, Oct. 2010, pp. 1–4.
[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, Dec. 2016.
[15] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, no. 6, May 2013, pp. 6645–6649.
[16] M. Sundermeyer, R. Schl, and H. Ney, "LSTM neural networks for language modeling," in *Proc. Interspeech*, 2012, pp. 194–197.
[17] L. Yao *et al.*, "Describing videos by exploiting temporal structure," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4507–4515.
[18] 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, Jan. 2014.

[19] A. Murad and J.-Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors*, vol. 17, no. 11, p. 2556, 2017.
[20] Z. Yang, P. Zhang, and L. Chen, "RFID-enabled indoor positioning method for a real-time manufacturing execution system using OS-ELM," *Neurocomputing*, vol. 174, pp. 121–133, Jan. 2016.
[21] 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," in *Proc. IEEE 15th Int. Conf. Adv. Robot. New Boundaries Robot. (ICAR)*, Jun. 2011, pp. 198–203.
[22] J. Jung, T. Oh, and H. Myung, "Magnetic field constraints and sequence-based matching for indoor pose graph SLAM," *Robot. Auton. Syst.*, vol. 70, pp. 92–105, Aug. 2015.
[23] J. Haverinen and A. Kemppainen, "Global indoor self-localization based on the ambient magnetic field," *Robot. Auto. Syst.*, vol. 57, no. 10, pp. 1028–1035, Oct. 2009.
[24] D. Navarro and G. Benet, "Magnetic map building for mobile robot localization purpose," in *Proc. IEEE Conf. Emerg. Technol. Fact. Autom. (ETFA)*, Sep. 2009, pp. 4–7.
[25] K. P. Subbu, B. Gozick, and R. Dantu, "LocateMe: Magnetic-fields-based indoor localization using smartphones," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, pp. 1–27, 2013.
[26] 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, Dec. 2011.
[27] H. Yu, Z.-H. Tan, Z. Ma, R. Martin, and J. Guo, "Spoofing detection in automatic speaker verification systems using DNN classifiers and dynamic acoustic features," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4633–4644, Oct. 2018.
[28] Z. Ma, H. Yu, W. Chen, and J. Guo, "Short utterance based speech language identification in intelligent vehicles with time-scale modifications and deep bottleneck features," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 121–128, Jan. 2019.
[29] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
[30] M. Myllymäki, T. Mrkvička, P. Grabarnik, H. Seijo, and U. Hahn, "Global envelope tests for spatial processes," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 79, no. 2, pp. 381–404, 2017.
[31] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Neural Networks*, Kremer and Kolen, Eds. Piscataway, NJ, USA: IEEE Press, 2001.
[32] Y. Wu *et al.*. (2016). "Google's neural machine translation system: Bridging the gap between human and machine translation." [Online]. Available: https://arxiv.org/abs/1609.08144
[33] M. Abadi *et al.* (2016). "TensorFlow: Large-scale machine learning on heterogeneous distributed systems." [Online]. Available: https://arxiv.org/abs/1603.04467
[34] I. Goodfellow, Y. Bengio, and A. Courville, "Book review: Deep learning," *Deep Learn.*, vol. 22, no. 4, pp. 351–354, 2016.
[35] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *Proc. Int. Conf. Front. Handwriting Recognit. (ICFHR)*, Sep. 2014, pp. 285–290.
[36] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.
[37] M. Yao and B. Vocational, "Research on learning evidence improvement for KNN based classification algorithm," *Int. J. Database Theory Appl.*, vol. 7, no. 1, pp. 103–110, 2014.

BIMAL BHATTARAI received the B.Tech. degree in electronics and communication engineering from JNTUA, India, in 2015, and the M.Sc. degree from the Department of Information and Communication Engineering, Chosun University, South Korea. He was a BSS Engineer with ZTE Corporation, Nepal. His research interests include interdisciplinary research, including indoor positioning and navigation, signal processing, machine learning, and deep learning applications.

ROHAN KUMAR YADAV received the B.E. degree in electronics and communication from the Sona College of Technology, Anna University, India, in 2015, and the master's degree in information and communication engineering from Chosun University, South Korea. He was a NOC Engineer with Huawei Technologies, Nepal, from 2015 to 2017. His research interests include indoor positioning and navigation, signal processing, machine learning, deep learning, and its applications.

HUI-SEON GANG received the B.S. and M.S. degrees in information and communication engineering from Chosun University, South Korea, in 2012 and 2014, respectively, where he is currently pursuing the Ph.D. degree. His research area includes video coding technique, error resilience and concealment, multimedia communication, wireless networks, signal processing, and indoor positioning techniques.

JAE-YOUNG PYUN received the B.S. degree from Chosun University, South Korea, in 1997, the M.S. degree from Chonnam University, in 1999, and the Ph.D. degree from Korea University, South Korea, in 2003. From 2001 to 2003, he was a Research Engineer with Dalitech Corporation. He has also served as a Research Engineer with Samsung Electronics for research and development of mobile phone communication system, from 2003 to 2004. He is currently a Professor with the Department of Information and Communication Engineering, Chosun University. His research interests include indoor localization, the IoT-based services, mobile QoS, IP QoS, wireless sensor networks, and video communication. He was selected for 2000 Outstanding Intellectual of 21st Century Award by the International Biographical Centre, and Great Minds of 21st Century by American Biographical Institute, in 2008.