Recurrent Neural Network-Based Hybrid Localization for Worker Tracking in an Offshore Environment

Gunwoo Lee

Department of Computer Science, Korea Advanced Institute of Science and Technology (KAIST),
Daejeon 34141, Korea; gwlee@kaist.ac.kr; Tel.: +82-042-350-7762

Received: 1 June 2020; Accepted: 8 July 2020; Published: 9 July 2020

Abstract: Accidents involving marine crew members and passengers are still an issue that must be studied and obviated. Preventing such accidents at sea can improve the quality of life on board by ensuring a safe ship environment. This paper proposes a hybrid indoor positioning method, an approach which is becoming common on land, to enhance maritime safety. Specifically, a recurrent neural network (RNN)-based hybrid localization system (RHLS) that provides accurate and efficient user-tracking results is proposed. RHLS performs hybrid positioning by receiving wireless signals, such as Wi-Fi and Bluetooth, as well as inertial measurement unit data from smartphones. It utilizes the RNN to solve the problem of tracking accuracy reduction that may occur when using data collected from various sensors at various times. The results of experiments conducted in an offshore environment confirm that RHLS provides accurate and efficient tracking results. The scalability of RHLS provides managers with more intuitive monitoring of assets and crews, and, by providing information such as the location of safety equipment to the crew, it promotes welfare and safety.

Keywords: indoor localization; recurrent neural network; hybrid positioning

1. Introduction

Despite hundreds of ship accidents annually over the past decade [1], a system for disaster response and accident prevention for crew and passengers is lacking. In offshore plant structures, a main ship and a support ship several meters apart are connected by gangways, and there is always the risk of explosion and flame. To cope with this danger via an emergency alarm, a public address/general alarm system and a wireless terminal are used. Nevertheless, if the location of the accident can be identified using a smartphone application, it could be dealt with more efficiently. Workers are always working in areas exposed to danger; thus, if an accident or breakdown is found, workers must be notified promptly. In addition, passengers unfamiliar with the structure of the ship should be provided with evacuation routes and lifeboat location information quickly and accurately. Recently, with the popularization of smartphones, information technology (IT)-based wireless network systems are becoming more common in commercial and passenger ships. Based on this wireless infrastructure, a technology capable of enhancing safety in an emergency using the smartphones of crew members, engineers, and passengers is required.

Global navigation satellite system technology can be used outdoors to locate users; however, indoors, this signal can be blocked and difficult to use [2]. To overcome this limitation, many studies for indoor localization were conducted in the past few years. A representative technique for indoor localization is to use a Wi-Fi signal, which uses triangulation between the wireless signals of users and access points (APs), or a signal fingerprinting method, using the strength of wireless signals obtained at specific coordinates [3,4]. Other methods include beacons using Bluetooth or infrared [5,6], tag recognition using radio frequency identification (RFID) [7,8], and quick response (QR) code recognition using a camera [9]. These techniques can derive absolute position coordinates with

relatively high accuracy. However, because the signal scan time of the device is relatively slow or the coordinates can be derived only at the location where the tag is installed, it is difficult to proceed with continuous positioning. In addition, there is a limitation in that positioning accuracy is difficult to improve beyond a certain value. Recently, these difficulties were solved using various sensors installed on smartphones. Specifically, the absolute position coordinates are derived from the above techniques, and the continuous position between these coordinates uses data from inertial measurement unit (IMU) sensors [10,11]. The simplest approach is to use a loosely coupled technique such as a Kalman filter [12] to weight each sensor's location and average it. However, this method has difficulty in evaluating the weight of each sensor position having different criteria, and there is a difficulty in integrating when a new sensor is additionally used for positioning.

On the other hand, probability-based positioning methods using particle filters [13] or hidden Markov models (HMMs) [14] can easily integrate new sensors without considering the weight of each sensor. However, it is necessary to analyze the signal distribution for each space or the probability distribution for sensor accuracy in advance, and this incurs a high cost in terms of the time required to calculate positions. When indoor positioning is performed in a non-line-of-sight area where obstacles exist, the stability of data reception is poor because of the reflection and diffraction of radio waves. The cumulative error for localization increases accordingly. In particular, the internal structure of ships and offshore plants has a narrower structure and more steel than the corridors of ordinary buildings, and electromagnetic interference caused by generators and engine rooms results in serious signal distortions, thereby reducing the accuracy of location recognition. Therefore, it is desirable to perform more accurate positioning by fusing data from various sensors, rather than performing absolute positioning using only a specific signal. In hybrid positioning, user tracking is performed by pre-learning data for deriving the absolute position and applying IMU sensor data to the positioning algorithm in real time. If the learning data are densely collected, it is possible to improve the accuracy by analyzing the signal distribution in advance at a specific location, but there may be cases in which the IMU sensor data cannot be used, because the amount of computation increases rapidly during positioning. In addition, it is difficult to improve accuracy if the different data collection cycles for each sensor are not properly synchronized.

To solve this problem, this paper proposes a method incorporating a localization engine that uses an artificial neural network (ANN), which creates a learning model when learning data are given, before classifying the new data. In the proposed method, the ANN is trained by receiving radio signals and IMU sensor values as inputs to the single localization model, consisting of a recurrent neural network (RNN). Unlike existing methods, the measurement is conducted without maintaining the learning data in a database; hence, the measurement calculation time and the cost for maintaining the database are reduced. In many studies, the various variables used for measurement are determined either empirically or artificially, either by the researcher or by analyzing large amounts of input data. In the proposed method, the variables derived from ANN training automatically can have the same effect. An experiment was conducted on an actual offshore plant to verify the reduction in localization time and the improvement in tracking accuracy. The proposed method achieved a 29% improvement on a deck, compared to a previously proposed method [15] that uses the extended Viterbi algorithm. With faster and more accurate location information, these results can be applied to communicate safety information more quickly to the people who need it in shipboard and offshore environments.

The remainder of this paper is organized as follows: Section 2 introduces related research on neural network-based positioning methods. Section 3 outlines the data collection method and describes the RNN-based hybrid localization system (RHLS). Section 4 presents and analyzes evaluation results obtained for the proposed system. Finally, Section 5 concludes this paper.

2. Related Work

Research using neural networks is conducted in various fields, and it was recently applied to the field of indoor positioning. Although fewer studies were conducted using deep learning than

the existing indoor positioning techniques, it can be seen from the results of research using neural networks that there are still many applications in the indoor positioning field. In Reference [16], instead of locating a mobile user's position one at a time as in the case of conventional methods, their RNN solution aims at trajectory positioning. Moreover, the proposed method considers the correlation among the received signal strength indicator (RSSI) measurements in a trajectory. However, as the data are collected using a robot, there may be a slight difference in the results obtained using data collected by a dedicated user.

Gan et al. [17] developed an algorithm based on deep belief networks, and they conducted evaluations using a combination of Wi-Fi and Bluetooth signals. Although they achieved a mean accuracy of 0.52 m, the time and battery consumption increased owing to the use of both signals. Belay et al. [18] filled in missing RSS values using regression, and then applied linear discriminant analysis to reduce features. Before applying a deep neural network (DNN) for localizing Wi-Fi users, they appended five basic service set identifications (BSSIDs) having the strongest RSS values with a reduced RSS vector. Xiao et al. [19] used a deep learning architecture for regression and a support vector machine for classification to output the estimated location directly from the measured fingerprint.

Xiao et al. [20] proposed a Bluetooth low energy (BLE) localization system using a denoising autoencoder to build a fingerprint database in three-dimensional (3D) space. However, if the target area becomes very wide, such as a shopping mall, many BLE devices must be installed; thus, positioning using only BLE is limited in practice. Wang et al. [21] proposed a deep learning scheme based on channel state information to obtain more fine-grained information on the wireless channel than RSS-based methods, such as the amplitude and phase of each subcarrier from each antenna. They used deep learning to train all the weights of a deep network as fingerprints in the offline training phase. In the online localization, they used a stochastic method based on the radial basis function to obtain the localization result. However, the experiments in their study were conducted in a very small space.

Several researchers also conducted indoor localization using convolutional neural networks (CNNs). Li et al. [22] focused on the pose regression problem, and they introduced a deep neural network architecture for RGB-Depth images and a training method for dual-stream CNNs. They discussed different depth image encoding methods and proposed a novel encoding method for indoor relocation. Liu et al. [23] presented a localization method that uses a hybrid wireless fingerprint based on a CNN. The proposed fingerprint method combines the ratio fingerprint and the RSSI to enhance the expression of indoor environment characteristics. Zhang et al. [24] set the fingerprinting dataset as several images. They used a CNN to extract reliable features from the images and then built internal representations between images and the locations of reference points based on the PyTorch computational framework. Some researchers also used ANNs [25] and multilayer perceptron networks [26] for Wi-Fi-based positioning inside buildings. Jang et al. [27] implemented an RNN for indoor positioning. However, their proposed system does not use Wi-Fi signals; it uses the magnetometer only.

The studies examined so far attempted to solve the problems with traditional approaches, such as by reducing the execution time that can occur from large data sizes, removing manual parameter tuning, and reducing positioning inaccuracies resulting from signal fluctuations when using a neural network. In all of these studies, the research was conducted by focusing on positioning accuracy rather than execution time. Nevertheless, the test for localization accuracy was omitted, or it was difficult to evaluate the performance of the system. Furthermore, these studies may not be suitable for practical environments using a small dataset, or it may be difficult to perform continuous positioning because additional sensor data are not used. Above all, these studies are focused only on typical indoor spaces; therefore, it is difficult to verify their performance in offshore environments.

3. Materials and Methods

Existing localization methods, which were introduced in Section 1, typically use a database that stores the characteristics of the target site, such as radio maps, and use a lazy-learning method that determines the location by comparing the database with the signal values scanned in real time. To use all the data given in real time, the localization result of each sensor should be completed before the localization result of the fastest sensor is estimated. However, the localization results of each sensor may exceed this time, as shown in Figure 1. In other words, because all the received data cannot be used, there may be a decrease in accuracy. The method proposed in this study uses an eager learning technique that determines the location only by inference of the input value. RHLS is a method that can reduce the positioning computation time caused by large-scale learning data and can solve the accuracy degradation caused by mismatching the positioning cycles of multiple sensors.

Figure 1. Positioning time of each sensor over time (geomagnetic data have an approximately 100-ms positioning period, with other sensors exceeding this speed).

3.1. Structure of RHLS

Figure 2 shows the structure of the RHLS method. When user tracking is in progress, the user's smartphone periodically scans the Wi-Fi signal through the built-in module. At the same time, geomagnetic and acceleration sensors also periodically collect data. RHLS uses pedestrian dead reckoning (PDR) for continuous positioning, which provides the values of geomagnetic, acceleration, and gyroscope sensors to the PDR module to calculate step detection, stride length, and direction of movement, respectively. The calculated values are provided to the recurrent network-based positioning module along with Wi-Fi signal data and geomagnetic sensor values. The positioning module calculates the positioning result based on the received sensor values. The positioning server in Figure 2 is located offshore. Owing to the nature of a ship, unlike on land, it is difficult to provide an internet environment continuously. Therefore, communication between modules is performed using the onboard network of the ship. However, the location server is not necessarily separated from the user device. That is, it may be embedded in the user's device depending on the implementation.

Figure 2. Structure of the proposed method.

RHLS provides geomagnetic, accelerometer, and gyroscope data to the positioning module, but sensors can also be used or removed depending on the implementer's choice. The wireless signal is not limited to Wi-Fi, and a wireless signal capable of deriving an absolute position, such as Bluetooth, can be added. On the other hand, in RHLS, the module that scans the radio signal and the module that reads sensor data operate independently of each other.

When the Wi-Fi signal is scanned or the PDR module detects a step, RHLS immediately passes the collected geomagnetic values to the recurrent network positioning module. Geomagnetic data appear as values along three axes, x , y , and z , according to rules set inside the device. Most smartphone geomagnetic modules are designed to have this structure. The three-axis value measured by the geomagnetic sensor indicates a vector relative to magnetic north. When the pose of the smartphone changes, the value of each axis changes. In other words, if the three axes continuously change when performing absolute positioning, the ambiguity becomes worse. Therefore, the proposed method includes a process of converting the magnetic field vector into a pose-independent value based on the theory below.

There are three types of values that can be calculated from magnetic field vectors: magnetic intensity, magnetic inclination, and magnetic declination. In Figure 3, the only axis clearly shown is the one that points to the ceiling from the center of Earth, and the surface is perpendicular to this axis. The magnitude of the magnetic field strength is calculated using the Euclidean norm of the magnet vector. That is, the strength of the magnetic field is constant depending on the location, regardless of the pose. The degree to which the magnetic field vector sinks toward the surface is called the magnetic inclination. To find this value, the angle between the gravity vector and the magnetic field vector is calculated, before subtracting 90° . This value is also not affected by pose. The gravity vector can be calculated by any smartphone with a built-in accelerometer, and it can be determined relatively accurately using the built-in gyroscope. Finally, the magnetic declination means the angle between the vector generated when the magnetic field vector is projected onto the surface and the vector pointing true north. However, because there is no way to know true north using an internal sensor, this value cannot be used. Therefore, in RHLS, magnet vectors, which can use vector values regardless of poses and magnetic inclination values, are used for positioning.

Figure 3. Parameters of the magnetic field.

The RNN positioning module is initialized when Wi-Fi scanning is completed on the smartphone. At this time, the strength of the scanned AP and the signals received from the AP are arranged in a predetermined order. The vector, the value measured by the geomagnetic sensor, and the moving distance and direction calculated by the PDR module are input to the recurrent network positioning module. When input is made to the positioning module, the resulting value in metric coordinates is output, as explained in Section 3.4.3. If a step is recognized in the PDR module before the next Wi-Fi scan is performed, the process of passing the geomagnetic and PDR module values as inputs to the positioning module and deriving the coordinates is repeated.

3.2. Learning Data Collection

RHLS checks the path to collect through the walking survey method, which collects signals during walking at the target site, and then walks along the path to store the Wi-Fi fingerprint and inertial sensor (INS) data along with the collection time. Step detection is a method of estimating when a pedestrian stepped using the accelerometer data collected from a device. Estimating the location of the fingerprint through step detection is based on the assumption that the step stride of the pedestrian is constant. By calculating the length of the path to be collected and the total number of steps generated during collecting, the moving distance per unit step can be determined. By comparing the timestamp of the collected fingerprint with the timestamp of the steps that occurred during walking, it is possible to ascertain how many steps there were up to an arbitrary fingerprint. Then, by multiplying the corresponding value and the moving distance per unit step, it is possible to determine the length from the starting point of the path until the corresponding fingerprint is collected. This can be summarized as Equation (1).

$$l(f_t) = s(f_t) \times \frac{length}{stepCnt}, \quad (1)$$

where the function $s(f_t)$ receives the timestamp value when any fingerprint f is collected and returns the accumulated step, $length$ denotes the total length of the collection path, and $stepCnt$ is the total number of steps during collection.

If the step detection does not work correctly and a step length different from the step actually taken is determined, the distance from the starting point of any fingerprint estimated through step detection may differ from the actual distance. An optimization algorithm is used to compensate for errors that may occur owing to inaccurate step detection. The location where the random fingerprint was actually collected and the distance on the path to the starting point are the values that the optimization algorithm seeks. We set this value as x_i and use Equation (2) to find the value of x_i that minimizes the objective function. The Nelder–Mead algorithm is used to find the optimal solution of the objective function.

$$\arg \min_{\langle x_0, \dots, x_n \rangle} \left(\sum_{i=0}^{n-1} w_1 (\Delta x_i - \Delta s_i)^2 + w_0 (\Delta \theta_i - a_i)^2 \right), \quad (2)$$

where n denotes the number of collected fingerprints, $\langle x_0, \dots, x_n \rangle$ represents the distance from the starting point to each fingerprint, and Δx_i means $x_i - x_{i-1}$. Δs denotes $s_i - s_{i-1}$ when the distance from the starting point of the i -th fingerprint through step detection is s_i . θ_i represents the angle at which the fingerprint was rotated clockwise relative to the starting path when the distance from the starting point of any i -th fingerprint is x_i . a_i means the angle rotated clockwise from the starting path calculated from the gyroscope values collected when the corresponding fingerprint was collected. w_0 and w_1 are coefficients for the objective function to work correctly, and they have values of 0.3 and 0.7, respectively. In Equation (2), w_0 and w_1 are set to values that derive maximum accuracy through iterative simulation. In order to build more accurate learning data, these constants are determined by simulation in the offline phase. However, detailed explanation of this is omitted because the problem of optimizing these coefficients is outside the scope of this paper. The first term of the objective function maintains the gap between the fingerprint locations obtained through step detection as much as possible, and the second term compensates for errors that step detection may contain. That is, it is a term to place the fingerprint so that the difference between the angle rotated from the starting path, when the i -th fingerprint is estimated to be x_i from the starting point, and the angle rotated from the starting path, when the corresponding fingerprint was actually collected, is minimal. The values of x_i found through the corresponding algorithm represent the distance on the path from the starting point. The values converted to the two-dimensional (2D) coordinate system are estimated to be the closest to the location when each fingerprint was actually collected, and a radio map is constructed with the corresponding values and fingerprints.

3.3. Training of RHLS

To perform positioning through the structure of RHLS, it is necessary to train weights and biases in the network using extensive data. As described in Section 3.2, the area in which positioning service is to be provided is firstly set, and then the intersection and end point of the deck corridor are marked. The wireless signal, accelerometer, geomagnetic, and gyroscope sensing values are collected by the moving distance between the points where the location is known. Steps are detected using the accelerometer value, and the direction is estimated using the geomagnetic and angular velocity values. Then, the location coordinates of the starting point and the ending point are labeled with the value using the time at which the step was detected. For the location value, the value is obtained by scaling the location coordinates to a value between zero and one using the maximum and minimum values among the recorded coordinate values. Finally, the data are divided, according to the time at which the Wi-Fi signal is input, to match the input value introduced above.

A Wi-Fi scanning period is generally 3 to 4 s, and a person can walk at least three to four steps and at most six to eight steps during that time. Here, training is disrupted. In general, when training the RNN, an input sequence of a certain length is used, but considering the human step, the length of the input sequence is variable from three to eight. Naturally, when considering only the structure of the general RNN, the sequence needs not be constant. However, the longer the sequence is, the smaller the gradient value transmitted to the backpropagation becomes, such that smooth propagation is not achieved. Therefore, in general, the length of the sequence is fixed. However, in the case of this problem in which the sequence length is not constant, it is necessary to use an RNN cell that accepts the sequence length as additional information. The maximum length of the sequence is set to eight, and in the case of a sequence having a length shorter than this, only the information of the corresponding length is used. The RNN cell up to the maximum length was set to derive the zero vector as the output value, thereby solving the problem of the sequence length not being constant. On the one hand, when data exceed the limit set in relation to the mask used in the experiment, the limit must be increased and the input vector trained.

If the length of the sequence is not constant, the output value has to be processed. This is because a zero vector is output after the length of the input sequence. These zero vectors are output as meaningless values through the long short-term memory (LSTM) module of the middle layer and the

sigmoid layer of the output layer. Because this value generates and transmits an incorrect gradient during the backpropagation process, it is necessary to remove this value and use the output value only for the length of the input sequence. As a way to handle this, the output value is multiplied by a mask.

The mask is a Boolean vector, which has a value of one as long as it has a valid value, and zero after that. For example, if the maximum sequence length is five, and the input sequence length is three, the mask is composed of [1 1 1 0 0], as shown in Figure 4. The final output value is then multiplied, leaving only valid values. Using this method, in batch training, where different input sequences are input and trained at the same time, training can be performed without being influenced by a false gradient.

Figure 4. Masking result.

3.4. Positioning Module of RHLS

The positioning module of RHLS is composed of ANNs, as shown in Figure 5. The positioning module is divided into an input layer that processes input values, a middle layer composed of artificial neurons, and an output layer that converts the resulting values into metric coordinates through scaling.

Figure 5. Structure of the recurrent neural network (RNN)-based hybrid localization system (RHLS) positioning module.

3.4.1. Input Layer

The input layer of the proposed system processes Wi-Fi fingerprint and multiple sensor data in one sequence. The input layer is normalized using min–max feature scaling, such that the received wireless signal data and sensor data are between zero and one, to train the middle layer composed of artificial neurons more efficiently. For wireless signals conforming to the institute of electrical and electronics engineers (IEEE) 802.11 standard, the minimum signal strength that can be received is -100 dBm, and the maximum signal strength that can be received is -10 dBm. Based on this, we create a conversion function as in Figure 6a with -100 dBm matched to “0” and -10 dBm matched to “1”.

Figure 6. Translate functions of the input layer: (a) Wi-Fi signal strength; (b) geomagnetic intensity; (c) geomagnetic inclination; (d) moving direction.

In the case of a signal that is not captured, it is matched with zero. In the case of the geomagnetism value, it has a value of 25 to 65 μT , but it may be larger or smaller owing to various factors such as the steel structure in an indoor environment. Based on an analysis of data from various locations in offshore environments, the maximum magnetic value was set to 200 μT , and, using this, the geomagnetic value was adjusted to between zero and one, as shown in Figure 6b. Magnetic inclination, as mentioned earlier, is a value representing an angle, and, because of its characteristics, it has only a value between -90° and 90° . Therefore, by dividing the collected value by 90, the value was adjusted to range from -1 to 1 , as shown in Figure 6c. In the case of moving direction, because it is expressed between -180° and 180° , as shown in Figure 6d, the direction was divided by 180 to give a value between -1 and 1 . Equation (3) expresses these functions mathematically. The Wi-Fi signal strength is s , the geomagnetic strength value is h , the magnetic inclination is i , and the direction is represented by o .

$$s_{normalized} = \begin{cases} 0 & , s < -100, \\ \frac{s - S_{min}}{S_{max} - S_{min}} = \frac{s + 100}{90} & , -100 \leq s < -10, \\ 1 & , s \geq -10. \end{cases} \quad (3)$$

$$h_{normalized} = \begin{cases} \frac{h - H_{min}}{H_{max} - H_{min}} = \frac{h - 200}{200} & , 0 \leq h < 200 \\ 1 & , h \geq 200. \end{cases} \quad (4)$$

$$i_{normalized} = \frac{i}{180}. \quad (5)$$

$$o_{normalized} = \frac{o}{360}. \quad (6)$$

3.4.2. Middle Layer

The middle layer of the RNN positioning module is mainly composed of an LSTM cell. Firstly, the normalized Wi-Fi fingerprint $\mathbb{x}_0$ at the input layer is passed through a single sigmoid layer to obtain processed feature vector $\mathbb{h}_0$.

$$\mathbb{h}_0 = \sigma(W_0 \mathbb{x}_0 + \mathbb{b}_0), \quad (7)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (8)$$

where W_0 is the weight matrix of the sigmoid layer, and $\mathbb{b}_0$ is the bias vector of the sigmoid layer. The composition of input vector $\mathbb{x}_t$ at a specific time t is set in the order of the length of the step l_t , direction of movement o_t , magnetic field intensity h_t , and magnetic inclination i_t .

$$\mathbb{x}_t = \begin{bmatrix} l_t & o_t & h_t & i_t \end{bmatrix}. \quad (9)$$

When the input value enters the LSTM cell, the result value is output through the following process:

$$\mathbb{f}_t = \sigma(W_f \cdot [\mathbb{h}_{t-1}, \mathbb{x}_t] + \mathbb{b}_f), \quad (10)$$

$$\mathbb{i}_t = \sigma(W_i \cdot [\mathbb{h}_{t-1}, \mathbb{x}_t] + \mathbb{b}_i), \quad (11)$$

$$\tilde{\mathbb{c}}_t = \tanh(W_c \cdot [\mathbb{h}_{t-1}, \mathbb{x}_t] + \mathbb{b}_c), \quad (12)$$

$$\mathbb{c}_t = \mathbb{f}_t * \mathbb{c}_{t-1} + \mathbb{i}_t * \tilde{\mathbb{c}}_t, \quad (13)$$

$$\mathbb{o}_t = \sigma(W_o \cdot [\mathbb{h}_{t-1}, \mathbb{x}_t] + \mathbb{b}_o), \quad (14)$$

$$\mathbb{h}_t = \mathbb{o}_t * \tanh(\mathbb{c}_t), \quad (15)$$

where result value $\mathbb{h}_t$ is a feature vector after t seconds from feature vector $\mathbb{h}_0$ of the basic Wi-Fi fingerprint.

Looking at the middle layer for a continuous time span, it appears similar to the structure on the right in Figure 7. In the case of a general RNN, the length of the sequence is determined, and, when the input is received for this sequence, the RNN is initialized. However, in the proposed system, the RNN cell is initialized in accordance with the Wi-Fi signal scan. When the Wi-Fi signal scan is finished and the RSS vector is constructed, the vector is normalized through the input layer. This vector is processed once through the sigmoid layer and initializes the LSTM. Then, the LSTM cell continues to store and use the values until a new scan is completed. This structure creates a tracking effect between Wi-Fi scans.

Figure 7. Structure of the middle layer.

3.4.3. Output Layer

The output layer of the recurrent network positioning module is responsible for the final metric calculation. Each element of feature vector $\mathbb{h}_t$ at time t generated through the middle layer has a value between -1 and 1 . This is passed through another sigmoid layer to process the vector once more. This is expressed as follows:

$$y_t = \sigma(W_1 \cdot \mathbb{h}_t + \mathbb{b}_1). \quad (16)$$

As a result of the final regression through the above processes, y_t becomes a two-dimensional vector in which each element value is scaled between zero and one. The actual coordinate value is calculated by increasing this to the original map scale as shown in Figure 8. For example, if the area where the signal is collected is an area of 10 m in width and 5 m in length, the output horizontal value is 10 times, and the vertical value is five times, which are converted into coordinates in meters.

Figure 8. RHLS scaling function.

4. Results

4.1. Experimental Set-Up

Experiments were conducted offshore at the Daewoo Shipbuilding and Marine Engineering shipyard, commonly known as DSME. In the experiments, the performance of the RHLS was evaluated using four decks in the residence. Figure 9 depicts the experimental area. Each deck has a narrow corridor structure (52.7, 36.4, 50.7, and 67.8 m) in total length. In the tracking accuracy test, Wi-Fi, Bluetooth, and geomagnetic data were used for absolute positioning, and a gyroscope and accelerometer were used as PDR sensors. For learning data construction, three to six Wi-Fi APs and BLE beacons were installed for each deck at approximately 5–10-m intervals, for a total of 21 APs and beacons. Figure 9 shows the details of the test space and the locations of the installed APs and beacons. A Galaxy Nexus was used as a reference device to collect training and test data. The specifications of the computer used in the experiments were as follows: Intel i3-4170 (3.7 GHz, 2C/4T) central processing unit (CPU), Nvidia GeForce GTX 1060 (6 GB) graphics processing unit (GPU), and DDR3 8 GB random-access memory (RAM). The proposed method using LSTM was performed using a GPU. The remainder of the comparison group computations were performed using only the CPU.

Geomagnetic learning data are the norm values of the x , y , and z axes and were assigned to the magnet intensity, while the averages were used along with the inclination values. The average values of Wi-Fi, Bluetooth, and magnetic field data in the target area were stored as training data, along with the location coordinates, during the construction of the training database. These previously collected data received an average of 9.86 AP or beacon signals for each fingerprint. An Adam optimizer was used as the optimization algorithm, and the training rate was set from 0.0005 to 0.001. The proposed structure was trained by grouping eight input datasets and repeating it up to 1,000,000 times. For the accuracy test, the test point coordinates were stored in the test data, and the accuracy was calculated using the difference in distance between the test point and ground truth. The coordinates were recorded manually directly at each test point. At the test point, the difference in distance between the coordinates that the algorithm derived and the handwritten coordinates represents the positioning accuracy. Figure 10 shows the radio map collected for each deck in the form of heatmaps according to the collection and signal density.

Figure 9. Hybrid positioning test space and access point (AP) beacon locations: (a) main deck; (b) A deck; (c) first deck; (d) second deck.

Figure 10. Collected radio maps for each deck: (a) main deck; (b) A deck; (c) first deck; (d) second deck.

The test data used in the positioning accuracy experiment were collected continuously along a path. Test data were collected twice for each route to minimize bias according to the test route. Figure 11 shows the test paths for each deck.

4.2. Tracking Accuracy Test

The tracking accuracy achieved by the RHLS was compared with that of a model built in a supervised manner using the ground truth location labels. For the positioning accuracy test of the four different decks, it was necessary to conduct the training and test for each deck separately. The average error distances were measured according to the time sequences of the test data. For fast training, eight data readings were processed simultaneously, and six readings for each route, that is, a total of six sets of data, were used for training. Two readings for each test were selected for each route, and training was terminated when the error of these data became less than 1.5.

For comparison, the HMM-based Viterbi algorithm was evaluated by setting the maximum allowable single-step moving distance to 1.1 m. In addition, a commonly used k-nearest neighbor (kNN) method was used to compare the differences arising from continuous and static positioning (k was set to three). For the internal parameters of the Viterbi algorithm, the standard deviation of the stride was set to 0.6 m, the standard deviation of the heading was set to 45° , the standard deviation of

the Wi-Fi signal was set to 6 dBm, the standard deviation of the magnetic field strength was set to $6 \mu T$, and the standard deviation of the magnetic inclination was set to 0.5 radians. Under the assumption that APs were properly deployed, 20 APs were used. Figure 12 is a floor plan showing the estimated location, ground truth, and error distance for each test path. The blue dots indicate the ground truth on the test path, the red dots indicate the estimated location, and the solid line between the two points indicates the error distance. Table 1 shows the results of comparison with the proposed method.

Figure 11. Test data collection path for each deck: (a) main deck; (b) A deck; (c) first deck; (d) second deck.

Figure 12. Hybrid positioning result example by deck: (a) main deck; (b) A deck; (c) first deck; (d) second deck.

Table 1. Positioning accuracy by deck. k-NN—k-nearest neighbor.

	RHLS	Viterbi	k-NN
Main deck	2.88	3.24	3.38
A deck	2.32	2.76	3.41
1st deck	2.74	3.21	3.44
2nd deck	2.94	3.35	3.74
Mean error (m)	2.72	3.14	3.49
Compute time (ms)	0.32	47.10	1.97

Although there were some errors depending on the environment, we confirmed that RHLS exhibits sufficient location accuracy for the monitoring service despite the presence of diffuse reflections of the signal. This is because the result of localization through fingerprinting and tracking is corrected through the proximity and map matching techniques. Depending on the space, it was confirmed that the second deck, the hole-shaped space, exhibited the best positioning accuracy, and the corridor-type spaces exhibited relatively good positioning accuracy. The average error of the proposed method was 2.72 m, which is a good result compared to other methods. In addition, it can be seen that the time required for positioning was significantly reduced compared to other methods.

Figure 13 compares the tracking accuracy of the sequence for each deck between the RHLS and the Viterbi method. It can be seen that the proposed method shows overall high accuracy regardless of the shape of the deck. In the case of a ship environment, signal transmission such as Wi-Fi is difficult because of the steel plate structure with a narrow corridor, and the signal fluctuation range is greater than that in a general land building owing to diffuse reflection. In addition, in the case of a ship, the AP environment is not sufficiently rich to derive high positioning accuracy, unlike a land building, to which various positioning methods are applied (such as References [28,29]). Therefore, positioning accuracy is lower than that of a land building. The data collected and tested in this paper included all these environmental factors, and they were used in the experiment. Because the initial accuracy was not high, the most important factor was how much it was corrected through tracking. Although it can be seen that the tracking accuracy was corrected with time in the Viterbi method, it can be confirmed that the correction width was less than that of a general land building owing to unstable signal data. On the other hand, in the RHLS, the accuracy convergence was greater, and the data and features used in the LSTM learning process successfully reflected the special ship environment.

This is especially noticeable on the second deck, which took the form of an open space. In the open space, it was difficult to use the specificity of the indoor structure; hence, the accuracy correction by PDR was relatively small. Therefore, the results of the Viterbi experiment show that the initial accuracy and the accuracy after localization convergence rarely differed; however, in the RHLS, it can be seen that the proposed method showed an accuracy improvement of approximately 16% over the course of the sequence. On the A deck, where it was difficult to estimate the heading estimation effect of the PDR as a straight-line test trace, it was also found that the proposed method shows superior accuracy improvement compared to the existing Viterbi method. Because there was no direction change, it took a long time to converge compared to other decks, but the accuracy was continuously improved, resulting in an accuracy improvement of approximately 29%.

Figure 13. Comparison of tracking accuracy as the sequence progresses: (a) main deck; (b) A deck; (c) first deck; (d) second deck.

5. Conclusions

This paper proposed RHLS, an indoor positioning system that estimates locations by fusing various embedded sensors of smartphones using an RNN. The performance was verified through comparison by conducting experiments on a ship at anchor. To the best of our knowledge, there was never before an actual indoor location system test in an offshore environment.

In experiments, it was confirmed that RHLS had improved accuracy and computation time over the existing methods. The main feature of this system is that it organically combines data with different cycles. The existing probability-based sensor fusion algorithms must obtain the probability distribution of wireless signals or sensors through extensive data collection or assumptions. In contrast, the proposed method collects data and trains using those data; thus, it has a location service provision environment to reduce the required effort. The tracking effect is generated from the RNN structure; therefore, it can be confirmed that the accuracy improves over time. Finally, the RHLS was able to significantly increase the speed compared to existing sensor fusion algorithms. The proposed RHLS was able to dramatically reduce the prediction process while using an ANN. This can solve the problem of information loss according to the calculation time, which was a problem raised earlier.

In the proposed system, the user location is calculated by combining sensor values based on the Wi-Fi signal vector. If a new Wi-Fi or Bluetooth signal is added for positioning, it is necessary to train using the corresponding newly added data because of the feature of the proposed method. However, this situation does not happen very often, because the layout of the infrastructure on a ship is completely specified when the ship is first designed and, once set up, subsequent changes are

very rare. Nevertheless, if the AP is broken or removed and another Wi-Fi vector is input at the same location, it is likely to generate an error. In addition, the proposed method assumes that the user has a similar step length in consideration of a slow walking situation, not a running or fast walking situation. This is because it is not common to move quickly in a place like a ship. However, if the step length changes, the network should be retrained.

In this regard, improvements are planned for the proposed method such as applying more efficient and accurate RNN techniques for general land buildings with more in-depth experiments, including analysis of the completed weight matrix to investigate the RNN parameters and the positioning errors. In addition, it is necessary to conduct training or inspection as future work using data with different steps or a dynamically changing environment to confirm that the proposed system is robust.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Available online: <https://www.bsee.gov/stats-facts/offshore-incident-statistics> (accessed on 24 May 2020).
2. Jung, S.H.; Moon, B.C.; Han, D.S. Unsupervised learning for crowdsourced indoor localization in wireless networks. *IEEE Trans. Mobi. Comput.* **2016**, *15*, 2892–2906. [\[CrossRef\]](#)
3. Liu, H. Push the limit of WiFi based localization for smartphones. *ACM Mobicom*. **2012**, 305–316. [\[CrossRef\]](#)
4. Xiao, Z. Non-line-of-sight identification and mitigation using received signal strength. *IEEE Trans. Wireless Commun.* **2015**, *14*, 1689–1702. [\[CrossRef\]](#)
5. Sudarshan, S.C. Beacon placement for indoor localization using Bluetooth. In Proceedings of the 2008 11th International IEEE Conference on Intelligent Transportation Systems, Beijing, China, 12–15 October 2008. [\[CrossRef\]](#)
6. Want, R.; Hopper, A.; Falcao, V.; Gibbons, J. The active badge location system. *ACM Trans. Inf. Syst.* **1992**, *10*, 91–102. [\[CrossRef\]](#)
7. Bekkail, A.; Sanson, H.; Matsumoto, M. RFID indoor positioning based on probabilistic RFID map and Kalman filtering. In Proceedings of the Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, New York, NY, USA, 8–10 October 2007. [\[CrossRef\]](#)
8. Lionel, M.N.; Yunhao, L.; Yiu, C.L.; Abhishek, P.P. LANDMARC-Indoor location sensing using active RFID. *Wireless Network* **2004**, *10*, 701–710. [\[CrossRef\]](#)
9. Enrique, C.M.; Francisco, J.; David, C.; Ana, B.; Pedro, S.; Felipe, G. QR-Maps: An efficient tool for indoor user location based on QR-codes and Google Maps. In Proceedings of the 2011 IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA, 9–12 January 2011; pp. 928–932. [\[CrossRef\]](#)
10. Chun, Y.; Thao, N.; Erik, B. Mobile positioning via fusion of mixed signals of opportunity. *IEEE Aerosp. Electron. Syst. Mag.* **2014**, *29*, 34–46. [\[CrossRef\]](#)
11. Sudhir, K.; Rajesh, M. Multi-sensor data fusion for indoor localization under collinear ambiguity. *Perv. Mobi. Comput.* **2015**, *30*, 18–31. [\[CrossRef\]](#)
12. Chen, Z.; Zou, H.; Jiang, H.; Zhu, Q.; Chai, Y.; Xie, L. Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization. *Sensors* **2015**, *15*, 715–732. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Suhr, J.; Min, D.; Jung, H. Sensor fusion-based low-cost vehicle localization system for complex urban environments. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1078–1086. [\[CrossRef\]](#)
14. Xiang, H.; Daniel, N.; Jia, L. Probabilistic multi-sensor fusion based indoor positioning system on a mobile device. *Sensors* **2015**, *15*, 31464–31481. [\[CrossRef\]](#)
15. Jung, S.; Lee, G.; Han, D. Method and tools to construct a global indoor positioning system. *IEEE Trans. Syst. Man Cyber. Syst.* **2017**, *48*, 906–919. [\[CrossRef\]](#)
16. Minh, T.; Brosnan, Y.; Xiaodal, D.; Tao, L.; Robert, W.; Kishore, R. Recurrent neural networks for accurate RSSI indoor localization. *IEEE Internet Things* **2019**, *6*, 10639–10651. [\[CrossRef\]](#)
17. Gan, X.; Yu, B.; Huang, L.; Li, Y. Deep learning for weights training and indoor positioning using multi-sensor fingerprint. In Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation, Sapporo, Japan, 18–21 September 2017. [\[CrossRef\]](#)

18. Belay, A.; Lin, H.; Tarekegn, G.; Jeng, S. Applying deep neural network (DNN) for robust indoor localization in multi-building environment. *Appl. Sci.* **2018**, *8*, 1–14. [[CrossRef](#)]
19. Xiao, L.; Behboodi, A.; Mathar, R. A deep learning approach to fingerprinting indoor localization. In Proceedings of the 2017 27th International Telecommunication Networks and Applications Conference, Melbourne, Australia, 22–24 November 2017. [[CrossRef](#)]
20. Xiao, C.; Yang, D.; Chen, Z.; Tan, G. 3-D BLE indoor localization based on denoising autoencoder. *IEEE Access* **2017**, *5*, 12751–12760. [[CrossRef](#)]
21. Wang, X.; Gao, L.; Mao, S.; Pandey, S. CSI-based fingerprinting for indoor localization: A deep learning approach. *IEEE Trans. Vehic. Tech.* **2017**, *66*, 763–776. [[CrossRef](#)]
22. Li, R.; Liu, Q.; Gui, J.; Gu, D.; Hu, H. Indoor relocalization in challenging environments with dual-stream convolutional neural networks. *IEEE Trans. Autom. Sci. Engi.* **2018**, *15*, 651–662. [[CrossRef](#)]
23. Liu, Z.; Dai, B.; Wan, X.; Li, X. Hybrid wireless fingerprint indoor localization method based on a convolutional neural network. *Sensors* **2019**, *19*, 4597. [[CrossRef](#)] [[PubMed](#)]
24. Zhang, G.; Wang, P.; Chen, H.; Zhang, L. Wireless indoor localization using convolutional neural network and Gaussian process regression. *Sensors* **2019**, *19*, 2508. [[CrossRef](#)] [[PubMed](#)]
25. Takenga, C.; Xi, C.; Kyamakya, K. A hybrid neural network-data base correlation positioning in GSM network. In Proceedings of the 2006 10th IEEE Singapore International Conference on Communication Systems, Singapore, 30 November 2006. [[CrossRef](#)]
26. Stella, M.; Russo, M.; Begusic, D. Location determination in indoor environment based on RSS fingerprinting and artificial neural network. In Proceedings of the 2007 9th International Conference on Telecommunications, Zagreb, Croatia, 13–15 June 2007. [[CrossRef](#)]
27. Jang, H.; Shin, J.; Choi, L. Geomagnetic field based indoor localization using recurrent neural networks. In Proceedings of the GLOBECOM 2017-2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017. [[CrossRef](#)]
28. Valerie, R.; Miguel, O.; Johan, P.; Joaquín, T.; Antonio, R.; Antoni, P.; Germán, M.; Fernando, S.; Yael, L.; Revital, M.; et al. Evaluating indoor positioning systems in a shopping mall: The lessons learned from the IPIN 2018 competition. *IEEE Access* **2019**, *7*, 148594–148628. [[CrossRef](#)]
29. Joaquín, T.; Antonio, R.; Adriano, M.; Tomás, L.; Wei-Chung, L.; Stefan, K.; Germán, M.; Fernando, S.; Antoni, P.; Maria, J.; et al. Off-line evaluation of mobile-centric indoor positioning systems: The experiences from the 2017 IPIN competition. *Sensors* **2018**, *18*, 487. [[CrossRef](#)]

© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).