



Article

UWB Indoor Localization Using Deep Learning LSTM Networks

Alwin Poulouse  and Dong Seog Han * 

School of Electronics Engineering, Kyungpook National University, 80 Daehak-ro, Buk-gu, Daegu 41566, Korea; alwinpoulosepalatty@knu.ac.kr

* Correspondence: dshan@knu.ac.kr; Tel.: +82-53-950-6609

Received: 28 July 2020; Accepted: 7 September 2020; Published: 10 September 2020



Abstract: Localization using ultra-wide band (UWB) signals gives accurate position results for indoor localization. The penetrating characteristics of UWB pulses reduce the multipath effects and identify the user position with precise accuracy. In UWB-based localization, the localization accuracy depends on the distance estimation between anchor nodes (ANs) and the UWB tag based on the time of arrival (TOA) of UWB pulses. The TOA errors in the UWB system, reduce the distance estimation accuracy from ANs to the UWB tag and adds the localization error to the system. The position accuracy of a UWB system also depends on the line of sight (LOS) conditions between the UWB anchors and tag, and the computational complexity of localization algorithms used in the UWB system. To overcome these UWB system challenges for indoor localization, we propose a deep learning approach for UWB localization. The proposed deep learning model uses a long short-term memory (LSTM) network for predicting the user position. The proposed LSTM model receives the distance values from TOA-distance model of the UWB system and predicts the current user position. The performance of the proposed LSTM model-based UWB localization system is analyzed in terms of learning rate, optimizer, loss function, batch size, number of hidden nodes, timesteps, and we also compared the mean localization accuracy of the system with different deep learning models and conventional UWB localization approaches. The simulation results show that the proposed UWB localization approach achieved a 7 cm mean localization error as compared to conventional UWB localization approaches.

Keywords: Indoor localization; ultra-wide band (UWB) signals; time of arrival (TOA); deep learning; long short-term memory (LSTM); trilateration algorithm

1. Introduction

Recent advancements in indoor localization technologies [1] give accurate position results when the GPS signals are unable to detect user position in an indoor scenario. The most popular building dependent localization technology is the ultra-wideband (UWB) localization system. In UWB localization, multiple anchors are placed in the experiment area and the user uses a UWB tag to communicate with anchors. The anchors and the tag communicate through UWB signals and the time of arrival (TOA) [2] of UWB signals are used to estimate the distance of the user from anchors. The position accuracy of a UWB system depends on the UWB signal shapes, TOA of UWB signals and user distance estimation from the anchors. For an accurate user position estimation in UWB systems, it is necessary to estimate the user distance with a minimum error. In UWB localization, the localization algorithms use the user distance information from the TOA-distance model. The trilateration algorithm is the most common localization algorithm used in UWB localization. However, the position accuracy of the trilateration algorithm depends on the user distance from the anchors [3]. The multipath effects,

channel conditions between anchors and tag, and the UWB signal shapes affect the distance estimation accuracy. This leads to a high position error in the UWB localization systems.

To reduce the user position errors in the UWB system, we propose a deep learning approach to UWB localization. The proposed model consists of a deep neural network which takes as input the distance information from the TOA-distance model and predicts the user positions. The proposed model uses a two layer long short-term memory (LSTM) network [4] which identifies the accurate user positions. As compared to neural network (NN), the LSTM has feedback loop which serves as a memory and gives better performance for time series analysis. The flexibility in controlling the input data from the LSTM model gives a better localization performance than other models. The constant error backpropagation within memory cells results in LSTM model can able to bridge very long-time lags for UWB localization. The unique features of LSTM model show that it is the best choice for our localization purpose.

Deep learning for UWB localization has been investigated through several approaches to improve localization performance. Many researchers proposed different deep learning approaches such as convolutional neural networks (CNN) for UWB transmitter-receiver distance estimation [5], non-line of sight (NLOS) channel classification using channel impulse response (CIR) information [6,7] and the neural network for UWB fingerprint-based localization [8]. However, to the best of our knowledge, there is no comparative analysis of hyperparameter tuning for LSTM based UWB localization. Therefore, it is necessary to investigate the best hyperparameters for LSTM based UWB localization and this paper analyses the performance of LSTM deep learning networks in terms of learning rate, optimizer, loss function, batch size, number of hidden nodes and timesteps. The best parameters for LSTM networks are identified through several simulation results and finally, we compare the proposed model prediction results with conventional UWB localization approaches such as linearized least square estimation (LLSE), fingerprint estimation (FPE), maximum likelihood estimation (MLE) and weighted centroid estimation (WCE) algorithms [9]. The performance of the LSTM model also analyzed by mean localization error with a different number of LSTM layers, training and testing time of UWB data in LSTM model and the mean localization error with a different number of anchors. Through extensive simulations, we demonstrated the superior performance of the proposed LSTM model. The following are the main contributions to this paper:

- We simulated a UWB system with three anchors and one tag, and generated Gaussian pulses as UWB signals. The Gaussian pulses transmitted through UWB channel conditions from anchors to tag and estimated the TOA of UWB pulses.
- We formulated a TOA-distance model and estimated the tag distance from anchors. The estimated tag distances from anchors are used for predicting the user position.
- We introduced the LSTM deep learning architecture for UWB localization. The performance of the LSTM architecture was analyzed and we estimated the best parameters for the LSTM model.

The rest of the paper is organized as follows—Section 2 summarizes the previous work on UWB localization, Section 3 discusses the proposed UWB indoor localization approach using deep learning LSTM networks. The analysis of simulation results is provided in Section 4. Finally, the conclusions are summarized in Section 5.

2. Related Work

Indoor localization using UWB measurement has been studied in the past and in recent times based on TOA [10], time difference of arrival (TDOA) [11], phase of arrival (PoA) [12], angle of arrival (AOA) [13], joint TOA and AOA [14] and UWB localization algorithms have also been studied [15,16]. In this section, we discuss different UWB localization systems that leverage the advantages of deep learning networks.

The idea of machine learning in the UWB indoor localization research area was proposed in Reference [17] and this proposed system combines grid-search-based kernel support vector machine

and principle component analysis (PCA). The results from Reference [17] show that the proposed approach reduces the localization error and improves the computational efficiency when compared to k-nearest neighbor (kNN), back propagation neural network (BPNN) and support vector machine (SVM) based methods. An SVM-based localization approach which classifies the UWB NLOS conditions for indoor localization is presented in Reference [18]. The SVM model in Reference [18] reached 92% average identification accuracy for NLOS conditions and improved the UWB position accuracy in multipath conditions. To improve the localization accuracy of the UWB system during NLOS conditions, a two-stage SVM classification for UWB LOS/NLOS channel conditions is discussed in Reference [19]. The system from Reference [19] shows 93.7% efficiency for identifying the LOS/NLOS channel conditions. However, selecting the adequate kernel and the parameter optimization are the major challenges faced by the SVM-based localization systems. A logistic regression-based localization approach is an alternative way to improve the UWB localization performance for the non-line-of-sight (NLOS) condition. The regression model presented in Reference [20] addressed the NLOS detection error mitigation of differential time of arrival (TDoA) topologies and suppresses the NLOS error up to 80%. In Reference [21], the authors also introduced another regression-based localization method for identifying the LOS conditions. The PCA, kNN, SVM and logistic regression approaches give better performance for UWB localization, however, the multipath effects, signal interference, sensitive to channel inconsistency and computational complexity of the models add localization errors to the UWB system. To extract the correct ToA during multipath scenarios, a signal processing method that uses deep learning to estimate the absolute UWB tag position directly from the raw channel impulse response (CIR) data is proposed in Reference [22]. The localization system explained in Reference [22] uses convolutional neural network (CNN) and achieved 17.3 cm mean absolute localization error for NLOS conditions. However, the overfitting & underfitting, parameter tuning and processing time of data from CNN models give challenges to the UWB localization. Most recent works on deep learning for UWB localization are explained in References [23–25]. The discussed localization systems from References [23–25] effectively minimize the ranging errors and computational burden and achieve excellent localization performance with much lower network complexity for UWB indoor localization.

The UWB localization systems discussed in the related works addressed localization accuracy challenges and proposed a number of systems to improve indoor position accuracy. However, the UWB system still needs improvement in terms of localization accuracy. Therefore, it is necessary to propose a new localization approach for UWB-based indoor localization. To reduce the localization error, this paper proposes an LSTM-based localization approach based on the simulation environment. The simulation results indicate that the proposed approach gives reasonable position accuracy for indoor localization and reduces the UWB localization errors.

3. The Conventional and Proposed UWB Localization Approaches

Conventional UWB localization approaches tend to use different localization algorithms in the system controller to estimate the current user positions. The positional accuracy of the conventional methods depends on the type of the localization algorithm and the configuration of the system controller. To improve the indoor position accuracy for UWB localization, we used deep learning instead of localization algorithms and this proposed UWB indoor localization system leverages the advantages of LSTM networks in predicting user positions. As compared to the conventional UWB system, the proposed system utilizes a deep learning architecture in the system controller to predict the user position. The deep learning model is trained on user distance from anchors and gives accurate position results. Figure 1 shows the conventional and proposed UWB localization systems.

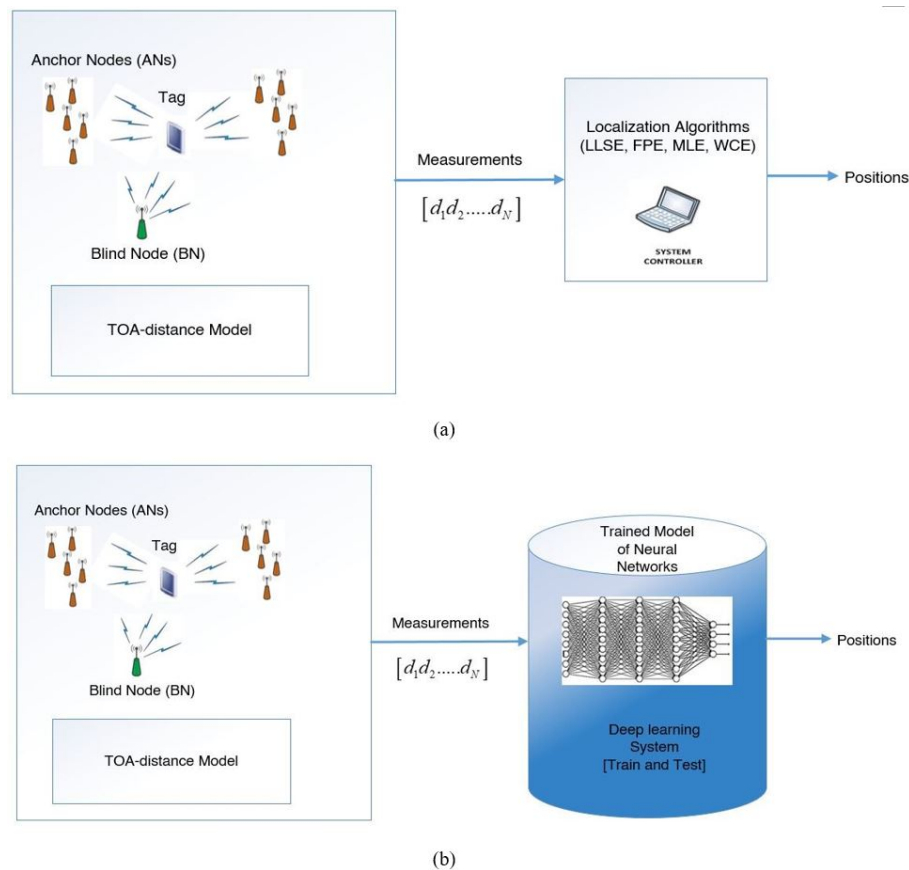


Figure 1. Ultra-wideband (UWB) localization systems. (a) Conventional UWB localization system. (b) Proposed UWB localization system using deep learning.

The proposed model consists of anchor-tag communication using UWB pulses, the TOA-distance model and deep learning system for user position prediction. First, we place the anchor nodes (ANs) with known coordinates in the experiment area. The UWB tag sends UWB pulses to the ANs. All ANs receive UWB pulses and estimate the distances from each AN. Since the anchors receive UWB pulses at different intervals from the UWB tag, the signals arrive at each anchor at a different time. The signal arrival of time from UWB tag is multiplied by the constant speed of light in the air resulting in the ANs and UWB tag distances. To estimate the UWB tag position, the concept of trilateration is used. To use trilateration, it is necessary to use at least three anchors in the experiment area. In our simulation, we created an experiment area with three anchors and the anchors were placed in known coordinates to cover the entire experiment area. To generate UWB pulses, we used Gaussian pulses with the IEEE 802.15.4a UWB system standard. The generated UWB pulses transmitted from the UWB tag to all anchors. The UWB tag moved in the predefined paths and sends pulses to the ANs. The ANs use UWB TOA information and estimate the tag distances from ANs. The distance information matrix from TOA-distance model is used as the input to the deep learning system. The deep learning system uses two LSTM layers to predict the tag's x and y positions. The LSTM layer takes three inputs, the three distance values from three anchors, and predicts the current UWB tag position. The proposed LSTM model predicts accurate tag position results with minimal localization errors. The experiment results show that the proposed deep learning approach solved the LOS problems associated with UWB localization systems and reduces the computational complexity of localization algorithms. As compared to conventional localization approaches, the proposed method can be easily used for complex trajectories and estimate the user position with minimum position errors.

3.1. UWB Model

The model presented in Reference [26] is used for the UWB simulation. Figure 2 shows the flow chart of the UWB localization system.

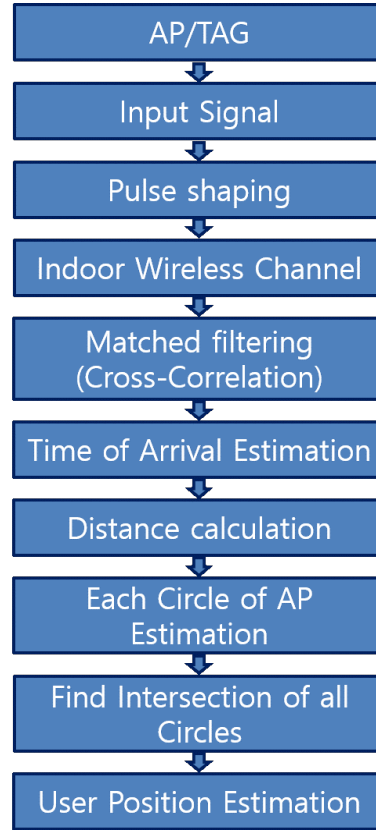


Figure 2. UWB based localization algorithm.

The UWB localization system consists of UWB signal generation, transmission of UWB signals through indoor channel, cross relation of received UWB signals with reference signal, TOA estimation of UWB signals, distance estimation from TOA and finally estimate the current user position using the trilateration approach. To generate the UWB pulses as input signal for UWB simulation, we use the Gaussian pulse and is defined as

$$y(t) = \frac{A}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{t^2}{2\sigma^2}\right)}, \quad (1)$$

where A is the amplitude of the signal and σ is the standard deviation related to the pulse shape factor. The simulation uses the pulse width of 0.5 ns and Figure 3 shows the generated UWB pulse for transmission.

The UWB transmitted signal is expressed as [26]

$$s(t) = \sum_{i=0}^{N-1} p(t - i \times T_{\text{int}}), \quad (2)$$

where $p(t)$ is the UWB pulse, T_{int} is the sum of guard time and pulse width, N is the number of pulses. Then the received signal is expressed as

$$r(t) = h(t) \times s(t) + n(t), \quad (3)$$

where $h(t)$ is the channel impulse response and $n(t)$ is the AWGN noise. When we consider the average time of received signal, the noise term is eliminated and is expressed as

$$r_{\text{avr}}(t) = \sum_{i=0}^{N-1} \{p(t) \times h_i(t + i \times T_{\text{int}})\}. \quad (4)$$

After the average operation of received signal, we perform a correlation between the averaged signal and the template signal and is expressed as

$$S_{\text{Crr}}(t) = \int_0^{T_{\text{int}}} r_{\text{avr}}(\tau) \times p(t - \tau) d\tau. \quad (5)$$

The correlated output signal from one anchor node to UWB tag is shown in Figure 4.

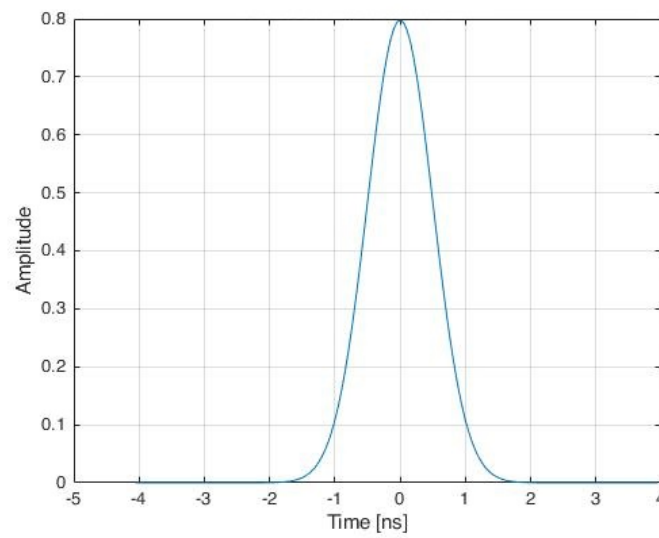


Figure 3. UWB pulse.

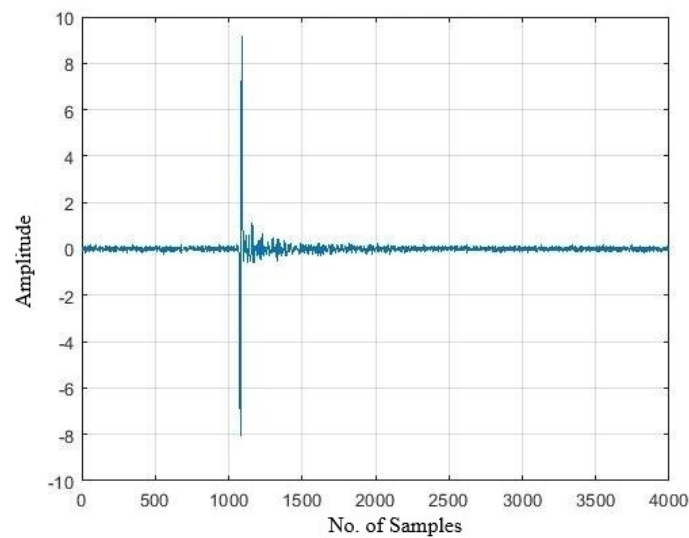


Figure 4. Correlated output signal.

The correlated output signal is used in the peak detector for estimating the TOA of signal. In the simulation process, we considered IEEE 802.15.4a UWB channel model [27], which characterizes

industrial LOS environment. The UWB pulses propagate through a large number of paths and the channel impulse response is expressed as

$$h(t) = \sum_{l=0}^L \sum_{k=0}^k \alpha_{k,l} \exp(j\phi_{k,l}) \delta(t - T_l - \tau_{k,l}), \quad (6)$$

where $\alpha_{k,l}$ is the multipath gain coefficient, T_l is the delay of the l^{th} cluster and $\tau_{k,l}$ is the delay of the k^{th} multipath component relative to the l^{th} cluster arrival time (T_l). The phases $\phi_{k,l}$ are uniformly distributed in the range $[0, 2\pi]$ and the number of clusters L are assumed to be the Poisson distribution. Table 1 shows the parameter settings for the UWB simulation.

Table 1. UWB simulation parameter values.

UWB Simulation Parameter	Value
Speed of Light	3×10^8 m/s
Pulse shape	Gaussian pulse
Number of anchors	3
Coordinates of anchors (x, y)	[5 18; 18 5; 18 18]
Tag's initial coordinate (x, y)	[0 2]
Number of bits	100
Pulse repetition interval	200 ns
SNR range (in dB)	3 dB
Sample rate	20 GHz
Pulse width	0.5 ns

The overall UWB simulation stages are summarized in Figure 5. To generate the UWB simulation data, our simulation ran on MATLAB and defined an indoor experiment area in the MATLAB simulation environment. The anchors positions are defined and the UWB tag moved in the predefined trajectories. In the UWB transmitter, we generated Gaussian pulses for transmission and these signals are transmitted through the indoor UWB channel. On the receiver side, the UWB receiver receives the UWB signal with Gaussian noise through the UWB channel. To remove the Gaussian noise, the receiver took the average of the signal and passed the signal onto the correlation block. In the correlation block, correlation between the received signals and reference signal is performed and the correlation output is used in the peak detector block. The peak detector block estimates the TOA of the UWB signal based on the threshold setting and the overall UWB transmission-reception is illustrated in Figure 5 as a block diagram.

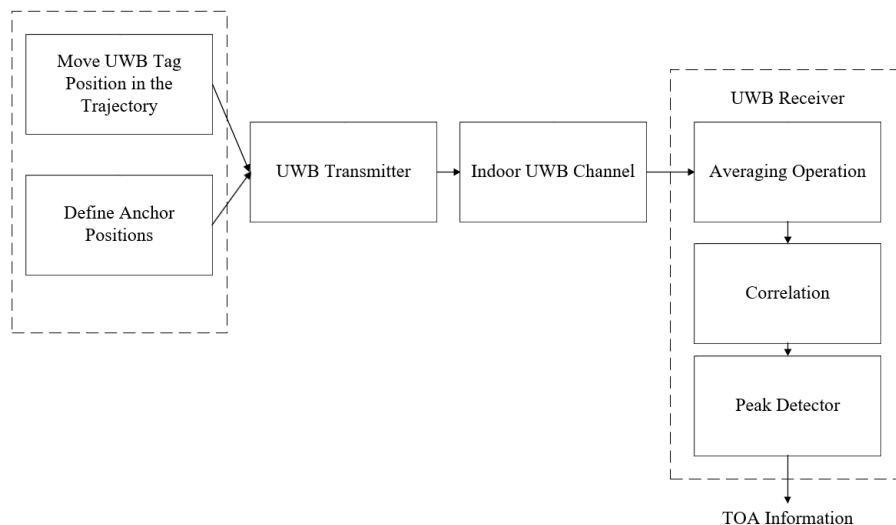


Figure 5. Block diagram of overall UWB simulation system.

3.2. Trilateration Approach for UWB Localization

To train the proposed LSTM network with distance information from anchors and user position coordinates, we use trilateration. The trilateration algorithm estimates the true trajectory coordinates and many position coordinates from trajectory which covers the entire experiment area. The UWB tag distance from anchors and the user positions from trilateration approach are used in training the LSTM model. Figure 6 shows the trilateration approach for UWB localization.

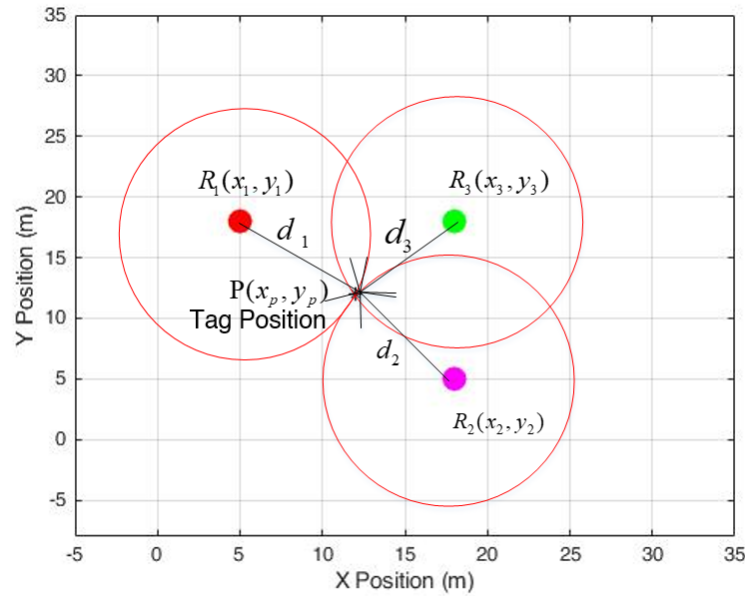


Figure 6. UWB localization using trilateration approach.

The distance d_i between the UWB tag $P(x_p, y_p)$ and the anchors $R_i(x_i, y_i)$ is expressed as

$$d_i^2 = (x_i - x_p)^2 + (y_i - y_p)^2. \quad (7)$$

The expanded form of (7) is

$$d_i^2 = x_i^2 + x_p^2 - 2x_i x_p + y_i^2 + y_p^2 - 2y_i y_p. \quad (8)$$

When we consider a reference point K , then (8) can be expressed as

$$d_k^2 = x_k^2 + x_p^2 - 2x_k x_p + y_k^2 + y_p^2 - 2y_k y_p. \quad (9)$$

Subtracting (9) from (8), we get

$$d_i^2 - d_k^2 + x_k^2 + y_k^2 - x_i^2 - y_i^2 = 2(x_k - x_i)x_p + 2(y_k - y_i)y_p. \quad (10)$$

Considering $i = 1$ and varying index $k = 2, 3$ we obtain

$$\begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} d_1^2 - d_2^2 + x_2^2 + y_2^2 - x_1^2 - y_1^2 \\ d_1^2 - d_3^2 + x_3^2 + y_3^2 - x_1^2 - y_1^2 \end{bmatrix}. \quad (11)$$

The coordinates $P(x_p, y_p)$ of the UWB tag can be obtained by solving the above system of equations. We can obtain a linear equation with three unknowns, $Ax = b$. The A , x and b values of liner equation is expressed

$$A = \begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \end{bmatrix} \quad (12)$$

$$x = \begin{bmatrix} x_p \\ y_p \end{bmatrix} \quad (13)$$

$$b = \begin{bmatrix} d_1^2 - d_2^2 + x_2^2 + y_2^2 - x_1^2 - y_1^2 \\ d_1^2 - d_3^2 + x_3^2 + y_3^2 - x_1^2 - y_1^2 \end{bmatrix}. \quad (14)$$

The solution of the equations can be (x_p, y_p) that minimizes the δ defined by the following

$$\delta = (Ax - b)^T (Ax - b) \quad (15)$$

$$x = \begin{bmatrix} x_p & y_p \end{bmatrix}^T. \quad (16)$$

Applying MMSE (Minimum Mean Square Error) method, we can obtain x with the following expression

$$x = (A^T A)^{-1} A^T b. \quad (17)$$

For more details on trilateration algorithm implementation, refer to our previous works in References [28,29].

3.3. LSTM Based UWB Localization

The UWB data is time dependent and LSTM model is the best deep learning model for UWB localization as compared to recurrent neural networks (RNN) [30], and extreme learning machine (ELM) [31] models. LSTM is the improved form of RNN and it is used to connect historical information to the current input. The long-term dependence problem of RNN makes them unsuitable for UWB-based localization systems and the unique long-time dependence characteristics of LSTM [32] gives accurate localization results when compared to other deep learning models. The LSTM result shows that it reduces the vanishing gradient problem [33] which exists when we use RNN networks for UWB localization. In this paper we followed the LSTM model presented in References [34,35] for deep LSTM implementation. The LSTM model consists of a layered structure with a state c (cell state) in each hidden layer. The output of the LSTM model is a regression output and not a classification one [36].

In a particular UWB indoor localization scenario, the number of anchors is N , the UWB localization could be estimated only with $N \in \{3, 4, 5, 6, \dots\}$. Assume that the localization system uses only one UWB tag, the distance received from anchor i , $i \in \{1, 2, 3, \dots, N\}$, can be written as $[D_{i-1}, D_{i-2}, \dots, D_{i-M_i}]$, where M_i is the distance from anchor i . The distance measurements from TOA-distance model converts into a fingerprint map and the fingerprint map in a certain location (x, y) is expressed as

$$D_{x,y} = \begin{bmatrix} D_{1-1}, D_{1-2}, \dots, D_{1-M_1}, \\ D_{2-1}, D_{2-2}, \dots, D_{2-M_2}, \\ \dots, \\ D_{N-1}, D_{N-2}, \dots, D_{N-M_N} \end{bmatrix}. \quad (18)$$

Let t denote the time interval of the current fingerprint, UWB distance data and $D_{(xt,yt)}, h_{(t-1)}$ and $c_{(t-1)}$ be the current network input values. The output of the LSTM network is expressed as [35]

$$h_t = [\tilde{x}_t, \tilde{y}_t], \quad (19)$$

where \tilde{x}_t and \tilde{y}_t are the coordinates of the predicted user position at a time t . The value of c_t is defined by the forget gate (FG) and input gate (IG) functions in the LSTM. The FG function saves the information of $c_{(t-1)}$ to c_t and IG saves the information c_t to h_t . The calculation process in FG is expressed as

$$f_t = \sigma \left(w_f \times [h_{t-1}, D_{xt,yt}] + b_f \right), \quad (20)$$

where σ is the rectified linear unit (ReLU), w_f is the weight matrix and b_f is the offset item. The calculation process in IG function is expressed as

$$g_t = \sigma(w_g \times [h_{t-1}, D_{xt,yt}] + b_g), \quad (21)$$

where w_g is the weight matrix of IG and b_g is the offset. The cell state in forward direction of LSTM is expressed as

$$\tilde{c}_t = \tanh(w_c \times [h_{t-1}, D_{xt,yt}] + b_c), \quad (22)$$

$$c_t = f_t \times c_{t-1} + g_t \times \tilde{c}_t, \quad (23)$$

where \tilde{c}_t and c_t are input and output of the cell state at time t . The long term message in LSTM network is formed from an iterative process and the output function (o_t) is expressed as

$$o_t = \sigma(w_o \times [h_{t-1}, D_{xt,yt}] + b_o), \quad (24)$$

where W_o and b_o are weight matrix and offset which are estimated from training process. The last stage of LSTM forward process is to estimate h_t from cell state and network and is defined as

$$h_t = o_t \times \tanh(c_t). \quad (25)$$

Figure 7 shows the overall forward propagation of the LSTM network. The output of this network is the user x and y position results $[\tilde{x}_t, \tilde{y}_t]$.

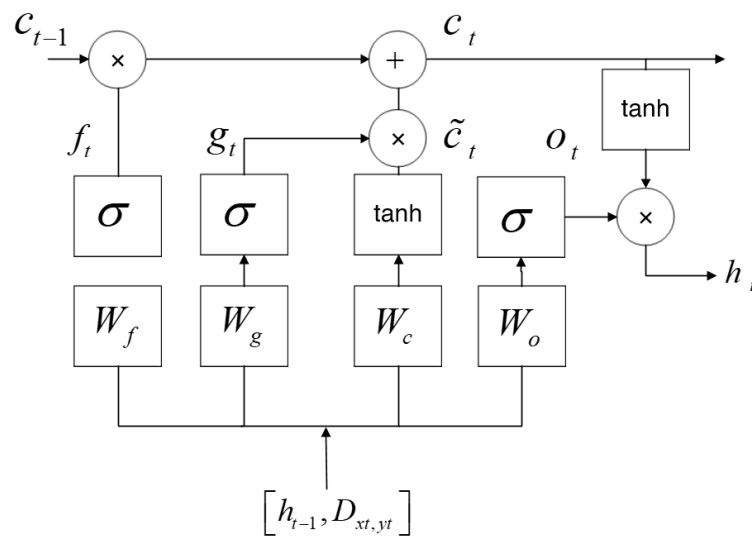


Figure 7. Forward propagation of long short-term memory (LSTM).

The error function from Figure 7 is expressed in time dimension and layer structure and is given by

$$E_t = \sqrt{(x_t - \tilde{x}_t)^2 + (y_t - \tilde{y}_t)^2}. \quad (26)$$

The input of LSTM network is the combination of $h_{(t-1)}$ and $D_{(xt,yt)}$, W_f , W_g , W_c and W_o . These values can be divided into $W_f x$, $W_f h$, $W_g x$, $W_g h$, $W_c h$, $W_f o x$ and $W_o h$. An example of backpropagation using $W_o h$ is expressed as

$$\frac{\partial E_t}{\partial W_{oh,t}} = \frac{\partial E_t}{\partial net_{o,t}} \times \frac{\partial net_{o,t}}{\partial W_{oh,t}} \quad (27)$$

$$\frac{\partial E_t}{\partial W_{oh,t}} = \delta_{o,t}^T \times h_{t-1}^T \quad (28)$$

$$\frac{\partial E_t}{\partial W_{oh,t}} = \delta_t^T \times \tanh(c_t) \times o_t \times (1 - o_t) \times h_{t-1}^T, \quad (29)$$

where δ_t^T is the error term in time dimension of layer o and is expressed as

$$\delta_{o,t}^T = \prod_{t_0}^{t-1} (\delta_{o,t}^T W_{oh} + \delta_{f,t}^T W_{fh} + \delta_{g,t}^T W_{gh} + \delta_{c,t}^T W_{ch}). \quad (30)$$

The LSTM network builds the regression model based on the iteration and the initial weight settings. The ReLu function in the model reduces the vanishing gradient problems and converges faster than RNN models. The localization algorithm using LSTM network is summarized in Algorithm 1.

Algorithm 1 LSTM Based Localization.

Input: $D_{(xt,yt)}$ in each time slot, M

output: Regression model based on LSTM

1. Collect the distance values from the trajectory when the user moves and repeat this step until the amount of data in this trajectory is enough.
 2. Generate the distance database in each time slot
 3. Initialize the structure of LSTM
 4. Train LSTM and compute the LSTM parameters
 5. Collect distance data for testing, then verify the trained LSTM model in step 4
 6. Change the LSTM parameters and estimate the best parameters for accurate localization using steps 5 and 6.
-

From Algorithm 1, it can be seen that the proposed approach uses the distance values from anchors to tag for localization. When the UWB system reaches the enough number of distance data from the simulation trajectory, the system initializes the two-layer LSTM model. The total amount of data collected from the trajectory is divided into train data and test data. The LSTM model uses the train data for training the model with initial hyperparameter settings. Next, the system changes the hyperparameter values by several simulations and reach the best LSTM hyperparameter values for localization. After the hyperparameter estimation, the system collects another set of data for testing. Using the test data, the system verifies the LSTM model and predicts the user positions.

Localization using the LSTM model is efficient when the user moves a certain trajectory in the experiment area. Figure 8 shows the LSTM structure used for the localization.

The model structure of LSTM from Figure 8 shows that the proposed model uses two LSTM layers after the input layer. The two LSTM layers improve the performance of our model and gives reasonable position results than conventional neural network models. To compensate the overfitting problem, we add a dropout layer after each LSTM layer and these dropout layers enhance the localization accuracy of our proposed model. For the last layer of the proposed network, we use a time distributed dense layer which enables an interaction between the values within its own timesteps.

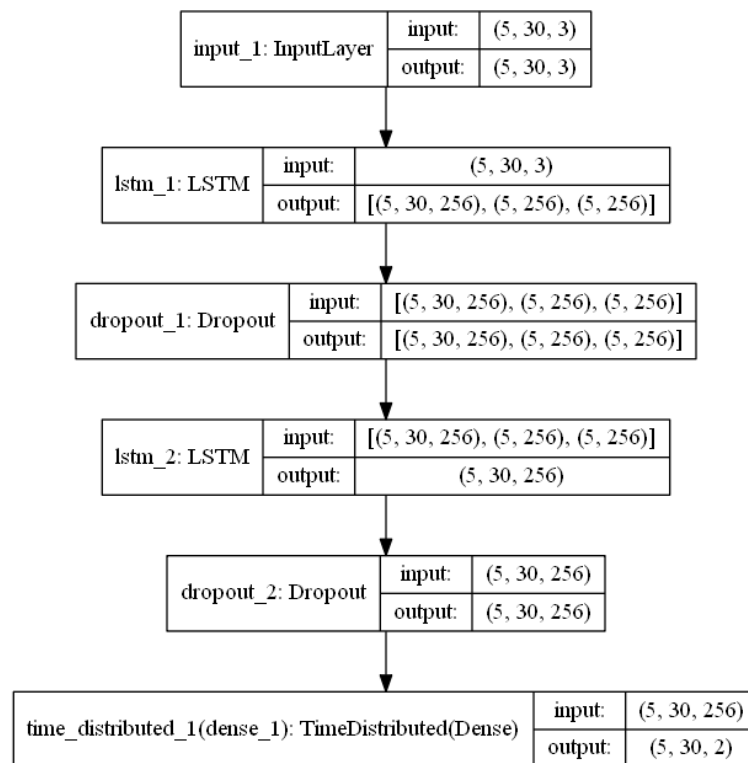


Figure 8. Model structure of proposed LSTM networks.

4. Simulation Results and Analysis

To evaluate the performance and position accuracy of our proposed LSTM model, we used the MATLAB R2019b (MathWorks, 1 Apple Hill Dr, Natick, MA 01760, United States, 2019) simulation environment with all built in tools. We simulated an experiment area with UWB anchors and a tag. Figure 9 indicates the UWB anchors and tag positions from a MATLAB simulation. In Figure 9, the starting position of the UWB tag is defined as (0, 2) and moved the UWB tag at a constant speed. We placed three anchors in the simulation environment based on the UWB tag motion. The positions of the anchors depend on the UWB tag motion and which covers the simulation environment. The UWB tag communicates with three anchors and estimates the tag distance from all anchors. The small circles in the Figure 9 show the anchor positions and the black star marks indicate the UWB tag motion. We moved the UWB tag in a random position based on a rectangular motion.

The TOA-distance model from the simulation estimates the distances for all tag positions in the experiment area and used the distances as training data. The data sample consists of 2582 samples with three distance and user x and y position values. Seventy-five percent of data samples were used for training the model and 25% samples used in testing. The LSTM model uses three input neurons for the three distance values from anchors and two output neurons to predict the user x and y positions. The performance of proposed LSTM model was analyzed in terms of learning rate, optimizer, loss function, batch size, number of hidden nodes and timesteps. Figure 10 shows the localization accuracy of training data with varies learning rates.

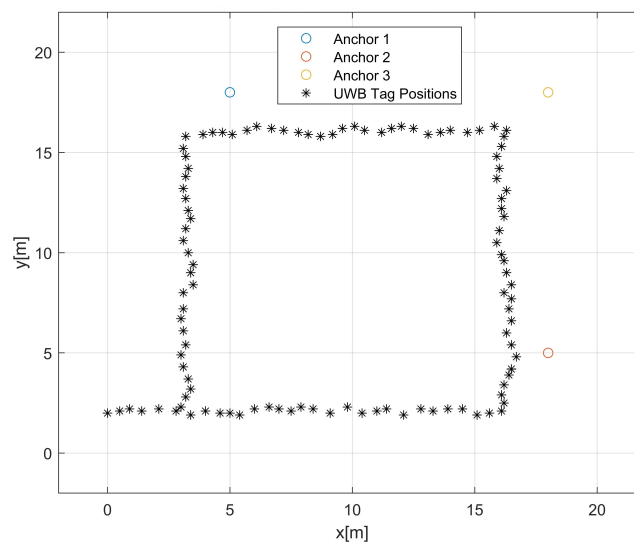


Figure 9. UWB anchors & tag positions.

From Figure 10, it can be seen that the accuracy of the system increases with a lower learning rate. For large learning rates, the model tends to be unstable during training and with very low rates, the system fails to train the model. To compensate for these issues, we simulated different learning rates and obtained the optimum learning rate for training the model through a process called hyperparameter tuning. In our simulation, we selected 0.0001 value as learning rate for training the model. The selected learning rate controls how much the model responds to the output error when the model weights are updated.

Next, we analyzed the significance of different optimizers for the LSTM model. The optimizers play a very crucial role to increase the accuracy of the model. When we consider deep neural networks for localization, the neural networks have the tendency to either vanish or explode as the energy is propagated through the function. When the function is more complex, the network performs worse due to the cumulative nature. To avoid the vanishing and exploding gradient issue, we use a suitable optimizer and it is necessary to identify the best optimizer for localization. In our simulation we analysed the performance of the stochastic gradient descent (SGD), Root Mean Square Propagation (RMSProp), adaptive gradient algorithm (Adagrad), adaptive moment estimation (Adam), nesterov-accelerated adaptive moment estimation (Nadam), Adadelata and Adamax. Figure 11 shows the performance of these different optimizers for the LSTM model.

From Figure 11, the Nadam optimizer gives best localization accuracy and a lower loss value when compared to other optimizers. The SGD and Adadelata shows the worst performance. The optimizers Adam, Adamax, Adagrad and RMSProp are very similar algorithms and RMSProp was found to slightly outperform the other optimizers.

Next, we consider the effect of different loss functions for the LSTM model. The loss function used in the deep leaning model reduces the error in the prediction. For the regression loss function, networks use mean squared error loss (MSE), mean squared logarithmic error loss (MSLE) and mean absolute error loss (MAE). To find the best loss function, we analysed the effect of different loss functions and Figure 12 shows their performance.

From Figure 12, the MSLE loss function shows a slightly better performance than MSE but much better performance than MAE loss function. However, the MSLE loss function has high possibilities to overfit and is commonly used to predict large values. The results from MSE are slightly similar to MSLE and it has a lower probability to overfit. The performance and convergence behavior of our model suggests that MSE loss function is a good match for training the LSTM network for our

regression problem. The results from the MAE loss function shows that it is not a good fit for our LSTM model.

Another important hyperparameter that affects the performance of the LSTM model is the batch size and it plays a major role in training the model faster. The batch size controls the amount of data samples that will be propagated through the LSTM networks. If we use a larger batch size to train the model, it increases the computational speed of the training data. However, a large batch size leads to poor generalization and the learning process converges slowly. Figure 13 shows the localization accuracy and network loss of training data in terms of the batch size.

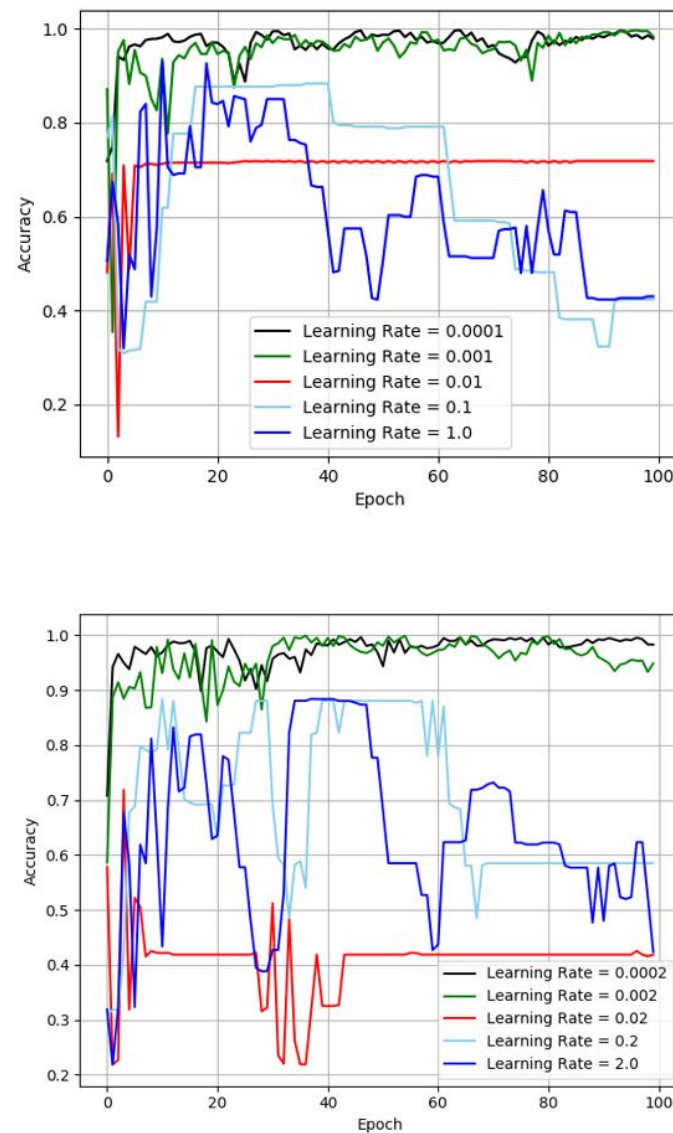


Figure 10. Localization accuracy of training data with different learning rates.

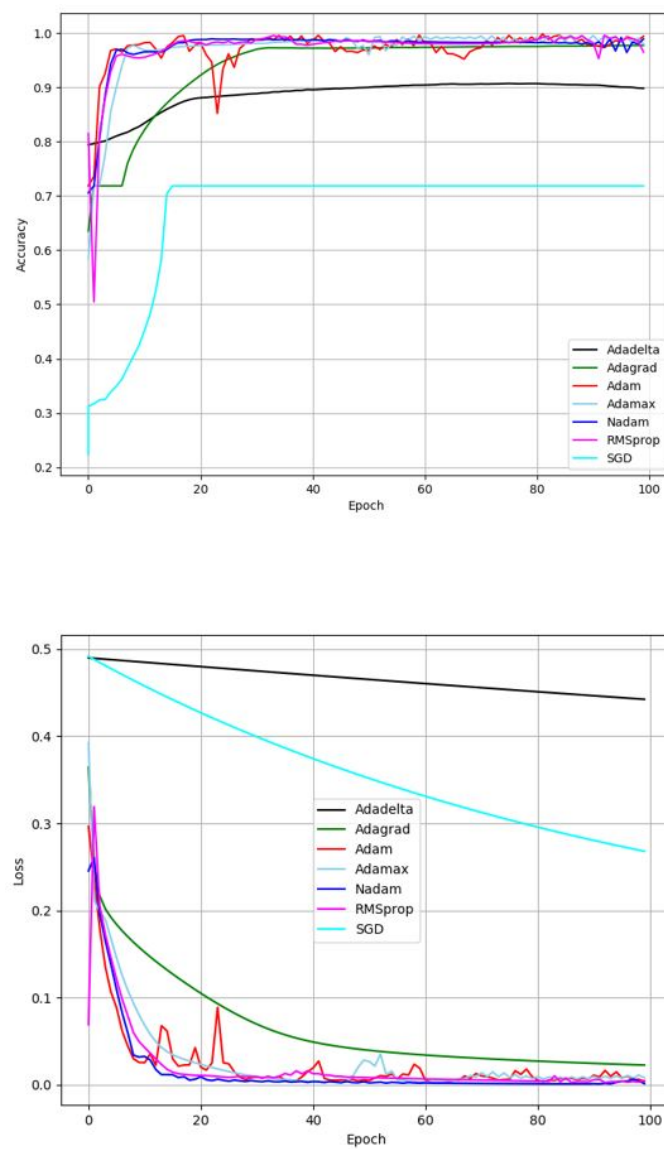


Figure 11. Performance of different optimizers for the LSTM model.

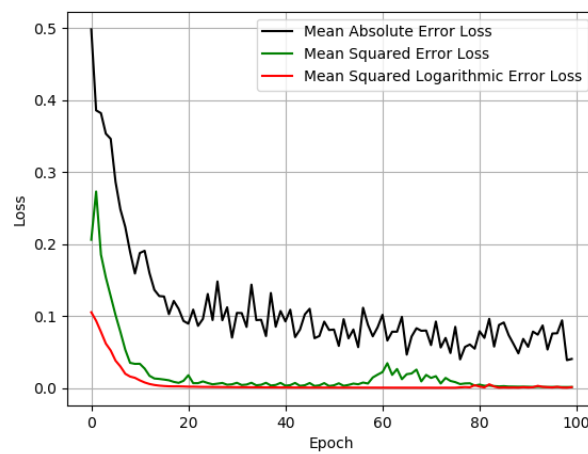


Figure 12. Performance of different loss functions.

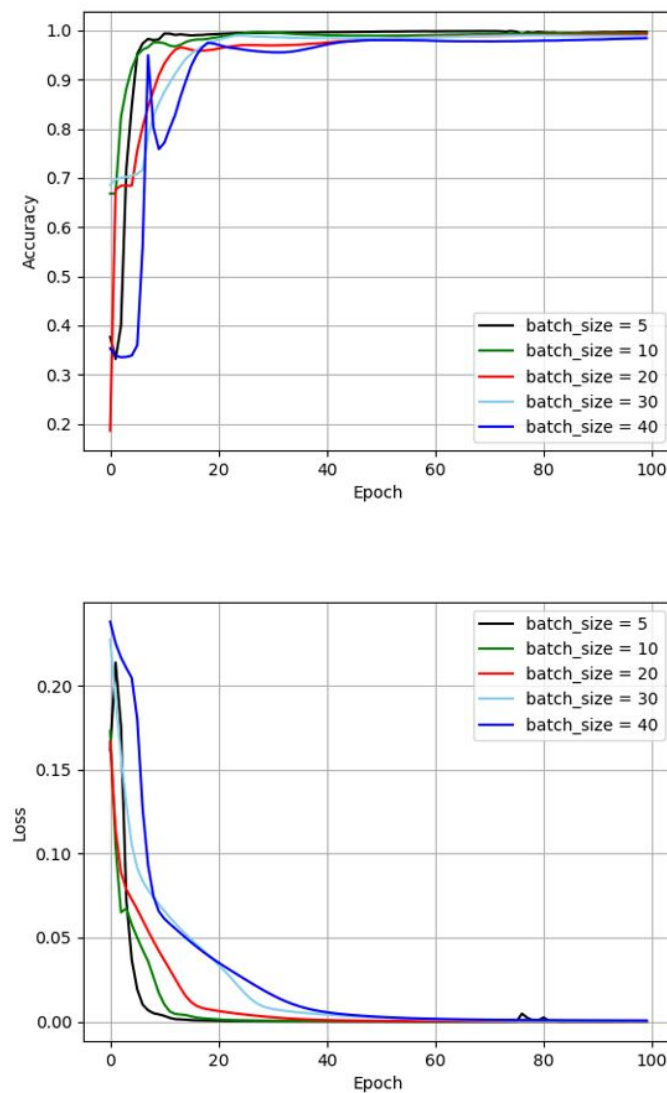


Figure 13. Localization accuracy and network loss of training data in terms of number of batch size.

From Figure 13, it can be seen that the lower batch size values give a learning process that converges quickly. The lower batch size value also uses less memory for training. Since the network is trained by fewer samples, the overall training procedure requires less memory. However, the lower value of batch size is not guaranteed to converge to the global optima. In our simulation, we used the batch size as 5 without any computational constraints.

The performance of LSTM model also depends on the number of hidden nodes. The number of hidden nodes depend on the complexity of the dataset, number of input features, number of data points and the data generating process. To find the optimum number of hidden nodes without overfitting, the following expression is used.

$$N_h = \frac{N_s}{(\alpha \times (N_i + N_o))}, \quad (31)$$

where N_i is the number of input neurons, N_o is the output neurons, N_s is the number of samples in training data and α is the arbitrary scaling factor usually 2–10. Figure 14 shows the localization accuracy and network loss of training data in terms of number of hidden nodes.

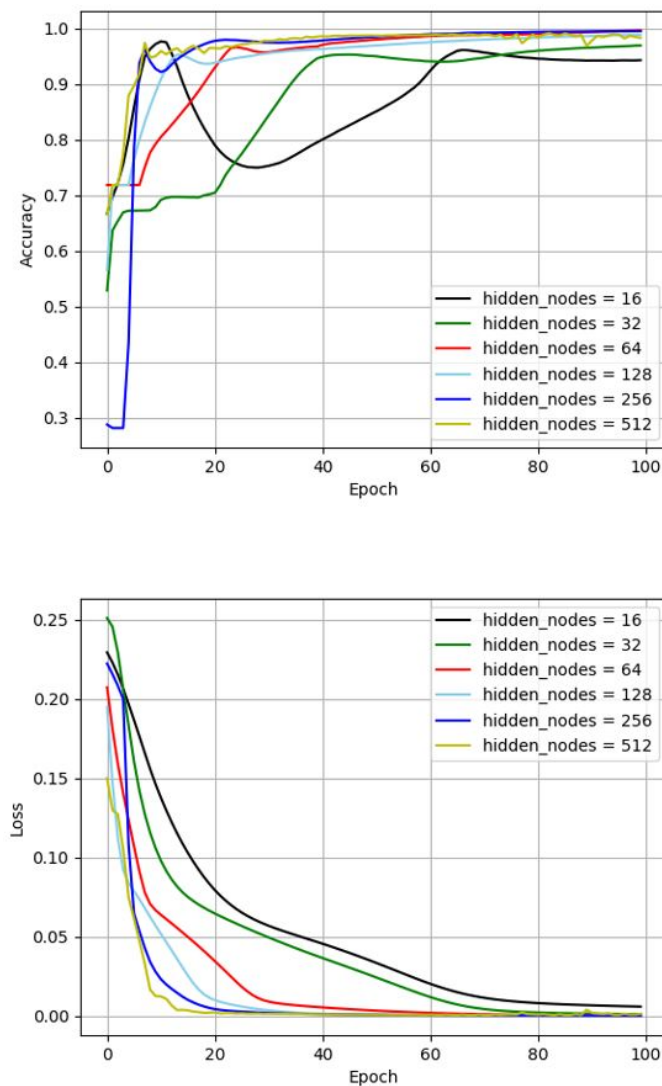


Figure 14. Localization accuracy and network loss of training data in terms of number of hidden node.

In Figure 14, the localization accuracy during the training in terms of epochs with varying hidden nodes from 16 to 512 with batch size of 5 is shown. If we use few hidden nodes, the result shows a high training error and high generalization error due to underfitting. On the other hand, for the high number of hidden nodes, the result shows low training error and high generalization error due to over fitting. Through our several simulation results, we obtained the number of hidden node as 256.

The localization accuracy of training data for different timesteps also affects the performance of LSTM model. It is necessary to analyse the performance of different timesteps for better localization accuracy. The overall performance of the model increases with increase in timesteps. The timestep is equivalent to the amount of timestep runs the LSTM model. Figure 15 shows the localization accuracy and network loss of training data in terms of timestep.

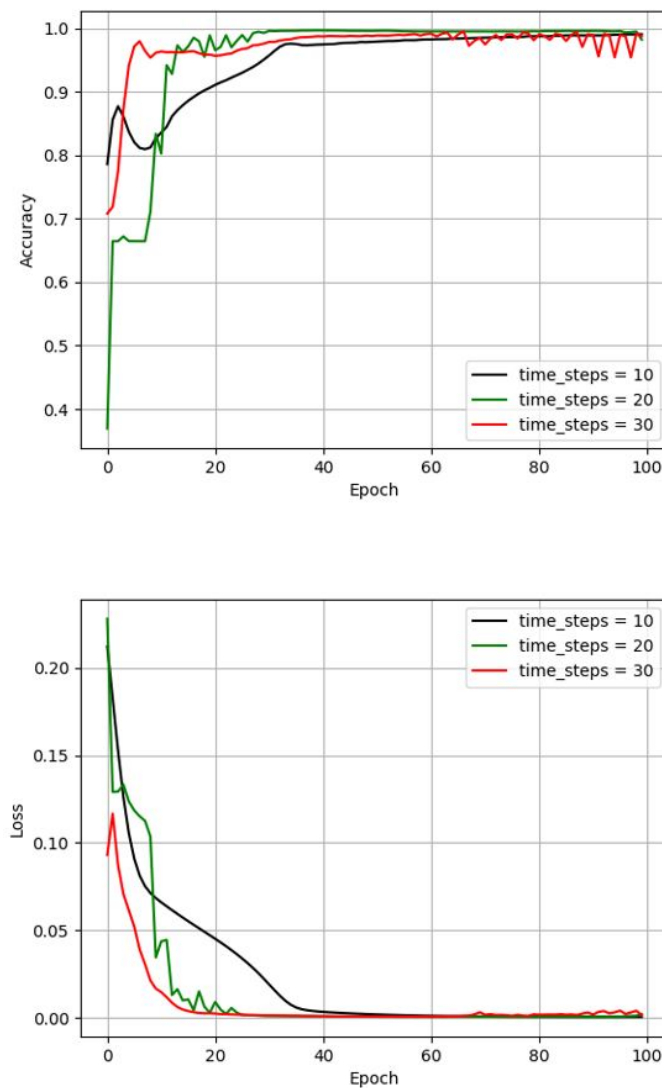


Figure 15. Localization accuracy of training data in terms of number of timesteps.

From Figure 15, it is clear that the higher timestep gives better accuracy and lower network loss. When we increase the timestep value from very low to high, the network converges quickly without any overfitting problems. From all the simulation results and analysis, we obtained the best LSTM hyperparameters for position prediction. Table 2 shows the optimized LSTM parameters based on the training data.

Table 2. LSTM parameter values.

LSTM Parameter	Value
Ratio of training data to overall data	0.75
Number of epochs	100
Batch size	5
Timesteps	30
Hidden nodes	256
Optimizer	Nadam
Loss	Mean squared error (MSE)
Dropout Rate	0.2

The Table 2 configuration of LSTM parameters are used in the proposed approach for predicting the current user position. To validate the performance of the proposed LSTM based localization, we estimated the mean localization error from average localization error (E) results. The estimation of E is expressed as

$$E = \sum_{i=1}^L \left((x_i^{true} - x_i^{est})^2 + (y_i^{true} - y_i^{est})^2 \right)^{0.5}, \quad (32)$$

where (x_i^{true}, y_i^{true}) is the actual UWB tag position and (x_i^{est}, y_i^{est}) denotes the estimated UWB tag position calculated by proposed localization approach. The L value indicates the total number of data samples used for the localization.

The performance of the proposed LSTM approach evaluated with a number of LSTM layers, training and testing time of the model and mean localization error with different number of UWB anchors. Figure 16 shows the mean localization errors with different number of LSTM layers.

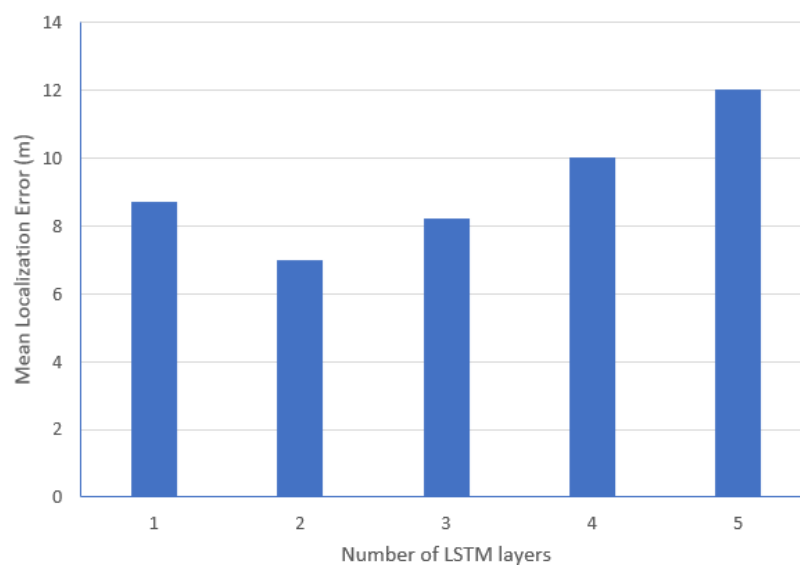


Figure 16. Mean localization errors with different number of LSTM layers.

From Figure 16, the proposed two-layer LSTM approach shows better performance than standard LSTM (one LSTM layer) and a higher number of LSTM layers. The results show that if we increase the number of LSTM layers, the UWB system shows worst performances. This is due to the overfitting of the networks with more layers. The proper selection of the number of LSTM layers is an influencing factor of the localization performances and our system used the two-layer LSTM network for better performances.

The training and testing time of the proposed model is an important parameter for real time indoor localization. The training and testing time of all models are summarized in Table 3.

Table 3. Training and testing time of all models.

Time	BP	ELM	LSTM	RNN
Training time (sec)	740	520	1080	900
Testing time (sec)	0.287	0.215	0.795	0.321

From Table 3, the training time of the proposed approach is still much higher than the approaches of BP, ELM and RNN. However, when we consider the accuracy of the localization system, the proposed LSTM approach gives acceptable localization accuracy with reasonable computational time. The testing time of the proposed approach with all testing samples is acceptable for real time indoor localization.

To investigate the impact of the number of anchors for UWB localization for all the approaches, we perform an additional simulation with the data from different number of anchors in the simulation environment. Figure 17 shows the simulation results with the number of anchors from 3 to 5 for all approaches.

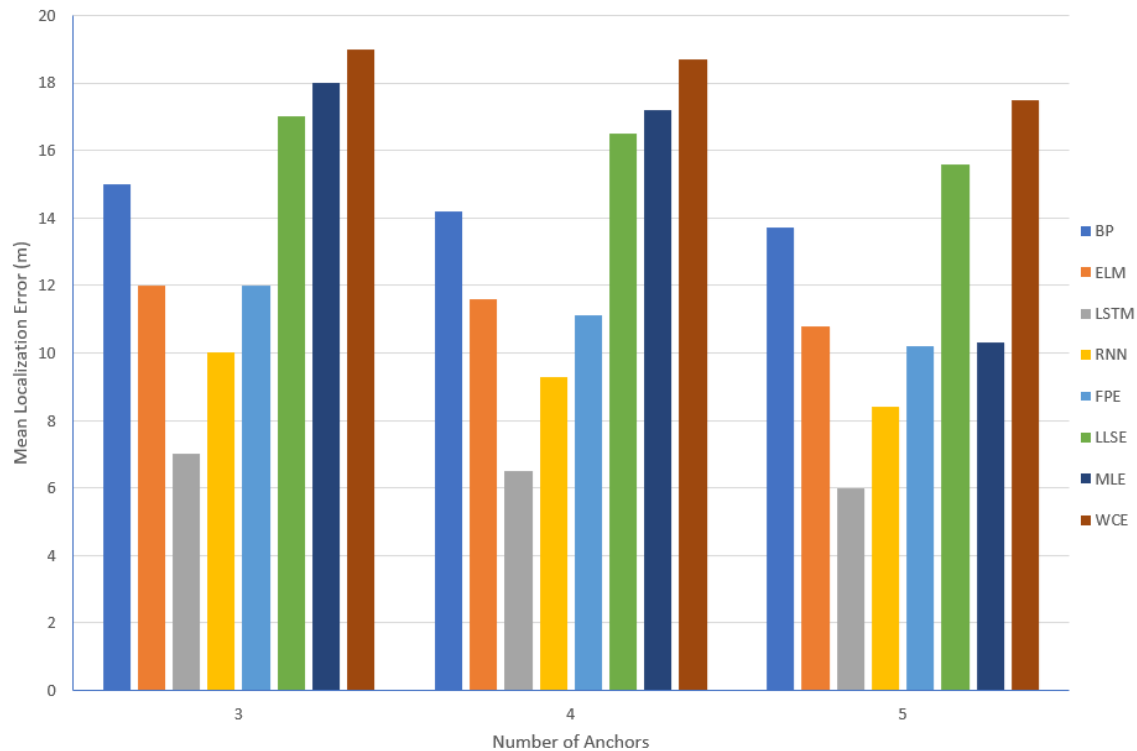


Figure 17. Mean localization errors with different number of anchors for all the approaches.

From Figure 17, the mean localization error decreases with increase the number of anchors. The mean localization error results indicate that the proposed approach achieves the best localization performance in all situations. The results from RNN are better than ELM and BP models. However, the ELM model gives faster learning speed, better generalization performance and with least human intervention when compared to BP model. The results from BP model are not suitable for UWB indoor localization when we use multiple anchors. In the case of conventional localization approaches, the FPE algorithm shows less localization error than other conventional approaches. However, localization using fingerprint maps is not a common approach in UWB localization systems due to the anchors and tag distance limitation. The UWB anchors and tag should maintain minimum distances for accurate localization. The most common conventional UWB localization approach is the LLSE and which gives accurate position results than the MLE and WCE approaches. The WCE approach has a high localization error and it is not suitable for our simulation scenario. From all the simulation results and analysis, the proposed LSTM approach outperforms conventional localization approaches and gives better localization accuracy for UWB localization.

5. Conclusions

In this paper, we presented a UWB indoor localization system using a deep learning approach. The proposed method uses two LSTM networks and predicts the user position using anchor-tag distance information. The performance of the proposed LSTM model is analyzed on the basis of the learning rate, optimizer, loss function, batch size, hidden nodes and timesteps. Through analyzing several simulation results, we estimated the best LSTM hyperparameters for user position prediction. We also analyzed the LSTM model performance in terms of mean localization error with different

number of LSTM layers, training and testing time of UWB data in LSTM model and the mean localization error with different number of anchors. From the simulation results, the proposed method shows reasonable position results for UWB based indoor localization. The proposed approach solved the LOS problems associated UWB system and gives a better localization accuracy than conventional localization approaches. In the future work, we intend to implement a CNN-LSTM model for predicting the current user position.

Author Contributions: Writing—original draft, A.P.; Writing—review & editing, D.S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2016-0-00564, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding) and Samsung Electronics' University R&D Program. [MIMO System Architecture and Algorithm Development for 60 GHz Band Next Generation WLAN (SLSI-201507DD013)].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alarifi, A.; Al-Salman, A.; Alsaleh, M.; Alnafessah, A.; Al-Hadhrami, S.; Al-Ammar, M.A.; Al-Khalifa, H.S. Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors* **2016**, *16*, 707. [CrossRef] [PubMed]
2. Navarro, M.; Prior, S.; Najar, M. In Low complexity frequency domain TOA estimation for IR-UWB communications. In Proceedings of the IEEE Vehicular Technology Conference, Montreal, QC, Canada, 25–28 September 2006; pp. 1–5.
3. Chóliz, J.; Eguizábal, M.; Hernández-Solana, Á.; Valdovinos, A. Comparison of algorithms for uwb indoor location and tracking systems. In Proceedings of the 2011 IEEE 73rd Vehicular Technology Conference (VTC Spring), Yokohama, Japan, 15–18 May 2011; pp. 1–5.
4. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
5. Joung, J.; Jung, S.; Chung, S.; Jeong, E.-R. CNN-based Tx–Rx distance estimation for UWB system localisation. *Electron. Lett.* **2019**, *55*, 938–940. [CrossRef]
6. Bregar, K.; Mohorčič, M. Improving indoor localization using convolutional neural networks on computationally restricted devices. *IEEE Access* **2018**, *6*, 17429–17441. [CrossRef]
7. Luo, J.; Gao, H. Deep belief networks for fingerprinting indoor localization using ultrawideband technology. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 5840916. [CrossRef]
8. Taok, A.; Kandil, N.; Affes, S. Neural Networks for Fingerprinting-Based Indoor Localization Using Ultra-Wideband. *JCM* **2009**, *4*, 267–275. [CrossRef]
9. Poulouse, A.; Eyobu, O.S.; Kim, M.; Han, D.S. Localization Error Analysis of Indoor Positioning System Based on UWB Measurements. In Proceedings of the 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), Zagreb, Croatia, 2–5 July 2019; pp. 84–88.
10. Alsindi, N.A.; Alavi, B.; Pahlavan, K. Measurement and modeling of ultrawideband TOA-based ranging in indoor multipath environments. *IEEE Trans. Veh. Technol.* **2008**, *58*, 1046–1058. [CrossRef]
11. Tiemann, J.; Eckermann, F.; Wietfeld, C. Atlas—an open-source tdoa-based ultra-wideband localization system. In Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcalá de Henares, Spain, 4–7 October 2016; pp. 1–6.
12. Povalač, A.; Šebesta, J. Phase of arrival ranging method for UHF RFID tags using instantaneous frequency measurement. In Proceedings of the 20th International Conference on Applied Electromagnetics and Communications, Dubrovnik, Croatia, 20–23 September 2010; pp. 1–4.
13. Zhang, Y.; Brown, A.K.; Malik, W.Q.; Edwards, D.J. High resolution 3-D angle of arrival determination for indoor UWB multipath propagation. *IEEE Trans. Wirel. Commun.* **2008**, *7*, 3047–3055. [CrossRef]
14. Taponecco, L.; D'Amico, A.A.; Mengali, U. Joint TOA and AOA estimation for UWB localization applications. *IEEE Trans. Wirel. Commun.* **2011**, *10*, 2207–2217. [CrossRef]
15. Nilsson, M. Performance Comparison of Localization Algorithms for UWB Measurements with Closely Spaced Anchors. 2018. Available online: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1251202&dswid=-5043> (accessed on 7 February 2020).

16. Sookyoi, T. Experimental Analysis of Indoor Positioning System Based on Ultra-Wideband Measurements. 2016. Available online: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1055816&dsid=5375> (accessed on 20 January 2020).
17. Zhang, L.; Li, Y.; Gu, Y.; Yang, W. An efficient machine learning approach for indoor localization. *China Commun.* **2017**, *14*, 141–150. [[CrossRef](#)]
18. Kristensen, J.B.; Ginard, M.M.; Jensen, O.K.; Shen, M. Non-Line-of-Sight Identification for UWB Indoor Positioning Systems using Support Vector Machines. In Proceedings of the IEEE MTT-S International Wireless Symposium (IWS), Guangzhou, China, 19–22 May 2019; pp. 1–3.
19. Kołakowski, M.; Modelski, J. Detection of Direct Path Component Absence in NLOS UWB Channel. In Proceedings of the 22nd International Microwave and Radar Conference (MIKON), Poznan, Poland, 15–17 May 2018.
20. Zandian, R.; Witkowski, U. Differential NLOS Error Detection in UWB-based Localization Systems using Logistic Regression. In Proceedings of the 15th Workshop on Positioning, Navigation and Communications (WPNC), Bremen, Germany, 25–26 October 2018; pp. 1–6.
21. Marano, S.; Gifford, W.M.; Wymeersch, H.; Win, M.Z. NLOS Identification and Mitigation for Localization Based on UWB Experimental Data. *IEEE J. Sel. Areas Commun.* **2010**, *28*, 1026–1035. [[CrossRef](#)]
22. Niitsoo, A.; Edelh  user, T.; Mutschler, C. Convolutional neural networks for position estimation in tdoa-based locating systems. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Nantes, France, 24–27 September 2018; pp. 1–8.
23. Hsieh, C.-H.; Chen, J.-Y.; Nien, B.-H. Deep learning-based indoor localization using received signal strength and channel state information. *IEEE Access* **2019**, *7*, 33256–33267. [[CrossRef](#)]
24. Meghani, S.K.; Asif, M.; Awin, F.; Tepe, K. Empirical Based Ranging Error Mitigation in IR-UWB: A Fuzzy Approach. *IEEE Access* **2019**, *7*, 33686–33697. [[CrossRef](#)]
25. Xue, Y.; Su, W.; Wang, H.; Yang, D.; Jiang, Y. DeepTAL: Deep Learning for TDOA-Based Asynchronous Localization Security With Measurement Error and Missing Data. *IEEE Access* **2019**, *7*, 122492–122502. [[CrossRef](#)]
26. Khodjaev, J.; Narzullaev, A.; Park, Y.; Jung, W.; Lee, J.; Kim, S. Performance improvement of asynchronous UWB position location algorithm using multiple pulse transmission. In Proceedings of the 4th Workshop on Positioning, Navigation and Communication, Hannover, Germany, 22–22 March 2007; pp. 167–170.
27. Molisch, A.F.; Balakrishnan, K.; Chong, C.-C.; Emami, S.; Fort, A.; Karedal, J.; Kunisch, J.; Schantz, H.; Schuster, U.; Siwiak, K. IEEE 802.15. 4a channel model-final report. *IEEE P802* **2004**, *15*, 0662.
28. Poulou, A.; Kim, J.; Han, D.S. A sensor fusion framework for indoor localization using smartphone sensors and Wi-Fi RSSI measurements. *Appl. Sci.* **2019**, *9*, 4379. [[CrossRef](#)]
29. Poulou, A.; Eyobu, O.S.; Han, D.S. A combined PDR and Wi-Fi trilateration algorithm for indoor localization. In Proceedings of the International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), Okinawa, Japan, 11–13 February 2019; pp. 072–077.
30. Karim, F.; Majumdar, S.; Darabi, H.; Chen, S. LSTM fully convolutional networks for time series classification. *IEEE Access* **2017**, *6*, 1662–1669. [[CrossRef](#)]
31. Zhang, J.; Xiao, W.; Zhang, S.; Huang, S. Device-free localization via an extreme learning machine with parameterized geometrical feature extraction. *Sensors* **2017**, *17*, 879. [[CrossRef](#)]
32. Greff, K.; Srivastava, R.K.; Koutn  k, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2222–2232. [[CrossRef](#)]
33. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
34. Zhong, Z.; Tang, Z.; Li, X.; Yuan, T.; Yang, Y.; Wei, M.; Zhang, Y.; Sheng, R.; Grant, N.; Ling, C. XJTUIIndoorLoc: A new fingerprinting database for indoor localization and trajectory estimation based on Wi-Fi RSS and geomagnetic field. In Proceedings of the Sixth International Symposium on Computing and Networking Workshops (CANDARW), Takayama, Japan, 27–30 November 2018; pp. 228–234.

35. Xu, B.; Zhu, X.; Zhu, H. An efficient indoor localization method based on the long short-term memory recurrent neuron network. *IEEE Access* **2019**, *7*, 123912–123921. [[CrossRef](#)]
36. Dietterich, T.G. Ensemble methods in machine learning, International workshop on multiple classifier systems. In Proceedings of the First International Workshop on Multiple Classifier Systems (MCS 2000), Cagliari, Cagliari, Italy, 21–23 June 2000; pp. 1–15.



© 2020 by the authors. 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/>).