Article

Attention Mechanism and LSTM Network for Fingerprint-Based Indoor Location System

Zhen Wu * , Peng Hu , Shuangyue Liu , Tao Pang

Posted Date: 24 January 2024

doi: 10.20944/preprints202401.1768.v1

Keywords: fingerprinting; indoor localization system; long short-term memory (LSTM); self-attention mechanism

Article

Attention Mechanism and LSTM Network for Fingerprint-Based Indoor Location System

Zhen Wu *, Peng Hu, Shuanhgyue Liu and Tao Pang

Department of Mobile Communications and Terminal Research, China Telecom Research Institute, Guangzhou, China

* Correspondence: wuz16@chinatelecom.cn

Abstract: The demand for precise indoor localization services is steadily increasing. Among various methods, fingerprint-based indoor localization has become a popular choice due to its exceptional accuracy, cost-effectiveness, and ease of implementation. However, its performance degrades significantly as a result of multipath signal attenuation and environmental changes. In this paper, we propose an indoor localization method based on fingerprints using self-attention and long short-term memory (LSTM). By integrating a self-attention mechanism and LSTM network, the proposed method exhibits outstanding positioning accuracy and robustness in diverse experimental environments. The performance of the proposed method is evaluated under two different experimental scenarios, which involve 2D and 3D moving trajectories, respectively. The experimental results demonstrate that our approach achieves an average localization error of 1.76 m and 2.83 m in the respective scenarios, outperforming the existing state-of-the-art methods by 42.67% and 31.64%.

Keywords: fingerprinting; indoor localization system; long short-term memory (LSTM); self-attention mechanism

1. Introduction

The rapid development of global digitization has created a high demand for location-based services (LBS) in many industries [1]. These services have become essential for various systems and applications, including transportation, logistics, emergency response, etc. In outdoor environments, mobile users already have access to established outdoor positioning technologies such as the Global Positioning System (GPS) [2] and the BeiDou Satellite Navigation System (BDS) [3] to obtain accurate location information. However, the effectiveness of these technologies is often limited in indoor environments due to the scattering and attenuation effects of satellite signals.

In the field of indoor localization, various wireless signals have been proposed and utilized, including WiFi [4–7], Bluetooth [8,9], Ultra-Wide Bandwidth (UWB) [10,11], Radio Frequency Identification (RFID) [12], and custom radios [13]. Typical ranging-based methods for processing wireless signals in indoor localization involve using information such as Angle of Arrival (AOA) or Time of Arrival (TOA) to estimate the specific positions of the user equipment (UE) [14]. However, these methods require prior knowledge of the locations of access points (APs) and are susceptible to errors in the distance measurement between the UE and APs, which can negatively impact the accuracy of the positioning. In contrast to these methods, the fingerprint-based indoor localization method is characterized by simplicity and efficiency [15]. This technique relies on the unique characteristics of wireless signals in indoor environments to create a map or "fingerprint" of the Received Signal Strength Indicator (RSSI) at different locations. The fingerprint can then be used to estimate the position of the UE based on the signal strengths measured at that location. Fingerprint-based methods are highly accurate and can offer sub-meter-level positioning accuracy in many cases, making them a promising alternative to ranging-based methods. However, in the context of fingerprint-based methods, the radio propagation environment introduces multi-path effects, shadowing, signal fading, and other forms of signal degradation and distortion leads to a significant fluctuations in RSSI values. In the experiments described in this paper, the observed RSSI values for different APs at a fixed location exhibit a wide range of fluctuations, as illustrated in Figure 1. The fluctuation in RSSI makes it challenging to discern the pattern of RSSI between the test points (TPs) and reference points (RPs), thereby significantly impacting the accuracy of positioning.

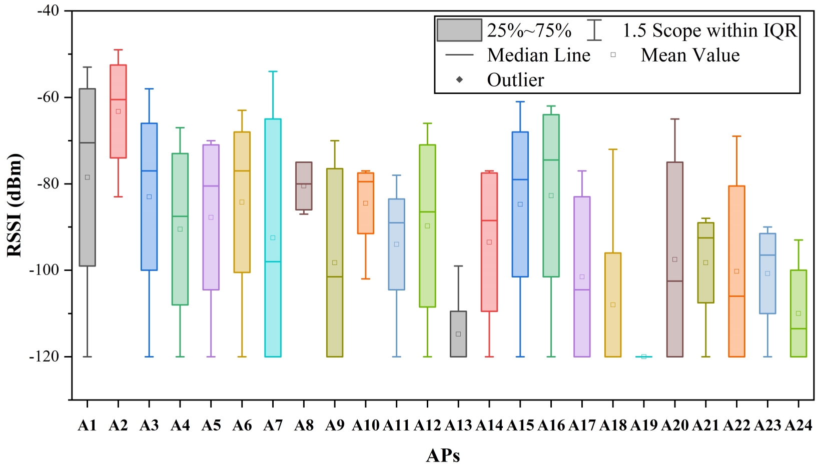


Figure 1. The range of variation in RSSI for APs observed at a single location.

With the development of machine learning algorithms over the past few decades, numerous machine learning algorithms have been proven to be effective in recognizing the RSSI pattern [16]. M. Brunato et al. proposed applying Support Vector Machines (SVM) in location fingerprint positioning systems [17]. Hoang et al. introduced a soft range-limited k-Nearest Neighbors (KNN) fingerprinting algorithm that addresses spatial ambiguity in localization by scaling the fingerprint distance with a range factor based on the physical distance between the previous position of users and the reference location in the database [18]. Fang et al. utilized Feedforward Neural Networks (FNN) to extract fingerprint features from the RSSI, enabling accurate localization of the actual position [19]. However, the performance of these algorithms can easily be limited when learning features in complex indoor environments. To achieve superior performance, some research studies have suggested using Long Short-Term Memory (LSTM) for handling sequential trajectory prediction in indoor localization systems [4,20,21], which has been experimentally demonstrated to be more effective than the conventional KNN method. Meanwhile, self-attention has been proposed as a promising technique for enhancing the performance of sequence processing tasks [22–25]. By enabling the model to attend to various regions of the input sequence, self-attention improves its capacity to capture the connections between various features in a sequence.

This paper introduces a novel method named SA-LSTM (Self-Attention and LSTM) that effectively improves positioning accuracy and robustness. We conducted experiments in two different scenarios to validate the effectiveness and robustness of the proposed approach. The experimental results demonstrate that SA-LSTM exhibits greater robustness and higher accuracy in indoor localization compared to some of the most advanced algorithms.

The main contributions of this paper are as follows:

1. We propose a novel SA-LSTM model that integrates the self-attention mechanism and LSTM networks. SA-LSTM treats the localization problem as a sequence learning task. It processes the RSSI values of consecutive time instances and predicts the position at the final moment in the input sequence. The self-attention mechanism enables the LSTM to more effectively capture the interdependencies between the RSSI values at different time instances, thereby facilitating improved extraction of location information and reducing the localization error.
2. We validate the performance of the proposed SA-LSTM model in two distinct experimental environments. The first experiment scenario involves collecting Bluetooth RSSI data while

moving in 2D trajectories on a specific floor. In the second experiment, we used an open-source WiFi RSSI dataset containing 3D-moving trajectories across various floors within a building.

3. We conduct a comparative analysis between our proposed model and several state-of-the-art methods. The experimental results reveal that our proposed SA-LSTM model achieves the highest localization accuracy in both experimental scenarios, demonstrating its robustness and precision.

The rest of this paper is structured as follows. Section II provides an overview of related works in the area of fingerprint indoor localization systems. Section III presents the technical details of our proposed model. Section IV outlines the experimental setup utilized in our study. Section V presents and analyzes the experimental results obtained from various datasets. Finally, Section VI offers concluding remarks and outlines our future research plans.

2. Related Work

In this section, we present an overview of the existing research on fingerprint-based indoor localization and the application of self-attention mechanisms.

The authors in [26] proposed applying the KNN algorithm for the first time in the field of fingerprint-based indoor localization. According to the article, the RSSI values from multiple base stations were recorded and processed as reference points stored in a database. During the testing phase, the positions of testing points were determined using the Euclidean distance. On average, the system achieved an accuracy of approximately 3 m, with 75% of localization errors falling below 4.7 m.

An improved version of the KNN method for indoor localization is the weighted KNN (WKNN), which was introduced by Brunato and Battiti [27]. In that paper, the positions of users are determined by calculating the weighted average of the RSSI distances between the estimated nearest neighbors and the current measurement. Tests performed in a real-world environment showed that the WKNN method achieved an accuracy of 3.1 ± 0.1 m, with the added benefit of low algorithmic complexity.

Yerbolat Khassanov et al. explored the use of end-to-end sequence models for WiFi-based indoor localization at a finer level [4]. The study showed that the localization task can be effectively formulated as a sequence learning problem using Recurrent Neural Networks (RNN) with regression output. The use of regression output allows for estimating three-dimensional positions and enables scalability to larger areas. The experiments conducted on the WiFine dataset reveal that RNN models outperform non-sequential models such as KNN and FNN, achieving an average positioning error of 3.05 m for finer-level localization tasks.

Furthermore, Zhenghua Chen et al. proposed a deep LSTM network for indoor localization using WiFi fingerprinting [28]. The network incorporates a local feature extractor that enables the encoding of temporal dependencies and the learning of high-level representations based on the extracted sequential local features. The experimental results demonstrate that the proposed approach achieves state-of-the-art localization performance, with mean localization errors of 1.48 m and 1.75 m in research lab and office environments, respectively.

To address neural machine translation tasks, Bahdanau et al. introduced the attention mechanism to the encoder-decoder model. This enables the model to learn alignment and translation simultaneously, allowing for adaptive selection of encoded vectors [29]. The proposed approach exhibits substantial improvements in translation performance compared to the basic encoder-decoder approach, especially with longer sentences. Furthermore, an LSTM structure based on self-attention mechanism was introduced in [30]. The overall results demonstrate the superiority of the proposed method in forecasting temporal sequences compared to other benchmark methods.

In general, LSTM has demonstrated exceptional performance in sequence prediction tasks, including fingerprint localization. It has been experimentally verified that it outperforms conventional methods such as KNN and WKNN. Additionally, the self-attention mechanism enables the model to consider the relationship between each element in the sequence. This leads to a better understanding of contextual information and more precise processing of sequence data. Based on that, we propose an

SA-LSTM model with high accuracy and strong robustness for indoor localization systems based on fingerprinting.

3. Methodology

In this section, we will begin by introducing the framework of the SA-LSTM-based localization algorithm. Subsequently, we will provide detailed introductions to the working principles of its subcomponents.

3.1. SA-LSTM based Localization Algorithm

Figure 2 illustrates the framework of the SA-LSTM-based localization algorithm, comprising an offline training stage and an online estimation stage. During the offline training stage, the RSSI values collected at different points and their corresponding coordinates of locations are recorded and stored in the fingerprint database. Subsequently, the collected RSSI data are normalized and used to train the SA-LSTM network. The trainable weights of the SA-LSTM network will be updated to minimize the loss between the output and the ground true locations. The trainable weights of the SA-LSTM network are adjusted to minimize the loss between the output and the actual locations. During the online estimation stage, real-time RSSI data from the device is normalized and input into the trained SA-LSTM model, which then generates real-time location estimates.

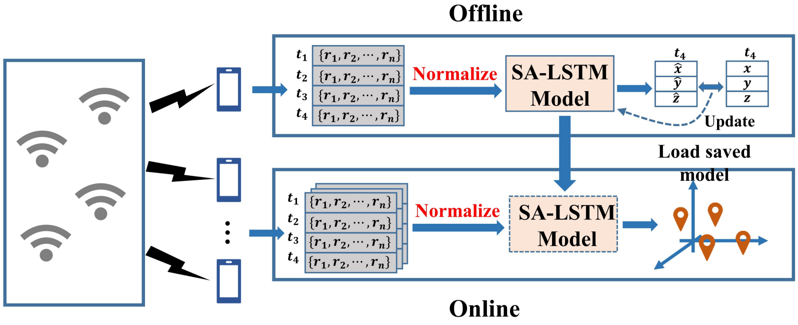

Offline
t_1 $\{r_1, r_2, \cdots, r_n\}$
t_2 $\{r_1, r_2, \cdots, r_n\}$
t_3 $\{r_1, r_2, \cdots, r_n\}$
t_4 $\{r_1, r_2, \cdots, r_n\}$
Normalize
SA-LSTM Model
t_4 $\hat{x}$ $\hat{y}$ $\hat{z}$
t_4 x y z
Update
Load saved model
t_1 $\{r_1, r_2, \cdots, r_n\}$
t_2 $\{r_1, r_2, \cdots, r_n\}$
t_3 $\{r_1, r_2, \cdots, r_n\}$
t_4 $\{r_1, r_2, \cdots, r_n\}$
Normalize
SA-LSTM Model
Online

Figure 2. The generation of the attention value.

3.2. LSTM Network

LSTM is a unique form of recurrent neural network that has been extensively researched in deep learning. In contrast to conventional RNN, the LSTM network introduces gated states to modulate the flow of information, thereby enabling it to selectively retain relevant information over extended periods while filtering out irrelevant data, which allows it to effectively analyze the long temporal sequences.

Figure 3 shows the common architecture of LSTM, which is composed by connecting memory units. In this context, $\boldsymbol{C_t}$ and $\boldsymbol{H_t}$ represent the unit state and hidden state at time t, respectively. Focus on the time t, the memory unit receives the $\boldsymbol{C_{t-1}}$ and $\boldsymbol{H_{t-1}}$ from the previous memory unit, as well as the current input value $\boldsymbol{x_t}$. After performing internal arithmetic operations, the unit generates the updated cell state $\boldsymbol{C_t}$ and hidden state $\boldsymbol{H_t}$, which are subsequently passed on to the next memory unit. The hidden state $\boldsymbol{H_t}$ also serves as the output result $\boldsymbol{y_t}$ corresponding to the current time step.

Each memory unit in the LSTM architecture comprises three components: a forget gate, an input gate, and an output gate. The forget gate can be expressed mathematically as follows:

$$f_t = \sigma(\boldsymbol{W_f}[\boldsymbol{H_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_f}) \tag{1}$$

Here, σ represents the activation function, while $\boldsymbol{W_f}$ and $\boldsymbol{b_f}$ denote the weights and bias of the forget gate, respectively. By multiplying with $\boldsymbol{C_{t-1}}$, the forget gate aims to decide what information should be forgotten in it. For the implementation of the input gate, the sigmoid activation function [31] is initially employed to determine the values that require updating, as illustrated in (2), where $\boldsymbol{W_i}$ and $\boldsymbol{b_i}$ are the weight matrices and the bias. Subsequently, the tanh activation function generates a new candidate value, denoted as $\boldsymbol{C_t'}$. The mathematical expression is shown in (3), where $\boldsymbol{W_c}$ and $\boldsymbol{b_c}$ represent the weight matrices and the bias, respectively.

$$\boldsymbol{i_t} = \sigma(\boldsymbol{W_i}[\boldsymbol{H_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_i}) \tag{2}$$

$$\boldsymbol{C_t'} = tanh(\boldsymbol{W_c}[\boldsymbol{H_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_c}) \tag{3}$$

These two stages are subsequently combined to generate an updated state value, which is then added to the unit state to update the long-term memory of LSTM (i.e., $\boldsymbol{C_t}$), as indicated by the following equation:

$$\boldsymbol{C_t} = \boldsymbol{f_t} * \boldsymbol{C_{t-1}} + \boldsymbol{i_t} * \boldsymbol{C_t'} \tag{4}$$

The output gate is responsible for generating the hidden state, which can be calculated as:

$$\boldsymbol{H_t} = \sigma(\boldsymbol{W_o}[\boldsymbol{H_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_o}) * tanh(\boldsymbol{C_t}) \tag{5}$$

where $\boldsymbol{W_o}$ and $\boldsymbol{b_o}$ are the weight matrix and the bias of the output gate. LSTM is capable of selectively memorizing and forgetting features via the regulation of three gates, thereby mitigating the issue of long-term dependency. Additionally, LSTM addresses the issue of vanishing gradients that often occurs in RNN. As a result, LSTM has gained widespread adoption in time series prediction tasks.

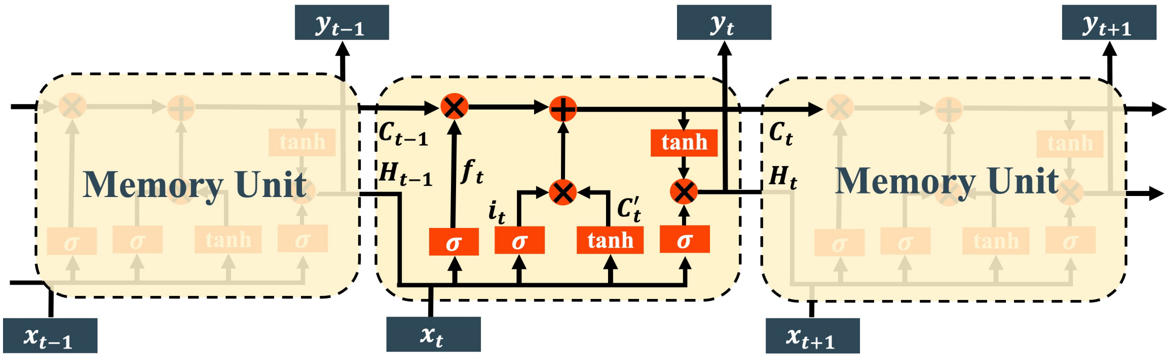


Figure 3. Architecture of LSTM.

3.3. Self-attention Mechanism

The attention mechanism is inspired by the human visual attention mechanism, which selectively focuses on specific regions of interest and allocates more attentional resources to extract relevant information while suppressing irrelevant information. Self-attention is a type of attention mechanism, which enables the model to capture the degree of association between each position in a sequence and all other positions. By computing the attention weight of each position with respect to all other positions, the model is able to selectively focus on the most relevant parts of the input sequence and generate more precise predictions or representations.

The self-attention mechanism is based on the query matrix $\boldsymbol{Q}$, the key matrix $\boldsymbol{K}$, and the value matrix $\boldsymbol{V}$, the generation of which is depicted in Figure 4. Given an input sequence $\boldsymbol{X}$, the attention mechanism employs three trainable weight matrices (corresponding to $\boldsymbol{W_Q}$, $\boldsymbol{W_K}$ and $\boldsymbol{W_V}$ in Figure 4) to compute the query matrix, the key matrix and the value matrix $\boldsymbol{V}$, respectively. By computing the dot product between $\boldsymbol{Q}$ and $\boldsymbol{K}$, and normalizing the resulting scores using a softmax function, the attention weight coefficients can be obtained, which can be expressed as:

$$AW(\boldsymbol{Q}, \boldsymbol{K}) = softmax\left(\frac{\boldsymbol{QK}^T}{\sqrt{d}}\right) \tag{6}$$

where d refers to the dimension of the hidden layer in the key and query matrices. Due to the potentially large dot product of the query matrix $\boldsymbol{Q}$ and the key matrix $\boldsymbol{K}$ when their dimensions are high, numerical instability may occur during training. To address this issue, dividing the dot product by $\sqrt{d}$ normalizes the scale of the product across all dimensions, enhancing the stability and performance of the model. Furthermore, based on the attention weight $AW(\boldsymbol{Q},\boldsymbol{K})$, the attention value can be expressed as:

$$A(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\boldsymbol{T}}}{\sqrt{d}}\right)\boldsymbol{V} \tag{7}$$

Specifically, for each position in the sequence, the corresponding value vector is multiplied by its attention weight coefficient. The resulting products are then summed to obtain the attention value, allowing the model to place greater emphasis on the most relevant positions. This process is illustrated in Figure 5, where $\{\alpha_{i,1}, \alpha_{i,2}, \cdots, \alpha_{i,d}\}$ represents the attention weight coefficients.

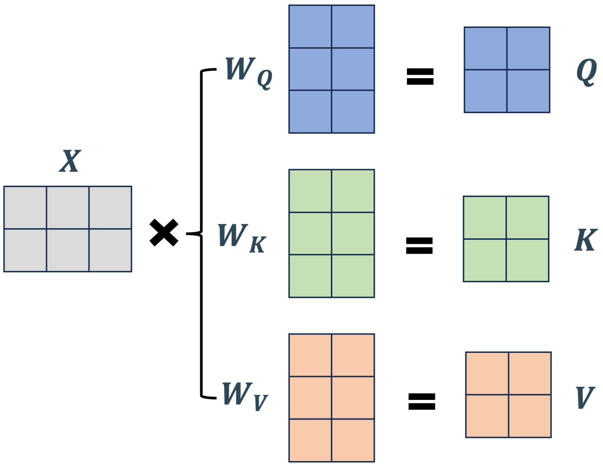


Figure 4. The generation of the $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ matrices.

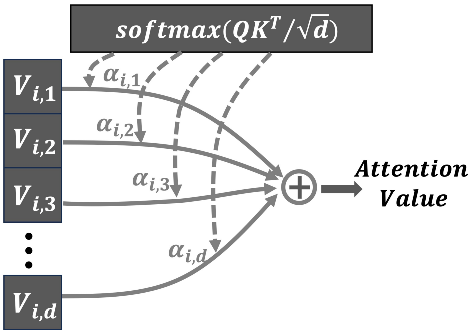


Figure 5. The generation of the attention value.

3.4. Proposed SA-LSTM Network

Based on the LSTM model and the self-attention mechanism, this paper proposes an SA-LSTM model for indoor localization enhancement. The framework of the SA-LSTM model is depicted in Figure 6. The input data for SA-LSTM is constructed using the collected RSSI data.

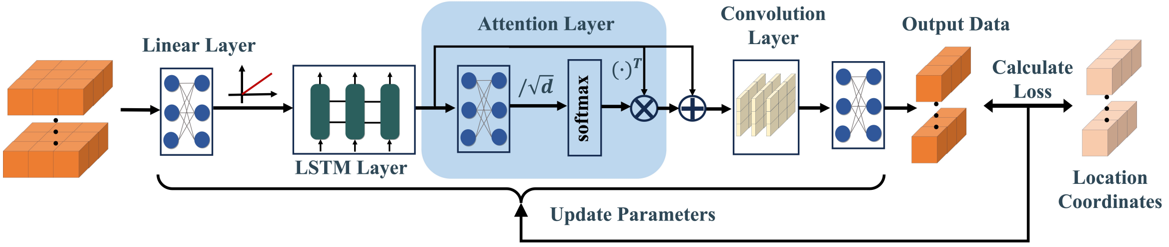


Figure 6. The framework of proposed SA-LSTM model.

3.4.1. Input Sequence Data

At first, a recorded trajectory can be expressed as a matrix:

$$\boldsymbol{R} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,N} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ r_{T,1} & r_{T,2} & \cdots & r_{T,N} \end{bmatrix} \tag{8}$$

In this context, N refers to the total number of APs, while T represents the length of a trajectory. Each element in the matrix $\boldsymbol{R}$ corresponds to the received RSSI. To prepare the data for analysis, we apply the normalization method described in [32]. This involves using the following expression:

$$r'_{i,j} = \left(\frac{r_{i,j} - c}{-c}\right)^e \tag{9}$$

Where e represents the Euler's number [33]. The constant value c should be set to a number less than or equal to the minimum value of RSSI. This ensures that all RSSI values can be scaled between 0 and 1 through normalization. Once the normalization is complete, trajectory segmentation will be performed on all the collected trajectories. Considering trajectories as $[(\widetilde{\boldsymbol{r}}_1, \boldsymbol{l}_1), (\widetilde{\boldsymbol{r}}_2, \boldsymbol{l}_2), \cdots, (\widetilde{\boldsymbol{r}}_T, \boldsymbol{l}_T)]$, where $\widetilde{\boldsymbol{r}}_i = [r_{1,1}, r_{1,2}, \cdots, r_{1,N}]$ represents the RSSI from all APs in a given position, while $\boldsymbol{l}_i = [x_i, y_i]$ represents the corresponding coordinates of this position. To facilitate analysis, each trajectory is divided into smaller segments using a sliding window of a fixed length, denoted as L. These segments are then used as inputs for the SA-LSTM model. Mathematically, this process can be expressed as follows:

$$\boldsymbol{l}_{i+L} = \boldsymbol{\mathcal{F}}\left(\widetilde{\boldsymbol{r}}_i, \widetilde{\boldsymbol{r}}_{i+1}, \cdots, \widetilde{\boldsymbol{r}}_{i+L-1}\right) \tag{10}$$

where $\boldsymbol{\mathcal{F}}(\cdot)$ is the mathematical expression of SA-LSTM, and the $\boldsymbol{l}_{i+L}$ represents the position of last time step for input data.

3.4.2. The Layers of Network

After preprocessing the data, the prepared dataset will be fed into the SA-LSTM model. The input layer of the SA-LSTM model employs a fully connected layer with a rectified linear (ReLU) activation function to increase the dimension of the feature space. Mathematically, this can be expressed as follows:

$$\boldsymbol{X}' = ReLU(\boldsymbol{W}_1\boldsymbol{X} + \boldsymbol{b}_1) \tag{11}$$

The resulting output will then be passed through an LSTM layer to generate the corresponding output for each time step. This output will serve as the input for the subsequent self-attention layer. Within the self-attention layer, several enhancements are implemented to decrease the number of network parameters. As shown in equation (6), the attention weights are computed using the query matrix $\boldsymbol{Q}$ and the key matrix $\boldsymbol{K}$. This computation can be further simplified as follows:

$$
\begin{aligned}
AW(\boldsymbol{X}, \boldsymbol{W_a}) &= softmax\left(\frac{\boldsymbol{QK}^T}{\sqrt{d}}\right) \\
&= softmax\left(\frac{(\boldsymbol{XW_Q})(\boldsymbol{XW_K})^T}{\sqrt{d}}\right) \\
&= softmax\left(\frac{\boldsymbol{X}(\boldsymbol{W_Q W_K^T})\boldsymbol{X}^T}{\sqrt{d}}\right)
\end{aligned} \tag{12}
$$

Given the relationship $\boldsymbol{W_A} = \boldsymbol{W_Q W_K^T}$, it follows that a fully connected layer with trainable weights $\boldsymbol{W_A}$ can be utilized in the attention layer to facilitate the computation of attention weights. Afterward, the output of the fully connected layer will be divided by $\sqrt{d}$ and normalized using the softmax function to obtain the attention weights. It is noteworthy that the output of the LSTM layer contains the information required for SA-LSTM, which means it can be considered as the key matrix $\boldsymbol{K}$ directly. After calculating the attention weights, the next step involves performing a dot product operation between the attention weights and the transposed output from the LSTM layer.

SA-LSTM utilizes a shortcut connection [34] to propagate the attention values obtained from the attention layer, which enhances the backpropagation of gradients and mitigates gradient vanishing. A convolutional layer is then applied to modify the data channels before moving on to the final layer. In the final layer, a fully connected layer is employed to convert the input into location coordinates. The model then calculates the Mean Square Error (MSE) between the predicted output $\tilde{\boldsymbol{Y}}$ and the practical location coordinates $\boldsymbol{Y}$. The loss is calculated as:

$$
\boldsymbol{\mathcal{L}_{MSE}}(\tilde{\boldsymbol{Y}}, \boldsymbol{Y}) = \frac{\sum_{i=1}^{n}(\|\boldsymbol{Y} - \tilde{\boldsymbol{Y}}\|^2)}{n} \tag{13}
$$

where n denotes the number of samples in a batch. According to the loss value, the gradients of the trainable parameters in the model will be computed through backpropagation. Simultaneously, the trainable parameters will be updated in the direction of the negative gradient to minimize the loss value.

4. Experimental Setup

To verify the performance of the proposed SA-LSTM method, Bluetooth and Wifi fingerprint data are applied, which are collected from 2D and 3D moving scenarios, respectively.

4.1. 2D-moving Experiment Setup

The experimental location for 2D-moving scenarios is located in an office room on the 28th floor of the Guangdong Telecom Science and Technology Building in China. In this experiment, we deployed 24 Bluetooth beacons at various locations within an office room. These beacons are used to track the movement and location of individuals. Figure 7 shows the layout of the office room, which has an area of 9.6 m × 20.4 m. The solid red dot in Figure 7 represents the origin point in a customized absolute coordinate system. The trajectories used for feature analysis are based on the coordinates of an absolute coordinate system, which serves as a reference point for all position measurements. Besides, the green cross marks in Figure 7 represent the positions of the Bluetooth beacons, while the blue dashed line indicates the trajectories followed during data collection. The E5 Pilot Positioning Beacon version V006 is applied as the Bluetooth signal transmitter. The specific product parameters are shown in Table 1.

Table 1. THE PRODUCT PARAMETERS OF BLUETOOTH BEACON.

Parameters	Values
Bluetooth Version	BLE 5.0
Bluetooth Protocol	iBeacon
Working Temperature	-30 ∼ 25°C
Maximum Transmission Distance	120 m
Transmitted Power	-30 ∼ +4 dBm (default: 0 dBm)
Broadcast Interval	100 ms ∼ 10 s (default: 500 ms)

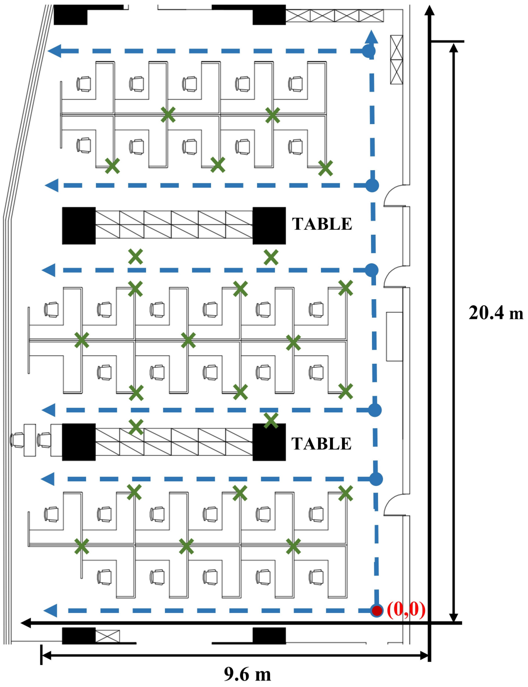
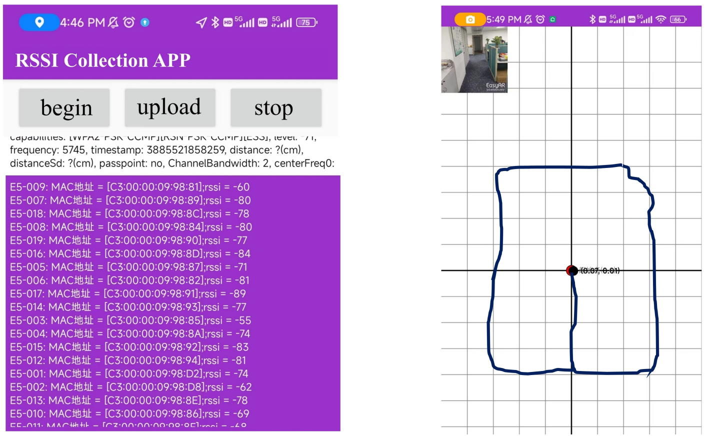


Figure 7. The layout of office room.

During the experiment, we employed a Xiaomi 10 Pro mobile phone and a ZTE Axon 40 mobile phone, both equipped with cameras. To facilitate the data collection task, we developed a mobile phone data collection application capable of capturing Bluetooth signals and logging user positions. In Figure 8, we depict the page of the application. This application leverages the Visual Simultaneous Localization and Mapping (VSLAM) framework to acquire real-time coordinates, which were then logged onto files for further analysis. The working principle of VSLAM involves analyzing the visual data captured by the camera to track the movement of the camera and identify features in the environment. By comparing these features with those from previous frames, VSLAM can estimate the motion of the camera and update its position in real time.

To ensure the accuracy of the collected position coordinates, we conducted data acquisition by moving the acquisition device at a constant speed along the predetermined trajectories. The trajectory data for RSSI collection was obtained by following the blue dashed lines shown in Figure 7. Specifically, we followed each dashed line from the starting point to the end and then retraced our steps from the end back to the exit point, creating two distinct trajectories. The two mobile phones used for data acquisition were programmed to perform signal acquisition and collect corresponding addresses at different times. Overall, these measures ensured that the collected data were of sufficient quality to support our research objectives. The sampling frequency of the collecting devices was set to 1 Hz while moving along the trajectories. In total, we collected 28 trajectories, which were subsequently partitioned into three sets: training, validation, and test sets, in a ratio of 3:1:1. The test and validation datasets mainly contain two categories of trajectories. The first category consists of trajectories that were not included in the training set. The second category includes trajectories that are identical to those in the training set but were collected using different devices.



```
capabilities: [WPA2-PSK-CCMP][RSN-PSK-CCMP][ESS], level: -71,
frequency: 5745, timestamp: 3885521858259, distance: ?(cm),
distanceSd: ?(cm), passpoint: no, ChannelBandwidth: 2, centerFreq0:
E5-009: MAC地址 = [C3:00:00:09:98:81];rssi = -60
E5-007: MAC地址 = [C3:00:00:09:98:89];rssi = -80
E5-018: MAC地址 = [C3:00:00:09:98:8C];rssi = -78
E5-008: MAC地址 = [C3:00:00:09:98:84];rssi = -80
E5-019: MAC地址 = [C3:00:00:09:98:90];rssi = -77
E5-016: MAC地址 = [C3:00:00:09:98:8D];rssi = -84
E5-005: MAC地址 = [C3:00:00:09:98:87];rssi = -71
E5-006: MAC地址 = [C3:00:00:09:98:82];rssi = -81
E5-017: MAC地址 = [C3:00:00:09:98:91];rssi = -89
E5-014: MAC地址 = [C3:00:00:09:98:93];rssi = -77
E5-003: MAC地址 = [C3:00:00:09:98:85];rssi = -55
E5-004: MAC地址 = [C3:00:00:09:98:8A];rssi = -74
E5-015: MAC地址 = [C3:00:00:09:98:92];rssi = -83
E5-012: MAC地址 = [C3:00:00:09:98:94];rssi = -81
E5-001: MAC地址 = [C3:00:00:09:98:D2];rssi = -74
E5-002: MAC地址 = [C3:00:00:09:98:D8];rssi = -62
E5-013: MAC地址 = [C3:00:00:09:98:8E];rssi = -78
E5-010: MAC地址 = [C3:00:00:09:98:86];rssi = -69
E5-011: MAC地址 = [C3:00:00:09:98:8F];rssi = -68
```

(a) **(b)**

Figure 8. The application used for RSSI and position collection.

4.2. 3D-moving Experiment Setup

The 3D-moving experiment dataset is publicly available as an open-source dataset [4]. In contrast to the 2D-moving experiment, the 3D-moving experiment dataset is based on WiFi fingerprints and covers trajectories across the fourth, fifth, and sixth floors of the C4 building at Nazarbayev University. This dataset provides a comprehensive and representative set of data, enabling a thorough evaluation of the performance of indoor localization systems in complex, multi-floor environments. This WiFi dataset comprises 290 trajectories that were sequentially collected with a fine spatiotemporal resolution. The dataset covers a total area of over 9564 m^2 across three floors. The experimental environment is equipped with 439 wireless access points. During the experiment, the validation and test trajectories were collected a few days after obtaining the training set. These trajectories were uniquely designed to be dissimilar from the training trajectories. Moreover, the users were authorized to switch floors using the four elevators installed in the building while collecting the data, which helps to evaluate the performance of the model in 3D-moving scenarios. A total of 170 unique trajectories were collected, with an even distribution between the validation and test sets.

4.3. SA-LSTM Training Setup

In the two experimental scenarios, the hyperparameters of the SA-LSTM model were adjusted differently. The details of these hyperparameters are presented in Table 2. For each L of consecutive input RSSI vectors at a given moment, the network predicts the exact location of the last recorded time point. The initial learning rate is set to 0.001 for both scenarios. During the training process, we reduce the learning rate to one-tenth of the previous rate after a fixed number of training epochs. In the 2D scenario, the learning rate was adjusted every 30 epochs, while in the 3D scenario, the learning rate was adjusted every 20 epochs. All models were trained using an NVIDIA GeForce RTX 2080 Ti GPU.

Table 2. THE HYPERPARAMETERS OF SA-LSTM.

Layer	2D Experiment	3D Experiment
Linear Layer 1	(24×64)	(436×128)
LSTM Layer	(64×64)	(128×128)
Linear Layer 2	(64×4)	(128×4)
Convolution Layer	3 × 3 kernels, 1 filter	3 × 3 kernels, 1 filter
Linear Layer 3	(62×2)	(126×3)
Batch Size	2	2
Initial Learning Rate	0.001	0.001
Optimizer	Adam	Adam
Loss Function	MSE	MSE
Training Epochs	200	100

5. Results and Discussion

Before comparing the performance of various methods, the sliding window length L for the SA-LSTM method needs to be determined. Figure 9 illustrates the mean positioning error as a function of the window size. As shown in the figure, SA-LSTM performs poorly when L is set to 1 or 2. As L increases, the average localization error of SA-LSTM shows a significant decrease. This occurs because when L is set to a smaller value, the network model obtains less information, resulting in lower positioning accuracy. When L is taken to 5 or 6, the average localization error fluctuates within a small range. To avoid additional computational complexity, L is determined to be set to 4.

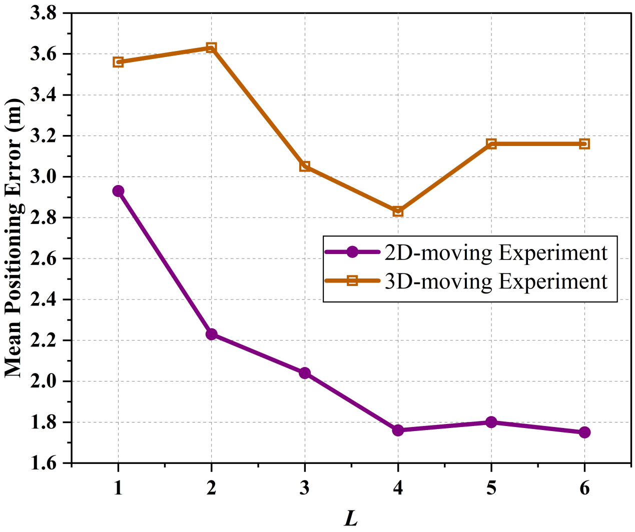


Figure 9. The length of Sliding window against mean positioning error.

To compare our indoor localization approach, we have implemented an indoor localization system network based on LSTM, as described in [28]. Additionally, we have implemented other methods such as RNN [4], KNN, WKNN, FNN and Linear Regression. We have adjusted the parameters of these models within a certain range to optimize their performance. During the training process, all the model was validated using the validation set after each training epoch, and the model with the minimum average position error was saved for further evaluation.

The average and maximum positioning errors of all these methods are presented in Table 3. Apparently, the SA-LSTM method outperforms other methods in terms of average positioning accuracy. Among these methods, the LSTM approach achieves the second-best performance in mean positioning accuracy, following the proposed SA-LSTM method. On the test set, the LSTM method results in a maximum error of 13.73 m and an average error of 3.07 m, which is 0.98 m and 1.31 m higher than the proposed SA-LSTM method. Compared to the RNN method, which has a mean positioning error of 4.16 m and a maximum error of 12.64 m, SA-LSTM improves the positioning accuracy by 2.4 m and 0.29 m. Moreover, SA-LSTM achieves a maximum improvement of 66.85% in average positioning accuracy compared to the Linear Regression method.

Table 3. THE POSITIONING ERROR FOR 2D-MOVING EXPERIMENT.

Method	Average Error (m)		Maximum Error (m)	
	Validation Set	Test Set	Validation Set	Test Set
KNN	2.53	3.36	18.39	15.22
WKNN	2.53	3.33	18.41	15.42
FNN	3.49	5.28	12.44	12.54
Linear Regression	3.64	5.31	13.06	**12.34**
RNN	3.37	4.16	12.67	12.64
LSTM	2.57	3.07	13.73	13.73
SA-LSTM	**1.67**	**1.76**	**12.35**	12.35

Figure 10 illustrates the MSE loss curve of the SA-LSTM and LSTM methods during the training process with 2D-moving trajectories. Our results indicate that exhibits a faster convergence rate in terms of training loss compared to the LSTM model. Moreover, after 200 epochs of training, the training loss of SA-LSTM converges to around 0, while the training loss of LSTM converges to around 0.5. The validation loss of SA-LSTM converges faster to near-stabilization values compared to LSTM, as demonstrated in the black dotted box in Figure 10. Throughout the entire training process, we observed that the SA-LSTM model achieved a slightly lower minimum validation loss than the LSTM model. These results suggest that the SA-LSTM model is more effective in terms of training efficiency with the help of self-attention mechanism and shortcut connection.

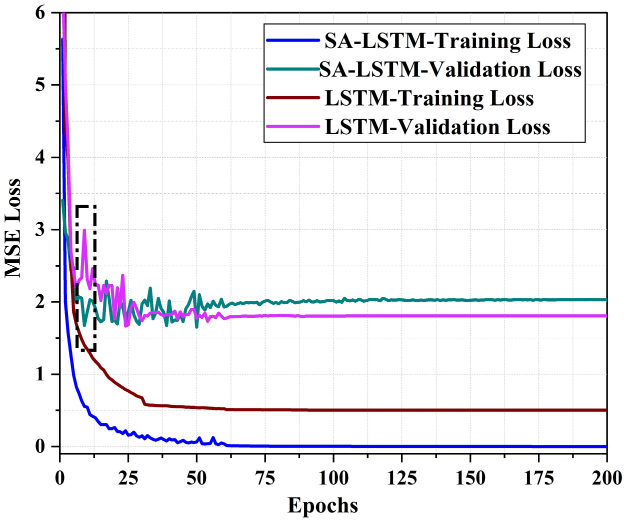


Figure 10. The MSE loss curve of SA-LSTM and LSTM methods in 2D-moving experiment.

Figure 11 illustrates the cumulative distribution function (CDF) of localization errors for the 2D-moving experiment. In total, a maximum localization error of 12.35 m is recorded for SA-LSTM, 15.22 m for KNN, and the largest maximum localization error of 15.42 m for WKNN. Compared to the KNN and WKNN methods, the SA-LSTM method showed a decrease in maximum localization error by 2.87 m and 3.07 m, respectively. Meanwhile, the maximum localization error of LSTM is 12.47 m, which is also higher than that of SA-LSTM. When considering the 90% percentile of the CDF, the proposed SA-LSTM model demonstrates a 90% location error of approximately under 3.86 m. In comparison, the LSTM, RNN, and KNN models exhibit location errors of around 4.36 m, 5.74 m, and

6.31 m, respectively. This suggests that the proposed SA-LSTM can achieve an improvement by 11.47%, 32.75%, and 63.47% in the 90% CDF compared to LSTM, RNN, and KNN, respectively.

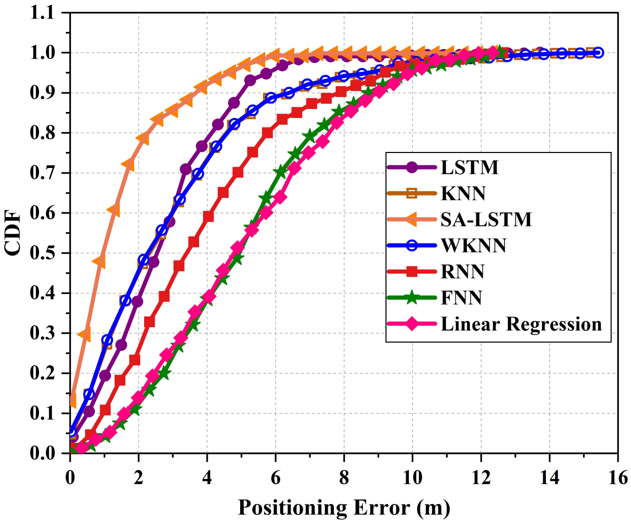


Figure 11. The CDF of localization errors for 2D-moving experiment.

Regarding the 3D-moving experiment, the proposed SA-LSTM model continues to exhibit superior performance in the localization system. Similarly, we compare the average and maximum positioning error of KNN, WKNN, FNN, Linear Regression, RNN, LSTM and SA-LSTM. As shown in Table 4, the proposed SA-LSTM achieves an average positioning error of 2.83 m and a maximum positioning error of 57.64 m in the 3D-moving experiment. Compared to LSTM, SA-LSTM improves the average positioning accuracy by 31.64%. In addition, SA-LSTM reduces the average positioning errors by 2.1 m and the maximum localization errors by 3.32 m compared to RNN. Compared to KNN and WKNN, the SA-LSTM has an average positioning error that is 0.62 m and 0.61 m lower, respectively. The SA-LSTM has achieved the lowest average positioning error and the maximum positioning error in scenes involving 3D motion.

Table 4. THE POSITIONING ERROR FOR 3D-MOVING EXPERIMENT.

Method	Average Error (m)		Maximum Error (m)	
	Validation Set	Test Set	Validation Set	Test Set
KNN	3.42	3.45	68.95	69.99
WKNN	3.41	3.44	68.95	69.99
FNN	6.41	6.81	68.79	58.70
Linear Regression	7.06	7.56	100.44	89.74
RNN	3.73	4.93	40.71	60.96
LSTM	3.91	4.14	66.91	69.29
SA-LSTM	**2.56**	**2.83**	**28.46**	**57.64**

The loss curves for SA-LSTM and LSTM in the 3D-moving experiment are depicted in Figure 12. The training loss of SA-LSTM and LSTM converge at a similar rate. As shown in the zoomed-in image in Figure 12, the final convergence value of SA-LSTM is a bit lower. In terms of the validation loss, the SA-LSTM model exhibited better performance than the LSTM model. Specifically, the validation loss of SA-LSTM could eventually converge to 3, while that of LSTM remained above 4. Based on these findings, we can conclude that our proposed SA-LSTM model is significantly more efficient in terms of training efficiency compared to the conventional LSTM model.

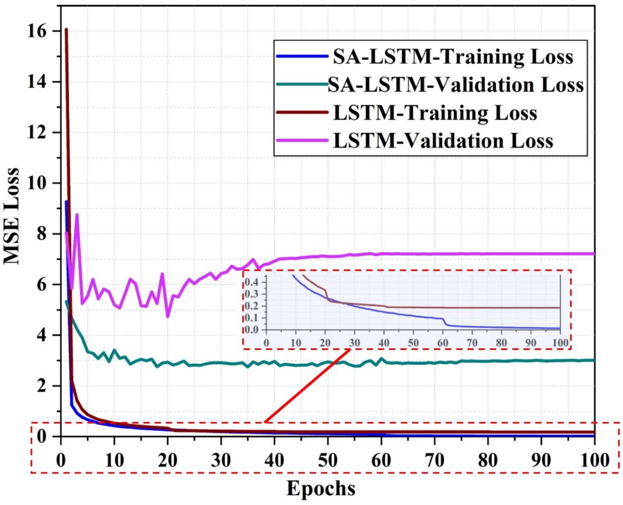


Figure 12. The MSE loss curve of SA-LSTM and LSTM methods in 3D-moving experiment.

Figure 14 illustrates the CDF of localization errors for the 3D-moving experiment. Overall, the proposed SA-LSTM still outperforms the other classical algorithms. LSTM network performs the second best, which achieves a 90% location error below 6 m, while RNN achieves a 90% location error below 8.45 m. compared to LSTM and RNN, SA-LSTM decreased the 90% CDF by 1.99 m and 4.44 m.

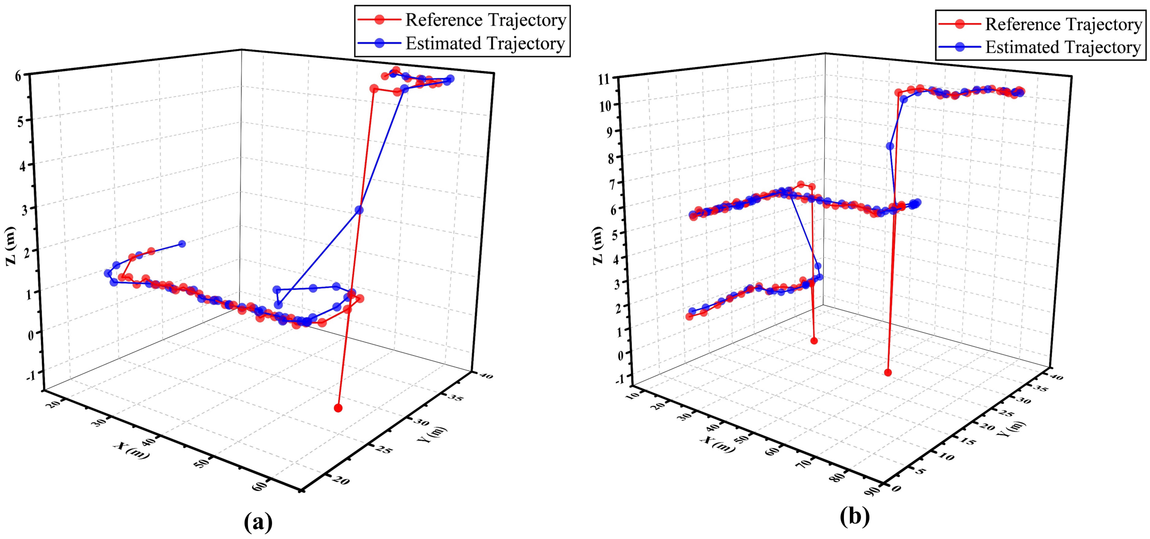


Figure 13. Schematic diagram of referenced and estimated trajectories with a range of movement involving (a) two floors and (b) three floors.

Furthermore, a couple of estimated trajectories are drawn in a 3D-moving experiment using the SA-LSTM model. Figure 13 (a) and Figure 13 (b) depict the moving trajectories, which involve transitions between two and three different floors, respectively. The red lines correspond to the reference trajectory, whereas the blue lines depict the estimated trajectories generated by SA-LSTM. The experimental results indicate that the measured position points in the referenced trajectories exhibit anomalous behavior during pedestrian transitions between different floors. This behavior is attributed to the reliance on elevators for inter-floor movement, which leads to abnormal fluctuations in the measurement signal, resulting in anomalous measured positions. From the trajectories shown in Figure 13 (a) and Figure 13 (b), it can be demonstrated that the proposed SA-LSTM model exhibits a satisfactory performance when the pedestrians under test move within a single floor. However, when pedestrians move between floors, the estimated position points generated by the SA-LSTM model may

doi:10.20944/preprints202401.1768.v1

exhibit some fluctuations within a narrow range. Nevertheless, once the pedestrians reach a specific floor, the SA-LSTM model can promptly resume its effective operation.

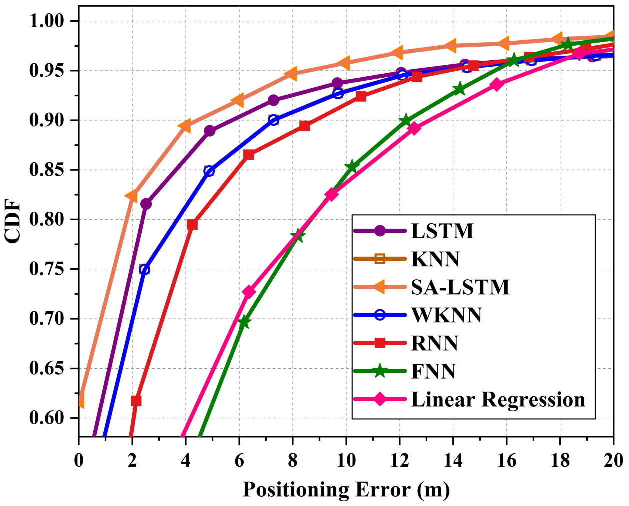


Figure 14. The CDF of localization errors for 3D-moving experiment.

The 90% quantile of CDF is an important performance evaluation metric in location systems, as highlighted in 3GPP Rel.18 [35]. To comprehensively evaluate the performance of each algorithm in both 2D-moving and 3D-moving experiments, we calculate the 90% error for each algorithm and present the results in Figure 15.

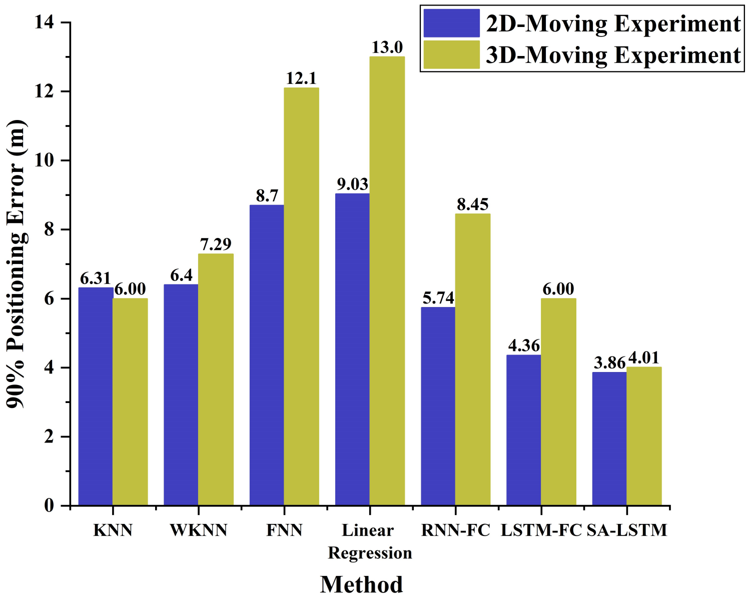


Figure 15. The histogram of 90% positioning error for two experiments.

In both experimental scenarios, SA-LSTM demonstrates the highest localization accuracy compared to the other algorithms, as indicated by its remarkably low 90% positioning error. Under the 3D-moving experimental environment, SA-LSTM achieves a 90% localization error under 3.86 m, which is 0.5 m and 1.88 m lower than that of LSTM and RNN, respectively. Compared to classical KNN algorithms, the SA-LSTM model consistently exhibits a lower 90% positioning error under both experimental environments. These results suggest that SA-LSTM demonstrates high accuracy and stability in the field of indoor positioning, highlighting its potential to outperform traditional methods and pave the way for more advanced and reliable indoor positioning systems.

6. Conclusion

This paper introduces a novel SA-LSTM method for fingerprint-based indoor localization systems. The proposed model utilizes the self-attention mechanism to calculate attention scores between each element and all other elements in the output sequence of the LSTM. This enables the SA-LSTM model to focus on the relationship between the position features at different time steps, thereby improving the accuracy of real-time position estimation. The performance of SA-LSTM has been evaluated under various experimental environments that involve 2D and 3D moving trajectories. The experimental results show that SA-LSTM achieves an average localization error of 1.76 m and 2.83 m in the respective scenarios, with 90% of positioning errors being under 3.86 m and 4.01 m, respectively. Furthermore, when compared with existing state-of-the-art methods in the same test environment, SA-LSTM exhibits a significant improvement in positioning accuracy by 42.67% to 31.64% under the same test environment.

Our study has successfully showcased the potential of the self-attention mechanism in enhancing the accuracy and efficiency of indoor localization systems. In our future work, we plan to conduct further research to explore the applicability and effectiveness of this mechanism in improving the accuracy of indoor localization.

Author Contributions: Conceptualization, Zhen Wu; Data curation, Peng Hu and Shuangyue Liu; Formal analysis, Zhen Wu; Methodology, Zhen Wu; Project administration, Tao Pang; Software, Zhen Wu; Writing—review & editing, Zhen Wu. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China under Grant 2022YFB3904700.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

LBS	Location-based services
GPS	Global Positioning System
BDS	BeiDou Satellite Navigation System
UWB	Ultra-Wide Bandwidth
RFID	Radio Frequency Identification
AOA	Angle of Arrival
TOA	Time of Arrival
APs	Access Points
UE	User Equipment
LSTM	Long Short-term Memory
RSSI	Received Signal Strength Indicator
TPs	Test Points
RPs	Reference Points
KNN	K-Nearest Neighbors
WKNN	Weighted K-Nearest Neighbors
SVM	Support Vector Machines
FNN	Feedforward Neural Networks
SA-LSTM	Self-Attention and LSTM
VSLAM	Simultaneous Localization and Mapping
RNN	Recurrent Neural Networks

References

1. Zhong, S.; Li, L.; Liu, Y.G.; Yang, Y.R. Privacy-preserving location-based services for mobile users in wireless networks. *Department of Computer Science, Yale University, Technical Report ALEU/DCS/TR-1297* **2004**, *26*.
2. Spilker Jr, J.J.; Axelrad, P.; Parkinson, B.W.; Enge, P. *Global positioning system: theory and applications, volume I*; American Institute of Aeronautics and Astronautics, 1996.
3. Yang, Y.; Gao, W.; Guo, S.; Mao, Y.; Yang, Y. Introduction to BeiDou-3 navigation satellite system. *Navigation* **2019**, *66*, 7–18.
4. Khassanov, Y.; Nurpeiissov, M.; Sarkytbayev, A.; Kuzdeuov, A.; Varol, H.A. Finer-level sequential wifi-based indoor localization. In Proceedings of the 2021 IEEE/SICE International Symposium on System Integration (SII). IEEE, 2021, pp. 163–169.
5. Salamah, A.H.; Tamazin, M.; Sharkas, M.A.; Khedr, M. An enhanced WiFi indoor localization system based on machine learning. In Proceedings of the 2016 International conference on indoor positioning and indoor navigation (IPIN). IEEE, 2016, pp. 1–8.
6. Abbas, M.; Elhamshary, M.; Rizk, H.; Torki, M.; Youssef, M. WiDeep: WiFi-based accurate and robust indoor localization system using deep learning. In Proceedings of the 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom. IEEE, 2019, pp. 1–10.
7. Chen, C.; Chen, Y.; Lai, H.Q.; Han, Y.; Liu, K.R. High accuracy indoor localization: A WiFi-based approach. In Proceedings of the 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2016, pp. 6245–6249.
8. Altini, M.; Brunelli, D.; Farella, E.; Benini, L. Bluetooth indoor localization with multiple neural networks. In Proceedings of the IEEE 5th International Symposium on Wireless Pervasive Computing 2010. IEEE, 2010, pp. 295–300.
9. Wang, Y.; Ye, Q.; Cheng, J.; Wang, L. RSSI-based bluetooth indoor localization. In Proceedings of the 2015 11th international conference on mobile ad-hoc and sensor networks (MSN). IEEE, 2015, pp. 165–171.
10. Zhang, C.; Kuhn, M.; Merkl, B.; Fathy, A.E.; Mahfouz, M. Accurate UWB indoor localization system utilizing time difference of arrival approach. In Proceedings of the 2006 IEEE radio and wireless symposium. IEEE, 2006, pp. 515–518.
11. Poulose, A.; Han, D.S. UWB indoor localization using deep learning LSTM networks. *Applied Sciences* **2020**, *10*, 6290.
12. Montaser, A.; Moselhi, O. RFID indoor location identification for construction projects. *Automation in Construction* **2014**, *39*, 167–179.
13. Chen, Y.; Lymberopoulos, D.; Liu, J.; Priyantha, B. Indoor localization using FM signals. *IEEE Transactions on Mobile Computing* **2013**, *12*, 1502–1517.
14. Lan, T.; Wang, X.; Chen, Z.; Zhu, J.; Zhang, S. Fingerprint augment based on super-resolution for WiFi fingerprint based indoor localization. *IEEE Sensors Journal* **2022**, *22*, 12152–12162.
15. Torres-Sospedra, J.; Montoliu, R.; Martínez-Usó, A.; Avariento, J.P.; Arnau, T.J.; Benedito-Bordonau, M.; Huerta, J. UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. In Proceedings of the 2014 international conference on indoor positioning and indoor navigation (IPIN). IEEE, 2014, pp. 261–270.
16. Roy, P.; Chowdhury, C. A survey of machine learning techniques for indoor localization and navigation systems. *Journal of Intelligent & Robotic Systems* **2021**, *101*, 63.
17. Brunato, M.; Battiti, R. Statistical learning theory for location fingerprinting in wireless LANs. *Computer Networks* **2005**, *47*, 825–845.
18. Hoang, M.T.; Zhu, Y.; Yuen, B.; Reese, T.; Dong, X.; Lu, T.; Westendorp, R.; Xie, M. A soft range limited K-nearest neighbors algorithm for indoor localization enhancement. *IEEE Sensors Journal* **2018**, *18*, 10208–10216.
19. Fang, S.H.; Lin, T.N. Indoor location system based on discriminant-adaptive neural network in IEEE 802.11 environments. *IEEE Transactions on Neural networks* **2008**, *19*, 1973–1978.
20. Nurpeiissov, M.; Kuzdeuov, A.; Assylkhanov, A.; Khassanov, Y.; Varol, H.A. End-to-end sequential indoor localization using smartphone inertial sensors and WiFi. In Proceedings of the 2022 IEEE/SICE International Symposium on System Integration (SII). IEEE, 2022, pp. 566–571.

21. Zhang, Y.; Qu, C.; Wang, Y. An indoor positioning method based on CSI by using features optimization mechanism with LSTM. *IEEE Sensors Journal* **2020**, *20*, 4868–4878.
22. Gibbons, F.X. Self-attention and behavior: A review and theoretical update. *Advances in experimental social psychology* **1990**, *23*, 249–303.
23. Zhao, H.; Jia, J.; Koltun, V. Exploring self-attention for image recognition. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10076–10085.
24. Humphreys, G.W.; Sui, J. Attentional control and the self: The self-attention network (SAN). *Cognitive neuroscience* **2016**, *7*, 5–17.
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.
26. Bahl, P.; Padmanabhan, V.N. RADAR: An in-building RF-based user location and tracking system. In Proceedings of the Proceedings IEEE INFOCOM 2000. Conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064). Ieee, 2000, Vol. 2, pp. 775–784.
27. Brunato, M.; Battiti, R. Statistical learning theory for location fingerprinting in wireless LANs. *Computer Networks* **2005**, *47*, 825–845.
28. Chen, Z.; Zou, H.; Yang, J.; Jiang, H.; Xie, L. WiFi fingerprinting indoor localization using local feature-based deep LSTM. *IEEE Systems Journal* **2019**, *14*, 3001–3010.
29. Chorowski, J.K.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y. Attention-based models for speech recognition. *Advances in neural information processing systems* **2015**, *28*.
30. Zang, H.; Xu, R.; Cheng, L.; Ding, T.; Liu, L.; Wei, Z.; Sun, G. Residential load forecasting based on LSTM fusing self-attention mechanism with pooling. *Energy* **2021**, *229*, 120682.
31. Finney, D.J. Probit analysis; a statistical treatment of the sigmoid response curve. **1947**.
32. Torres-Sospedra, J.; Montoliu, R.; Trilles, S.; Belmonte, Ó.; Huerta, J. Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems. *Expert Systems with Applications* **2015**, *42*, 9263–9278.
33. Song, X.; Fan, X.; Xiang, C.; Ye, Q.; Liu, L.; Wang, Z.; He, X.; Yang, N.; Fang, G. A novel convolutional neural network based indoor localization framework with WiFi fingerprinting. *IEEE Access* **2019**, *7*, 110698–110709.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
35. Morselli, F.; Razavi, S.M.; Win, M.Z.; Conti, A. Soft information based localization for 5G networks and beyond. *IEEE Transactions on Wireless Communications* **2023**.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.