electronics

MDPI

Article

A Hybrid CNN-LSTM-Based Approach for Pedestrian Dead Reckoning Using Multi-Sensor-Equipped Backpack

Feyissa Woyano [1,2], Sangjoon Park [1,2,*], Vladimirov Blagovest Iordanov [2] and Soyeon Lee [2]

[1] Department of Computer Software and Engineering, University of Science and Technology (UST), 217 Gajeong-ro, Yuseong-gu, Daejeon 34113, Republic of Korea; feissaw06@gmail.com
[2] Department of Computer Software, School of Electronics and Telecommunications Research Institute (ETRI), 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Republic of Korea; vladimirov@etri.re.kr (V.B.I.); sylee@etri.re.kr (S.L.)
* Correspondence: sangjoon@etri.re.kr

Abstract: Researchers in academics and companies working on location-based services (LBS) are paying close attention to indoor localization based on pedestrian dead reckoning (PDR) because of its infrastructure-free localization method. PDR is the fundamental localization technique that utilize human motion to perform localization in a relative sense with respect to the initial position. The size, weight, and power consumption of micromechanical systems (MEMS) embedded into smartphones are remarkably low, making them appropriate for localization and positioning. Traditional pedestrian PDR methods predict position and orientation using stride length and continuous integration of acceleration in step and heading system (SHS)-based PDR and inertial navigation system (INS)-PDR, respectively. However, these two approaches provide accumulations of error and do not effectively leverage the inertial measurement unit (IMU) sequences. The PDR navigation solution relays on the standard of the MEMS, which yields PDR with the acceleration and angular velocity from the accelerometer and gyroscope, respectively. However, low-cost small MEMSs endure enormous error sources such as bias and noise. Hence, MEMS assessments lead to navigation solution drifts when utilized as inputs to the PDR. As a consequence, numerous methods have been proposed to mitigate and model the errors related to MEMS. Deep learning-based dead reckoning algorithms are provided to address aforementioned issues owing to the end-to-end learning framework. This paper proposes a hybrid convolutional neural network (CNN) and long short-term memory network (LSTM)-based inertial PDR system that extracts inertial measurement units (IMU) sequence features. The end-to-end learning framework is introduced to leverage the efficiency of low-cost MEMS because data-driven solutions provide more complete knowledge of the ever-increasing data volume and computational power over the filtering model approach. A CNN-LSTM model was employed to capture local spatial and temporal features. Experiments conducted on odometry datasets collected from multi-sensor backpack devices demonstrated that the proposed architecture outperformed previous traditional PDR methods, demonstrating that the root mean square error (*RMSE*) for the best user was 0.52 m. On the handheld smartphone-only dataset the best achieved R^2 metric was 0.49.

Keywords: inertial sensor; deep learning; pedestrian dead reckoning; CNN; multi-sensor-backpack; LSTM; LiDAR

check for updates

Citation: Woyano, F.; Park, S.; Blagovest Iordanov, V.; Lee, S. A Hybrid CNN-LSTM-Based Approach for Pedestrian Dead Reckoning Using Multi-Sensor-Equipped Backpack. *Electronics* **2023**, *12*, 2957. https://doi.org/10.3390/electronics12132957

Academic Editors: Chi Cuong Vu and Jooyong Kim

Received: 26 May 2023
Revised: 16 June 2023
Accepted: 21 June 2023
Published: 5 July 2023

Copyright: © 2023 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 (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The extraordinary development of state-of-the-art indoor location-based services (LBS) is accelerating the expansion of indoor positioning techniques [1,2]. The current demand for determining the location of dynamic agents such as humans and robots in indoor environments has been achieving unprecedented importance for society and scientific purposes. There are different techniques of indoor localization approaches such as infrastructure-based approaches and infrastructure-free approaches. Wireless fidelity

(Wi-Fi) [3–5], radio frequency identification (RFID) [6–9], ultrawide-band (UWB) [10,11], and Bluetooth (BLE) [12] are among those techniques that require tailored infrastructure. These approaches need the distribution of Wi-Fi access points (Aps), tags, and BLE beacon signals when indoors to sense the environment. The infrastructure-free [13] approaches usually require the pre-built infrastructure to update the position. Each of these techniques has its own limitations as well as advantages. Among efficient cost-effective techniques, the self-contained PDR algorithm is the most popular. The convenience of utilizing a smartphone-based indoor localization system is that it requires no infrastructure deployments. In our case, we focused on the cost-effectiveness and scalability of large environment sensors such as inertial and LIDAR sensors. This approach is usually adopted for mobile indoor localization.

Odometry estimation is a critical ingredient across many domains, such as robotics ego-motion estimation [14], unmanned aerial systems (UAS) [15], and for humans' self-motion estimation through space [16–18]. As different sensing modalities have different capabilities, dynamic agents (human or robot) operating in indoor environments are often outfitted with numerous sensors, such as cameras [19], inertial measurement units (IMU) [20], and LiDARs [21]. Therefore, tracking motion using hand-held smartphones has become highly popular in indoor environments [22]. Among motion tracking methods, PDR has become an indispensable relative positioning technology that tracks the position and orientation (pose) in an infrastructure-free environment using wearable sensors that provide linear acceleration and rotational velocity. However, a PDR system based on inertial sensors alone is challenging in part because of its deficiencies of unbounded system error which leads to the development of complex models [23]. In positioning and navigation fields, developing a new architecture is a well-researched topic as it enables ubiquitous mobility for providing reliable pose information. Traditional inertial sensor-based odometry estimation methods predict position and orientation using stride length and continuous integration of acceleration in SHS-based PDR and INS-PDR, respectively. Existing traditional PDR solutions either rely on guessing latent states or rely on the periodicity of human walking and the sensors' fixed position. However, these two approaches provide error accumulation, and do not effectively leverage this algorithm to enhance the accuracy of PDR.

Recently, the demand for deep learning (DL) approaches has been significantly increasing in almost every domain. In climate analysis and weather forecasting, DL techniques have been used and applied with techniques such as DNN and RNN [24]. They model future climate status by focusing on the use of limited scope and data to investigate their models, and then use parameter tuning and cross-validation on different data. They use a novel CDLSTM model to investigate three important aspects, namely detecting rainfall and temperature trends, analyzing the correlation between temperature and rainfall, and forecasting the temperature and rainfall. LSTM is an improvement compared to RNN, designed to capture long-range dependencies in time series data. This network is immensely beneficial for a broad range of circumstances, and it is now broadly used in various applications. In the climate change forecasting and groundwater storage field, LSTM is being used broadly. LSTM is also used for botnet detection and classification in developing fast and efficient networks. The authors in [25] proposed a deep neural network for intrusion detection. They implemented a principal component-based convolution neural network (PCCNN) approach to improve precision. The PCA is used to reduce the dimension of the feature vector. More recent work also in the area of botnet detection and classification using deep neural network (DNN) models, DNNBoT1 and DNNBoT2, for the detection and classification of internet of things (IoT) botnet attacks [26].

Few studies are proposing deep learning-based data-driven methods to enhance the traditional odometry, i.e., the INS algorithm to make robust self-localization [27]. Hannink et al. [28] proposed a mobile stride length estimation system that constrains double integration approaches from a raw foot-mounted IMU using deep convolutional neural networks. The authors in [29] used a neural network model to classify walking patterns to improve step length estimation. The recent fundamental approach is presented in [30]

and repeated in [31], where the authors proposed reliable pose estimation using 6D IMU measurements for attention-driven rotation-equivariance-supervised inertial odometry. Many studies have been focusing on applying end-to-end learning approach to inertial odometry in order to enhance performance in terms of both robustness and accuracy. In the sequence-based approach, LSTMs are applied to learn temporal correlations between multivariate time series IMU data. The author in [32] proposed the recurrent neural network (RNN) method to propagate state and regress orientation from IMU measurements. Deep learning-based dead reckoning algorithms were provided to address the aforementioned issues owing to the end-to-end learning framework. However, the conventional models including RNNs, CNNs, and transformers have many disadvantages. Specifically, they need complex quadratic time, high memory usage, and inherit limitations of encoder–decoder architecture.

The merits of using deep learning (DL)-based PDR over traditional PDR are due to its robustness and accuracy. Given their ability for generalization without specifying an explicit analytic model, DL approaches are increasingly being used to learn motion from time-series data. The other advantage of DL-based PDR is that it decreases inertial sensor errors in the multi-mode system and is also capable of estimating motion and generating trajectories directly from raw inertial data without any handcrafted engineering. The conventional PDR approaches have difficulties in providing accurate state estimation for long-range distance and are inadequate to sufficiently control error explosion from primitive double integration. Additional limitations are due to complex users' body constraints and motion dynamics. In a departure from other studies which target estimation of the pose of the smartphone collecting the IMU data, here we focus on the scenario where input IMU data are collected from a smartphone while the target estimation pose is that of the user holding the phone.

The end-to-end learning framework for inertial odometry mitigates the challenge associated with IMU produced at high frequencies which is directed to long sequences [33]. The recurrent neural network is an extremely powerful sequence model suitable for the challenge involving sequence processing. Processing the raw IMU using long-range sequence is challenging and vulnerable to washout, while processing using CNN architecture. In order to solve the aforementioned challenges, another model aware preprocessing for compressing the raw IMU for motion measurement is required. After numerous efforts in the indoor positioning and localization communities over the last couple of years, state-of-the-art inertial odometry (IO) algorithms have presented magnificent performance. Due to the disadvantage of computational load, a filtering-based PDR solution approach using MEMS has proven to be highly challenging, and a sliding window-based optimization method and end-to-end learning approach are crucial for position and orientation estimation. In order to increase performance and efficiency, we provide an end-to-end deep learning architecture for inertial sensor modeling. The contribution of this paper is as follows:

- We proposed hybrid CNN and LSTM architecture to learn the spatial and temporal features from the input IMU. We built an end-to-end model designed for pedestrian dead reckoning. Then, from the datasets obtained, the aim was to predict the position and orientation of the pedestrian to estimate its trajectory;
- We presented a 2D LiDAR (position and orientation) dataset with IMU, a groundbreaking dataset for research on inertial sensor-based pedestrian navigation that would both encourage the use of data-driven techniques and serve as a standard reference;
- We demonstrated with the experimental results that our deep learning-based PDR trained only on 6-axis IMU data can also be used for 2D pedestrian trajectories estimation using conventional SHS PDR, therefore comparing the two approaches, this has good generalizability.

The rest of this work is structured as follows: Section 2 describes related works that are closely connected to the research topic of this paper. Section 3 explains the deep learning model's proposed approaches and the architecture of the IMU channel. Section 4 describes the experiment used to validate the estimated output pose, and Section 5 offers

a comparison with other studies. Section 6 presents and discusses the findings of the experiment. Section 7 of the study finishes with conclusions and future research activity.

2. Related Work

Inertial odometry has been studied for both odometry in 3D and classical inertial odometry. The model-based pedestrian dead reckoning (PDR) technique, empowered by either sensor's fixed positions or cyclic motion patterns, is broadly considered a type of self-contained indoor positioning system. However, the model-based PDR approach needs an accurate representation of state estimates with regard to the incoming measurements and is inadequate to preferably control error eruption from unsophisticated double integration and has basic limitations due to complex users' body constraints and motion dynamics.

Chen et al. [34] proposed one of the first data-driven inertial odometry using an inertial measurement unit. Recent studies have used IMU data to train neural networks to learn motion models and output velocity estimates directly from IMU measurements. With the use of body wearable IMU, data-driven models [35] and filtering models [36] have been implemented into several positioning and navigation fields of applications.

There are two indoor positioning approaches utilizing IMU sensors: SHS smartphone-based PDR [37] and using foot-mounted IMU-based PDR [38]. Several studies have significantly improved the efficiency of foot-mounted IMU-based inertial navigation systems. The first is the zero-velocity update (ZUPT), which improves model capacity by exploiting certain motion constraints. The second approach used to improve the efficiency of pose estimation is the zero angular rate update (ZARU), which reduces gyroscope drift. If a pedestrian dead reckoning system is constructed on a handheld smartphone, a zero-velocity update cannot be applied, carrying out the filter-based integrated navigation solution. The body wearable sensor embedded IMU is susceptible to different types of error caused by intrinsic properties. Step detection, heading, stride length estimate, and location updating are the three basic components of smartphone-based PDR [39]. In order to identify step peaks and segment the related inertial data, PDR uses the inertial data's threshold. However, inaccuracies in step length estimation and step detection can still happen, resulting in significant system error drifts. It can be realized that step detection using step peaks methods is still in the preliminary stage of research, and most stride detection is performed on learning-based methods at present. Most step detection currently requires an activity recognition (AR) model and more attention is given to machine learning (ML) approaches [40]. The second component of SHS-PDR, the step length estimation, also has the disadvantages of using traditional methods such as regression-based methods, biomechanical models, and empirical connections which are not appropriate for running because of larger gait parameter variations. Therefore, the smartphone-embedded IMU required for SHS-PDR is susceptible to different types of error caused by intrinsic properties.

Machine learning-based inertial odometry solutions eliminate the need for setting manually during testing and turn the incorporation of inertial navigation into a continuous time-series learning activity. Commonly used methods in INS modeling are adaptive neuro-fuzzy inference systems (ML-based-ANFIS) [41], fuzzy extended Kalman filter (AFEKF) [42], rotational symmetry of pedestrian dynamics [43], and Support vector machine(SVM) [44]. The author in [45] first proposed a mobile stride length estimation system that constrains double integration approaches from a raw foot-mounted IMU using deep convolutional neural networks. Considering the robust zero velocity detector (ZVD) method used in a foot-mounted IMU. The author in [46] employs histogram-based gradient boosting because of its efficiency and achieves comparable results for various types of motion. Many authors [47] focus on orientation estimate which leverages sequential models to include prior information about the sequential nature of IMU signals. The others focus on position estimation which continuously tracks IMU ego-motion and produces IO [48]. Sequence-based IO methods have achieved increasing attention to address the problem of error propagation. This approach breaks the data into an independent window to segment the inertial data. The state such as orientation, velocity, and position are not visible and

drive from the inertial data and propagate with time. In the sequence-based approach, LSTMs are applied to learn temporal correlations between multivariate time series IMU data. In these types of inertial navigation system models, automatically discovered features relevant to the task are used. Author [49] proposed an end –to -end deep learning framework to tackle the inertial attitude estimation problem based on IMU measurements.

3. System Overview and Methodology

3.1. System Overview

General PDR systems based on end-to-end learning frameworks can practically be categorized into two architectures. One is to use the two bi-directional CNN architecture to predict the spatial features. Then CNN is extracting spatial features of the input IMU from the handheld smartphone and provides the feature maps to LSTM. The other is the two bi-directional LSTM models that are employed to capture time series temporal features. A proposed hybrid CNN-LSTM-based PDR approach is used to estimate the position and orientation of pedestrians using the multisensory-equipped backpack device (accelerometer, gyroscope, and backpack LIDAR global pose). The input of our network is a six-axis IMU sequence of pedestrian trajectory in an indoor environment. The output of our network is the pedestrian position and orientation. Figure 1 shows the proposed architecture of CNN-LSTM-based PDR, which mainly consists of two combined modules: 1D CNN and two bidirectional LSTM. CNN and LSTM combination were used to extract effective features for motion estimation. We introduced a recurrent neural network (RNN) that was particularly worthy to problems that require sequence processing. In order to enable RNNs to learn longer-term trends, the LSTM architecture was developed. CNN makes layers that use convolution filters on local features. The basic CNN block is composed of standard pointwise linear functions, nonlinearities, and residual connections. The input IMU sequence is sequentially transformed into smaller and larger sequences through pooling layers.

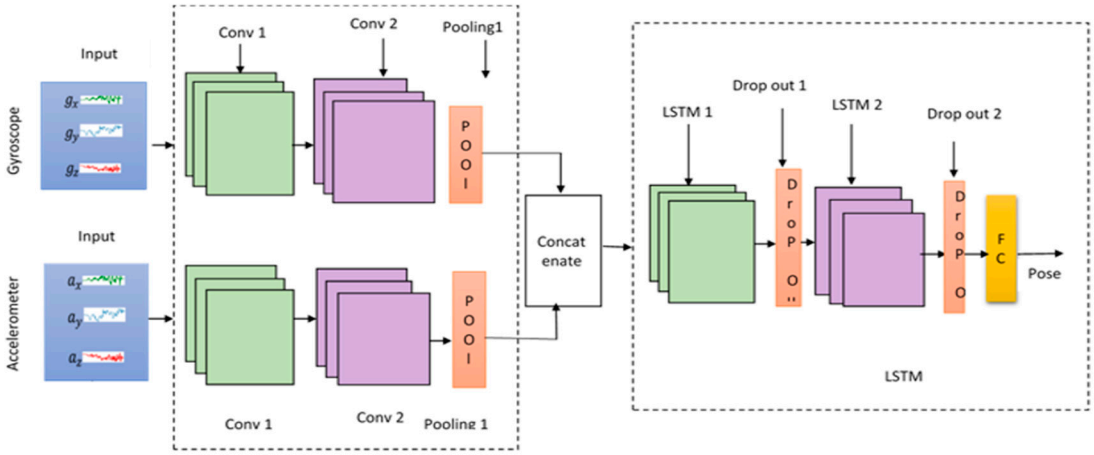
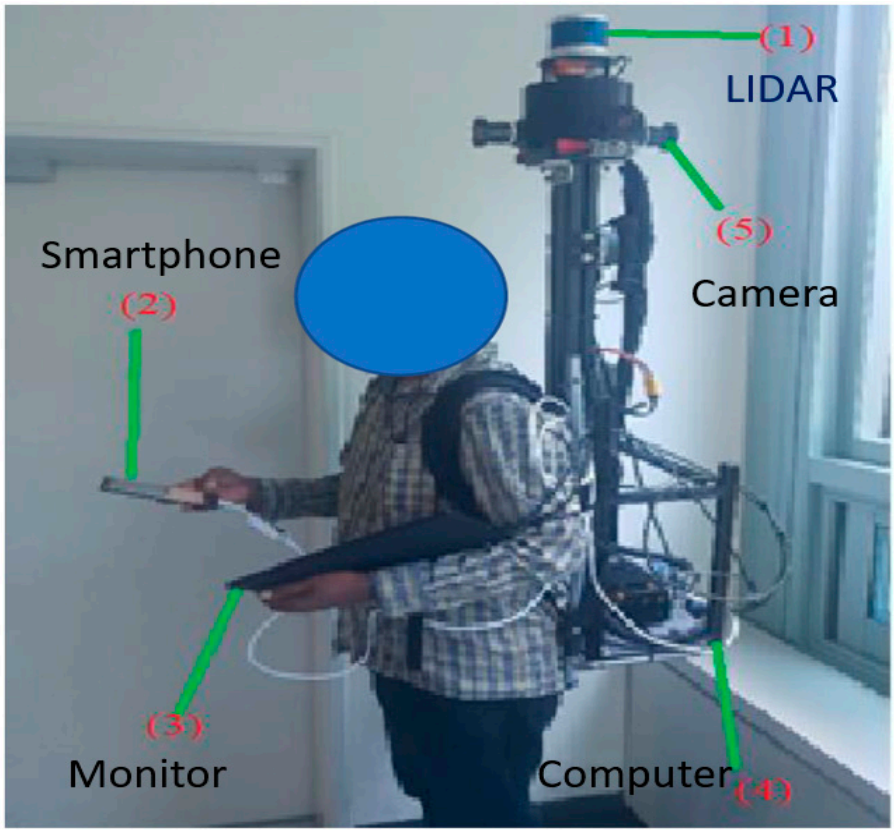
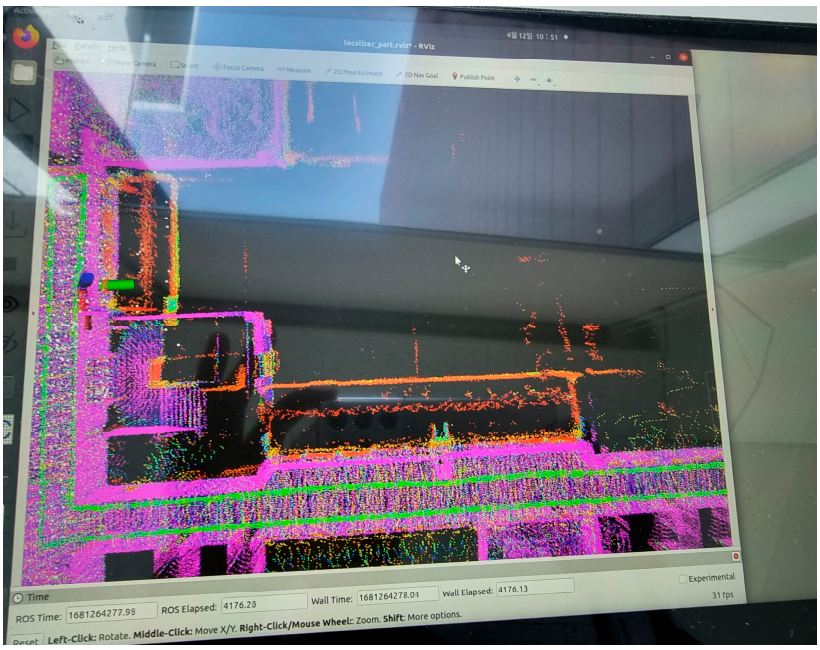


Figure 1. Overview of the proposed CNN-LSTM model.

For deep learning-based pedestrian dead reckoning, there is a cooperation between CNN and LSTM to acquire the benefit of both modules. As shown in Figure 2, the six-axis IMU input sequence of window 200 frames is processed by 1D convolutional layers. It has 11 kernels,128 features with a fixed dimension, and a maximum pooling layer of size 3. The output from concatenated 1D CNN-model fed to the first bidirectional LSTM. Since the future and previous IMU readings have an influence on the relative pose regression, the two-layer LSTM model was used. Hence, the output from the first bidirectional LSTM was the input to the second bidirectional LSTM model. In order to prevent overfitting, the dropout layers with a 0.25 rate were used. Finally, the estimated output relative pose was generated by a fully connected layer.

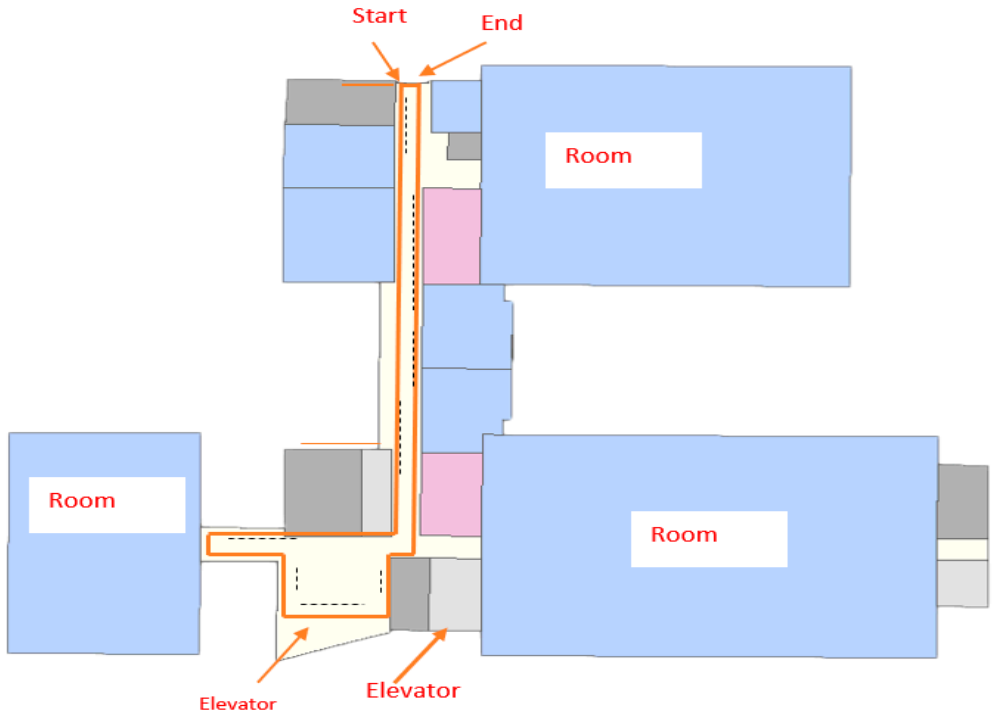


(**a**)

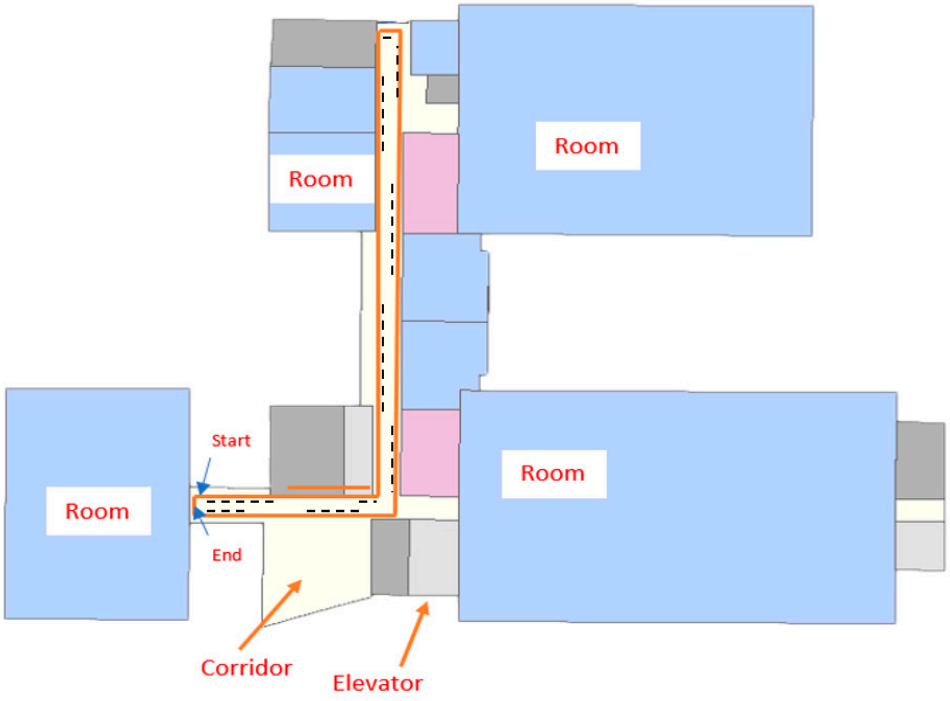


(**b**)

Figure 2. *Cont*.



(**c**)



(**d**)

Figure 2. The device used for the indoor experiment. It contains a multisensorial backpack device used in the indoor environment to collect the dataset. (**a**) User carrying backpack device. (**b**) Monitor for real-time display of backpack positioning system. (**c**) Scenario 1 ground truth plotted. (**d**) Scenario 2 ground truth plotted.

3.2. Six Degrees of Freedom (6DOF) Relative Position and Orientation Representation

A six degrees of freedom (6DOF) relative position and orientation can be described in numerous approaches. The first approach is an extension of the polar coordinate system

proposed in the 3D space by using the spherical coordinate system. The second approach uses 3D distance vectors and a unit quaternion. When dealing with motions in any direction, this characterization grasps the orientation accurately. Thus, by considering the first approach which is spherical coordinate system, the relative position and orientation is obtained by:

$$\begin{cases} x_t = x_{t-1} + \Delta l \,.\sin(\theta_{t-1} + \Delta\theta) \,.\cos(\varnothing_{t-1} + \Delta\varnothing) \\ y_t = y_{t-1} + \Delta l \,.\sin(\theta_{t-1} + \Delta\theta) \,.\cos(\varnothing_{t-1} + \Delta\varnothing) \\ z_t = z_{t-1} + \Delta l \,.\sin(\theta_{t-1} + \Delta\theta) \,.\cos(\varnothing_{t-1} + \Delta\varnothing) \end{cases} \quad (1)$$

where (x_t, y_t, z_t) and $(x_{t-1}, y_{t-1}, z_{t-1})$ are the current position and previous position, respectively. Δl is the traveled distance, $\Delta\theta$ is an inclination change, and $\Delta\varnothing$ is the change in heading.

3.3. Loss Function

The loss function consists of two parts: the position loss (p Loss) and the orientation loss (q loss). The loss for the trajectory value Δp and orientation change Δq are estimated separately due to the proposed model having a different scale. As described in [17], the loss function of the output is trained using multi-task learning. The basic uncomplicated way to estimate the loss for the six degrees of freedom (6DOF) odometry challenge is to consider a consistent weighting of the losses. The total loss function is given in Equation (2).

$$L_{total} = \sum_{i=1}^{n} e^{-\log \sigma_i^2} L_i + \log \sigma_i^2 \quad (2)$$

where σ_i^2 and L_i are the variance and loss function of the ith task. Let the estimated position and orientation be described as follows:

$$\begin{cases} \Delta p = R^T(q_{t-1})(p_t - p_{t-1}) \\ \Delta q = q_{t-1}^* \otimes q_t \end{cases} \quad (3)$$

where relative pose $(\Delta p, \Delta q)$ is calculated from previous and current positions. Orientations p_{t-1}, q_{t-1}, p_t, and q_t are associated with a given IMU data window.

4. Experiment

4.1. Experimental Setup

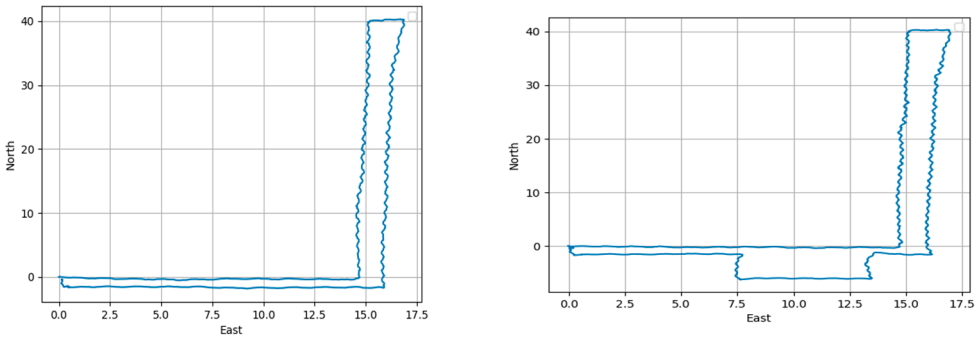
As shown in Figure 3, the whole sensors setup contains a smartphone and LIDAR scanner. We used a Samsung Galaxy S21 Ultra 5G phone to collect the IMU data in indoor scenes to measure the accuracy of the recovered trajectories. Backpacks were equipped with LIDAR equipment. We first conducted experiments to demonstrate the pedestrian dead reckoning problem. The data used in our system were extracted from the backpack device method beforehand.

For each location, a floor-plan and recorded sequence of LIDAR pose was described in Figure 2c,d for scenario 1 and scenario 2, respectively. A sequence of five data points for scenario 1 and a sequence of five data points for scenario 2 were gathered to be processed. To see trajectories through a corridor of our building, we plotted the LIDAR pose which was considered as ground truth as shown in Figure 3a,b.

4.2. Dataset Acquisition and Pre-Processing

In this section, we describe the dataset acquisition, data description, and preprocessing. To obtain the highly accurate neural network model for PDR, the acquisition of reliable dataset is very important. The data were collected using a backpack laser scanner and a handheld smartphone. A detailed description of the dataset is given in Table 1. The collected dataset was carried out along the corridor of ETRI building 12 on the fifth floor on flat ground. To describe the real-world applicability of pose estimation, the dataset was collected by only considering one motion mode, i.e., normal walking by four test subjects at two different scenarios (medium and long distance). The LiDAR pose global coordinate

was first changed to meters, where a Python script was used to convert using the Pyproj transformation coordinate system in the Republic of Korea where the data were collected to WGS84. Data recorded on the IMU inside the smartphone were read out and processed by software written in Python. See Figure 3a,b for the trajectories of the measurements carried out using the backpack scanner. The dataset contains a sequence of data labeled with a LIDAR coordinate absolute position which was converted to meters before being given to the model for training and testing.



(**a**) (**b**)

Figure 3. Scenarios' trajectory plot: (**a**) is scenario 1 plot; (**b**) is scenario 2 plot.

Table 1. Dataset description.

Subject count	4
Age Range	30–49
	1 young adult and 3 older adults
Gender Distribution	Male 3 and female 1
Apparatus	- backpack system with IMU, LIDAR, and camera - handheld smartphone
Sampling frequency	Kinematic data has been captured at 100 Hz
Activities recorded	Over ground normal walking
Description of time elapsed for each sequence of data collected	2 min of walking from each 4 different people

Table 1 shows the description of inertial odometry datasets collected with four different subjects—1 female and 3 males between the ages of 30 and 49. Each trajectory is about 2 min of normal walking along a corridor in a building as illustrated in Figure 2c,d. IMU data from the smartphone was used as input data for the model that tries to estimate the positions recorded by the backpack positioning system as shown in Figure 2a,b.

4.3. Model Training Details

The models were implemented on Python. For the compilation of our model built, Python 3.8 and Cuda 10.2 were used. We trained the CNN-LSTM model for self-collected datasets. The training tasks were done for 100 epochs, as we noticed that the performance did not improve with further training. We implemented the framework algorithms with Tensor Flow 2.3.0 and Keras 2.4.3, using the Adam optimizer with a learning rate of 0.0001. The computations were performed on Microsoft Windows 10.0 with GPU NVIDIA GeForce GTX TITAN X. All models were trained for a batch size of 64 samples. The datasets were split using a window size of 200 and a stride of 10. The window size determines the length of lookback IMU readings used to predict the difference between position and orientation at the beginning and the end of the stride interval. We split our data set into training and

validation datasets with a 9-to-1 ratio. After training was completed, the model saved during the training session with the best validation loss was used for testing.

4.4. Hyper-Parameter Tuning

Tuning the hyperparameters is a very important step in training neural network models. During the tests, the batch size was determined by the limits of the GPU memory. The window size and stride depend on the sampling rate of the sensor data and ground truth positions, and were selected after a preliminary search within a range similar to other studies. At this stage, we were concerned with selecting a model size with sufficient capacity to learn the task on this dataset. Reducing the model size to be able to run well on mobile devices was considered at a later stage. We tested two configurations with 128 and 192 as the number of filters, and 1,159,689 and 2,599,689 as the number of parameters correspondingly. The larger model achieved a slightly better minimum validation loss of -12.65 but the testing errors were slightly worse than the smaller model. Above certain model sizes, it seems that performance is not very sensitive to an additional increase in the number of parameters. Therefore, we used the smaller model size in the rest of the experiments.

To determine an appropriate optimizer and learning rate, we performed a grid search with Adam, SGD, and RMSprop optimizers and 1×10^{-4}, 3×10^{-4}, 6×10^{-4} and 1×10^{-3} learning rates. The training results are visualized in Figure 4. We selected 200 epochs for training, confirming that this was sufficient to observe the flattening of the loss curves. We can see that each optimizer approached a specific loss value that was not sensitive to the learning rate. The SGD optimizer performed worse than Adam and RMSprop which approached similar loss values. The Adam optimizer slightly outperformed RMSprop. The learning rate affected how fast we approached the minimum loss value.

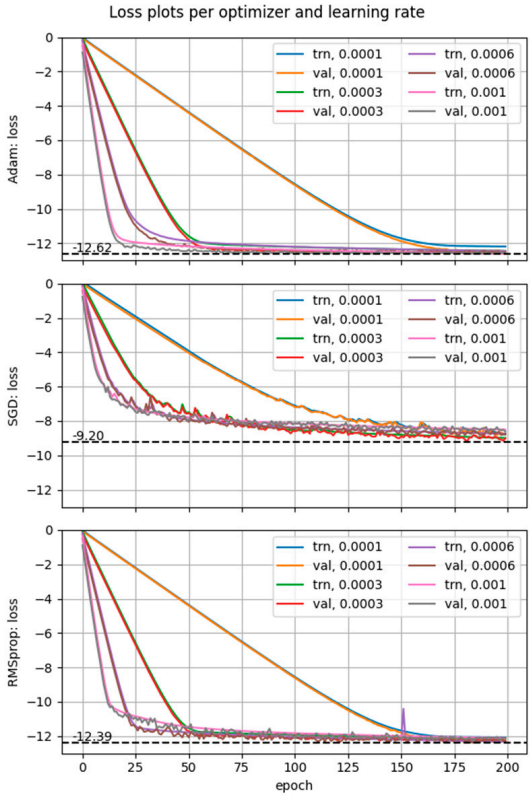


Figure 4. Loss plot per optimizer vs. learning rate.

These results motivated us to use the Adam optimizer with a learning rate 1×10^{-4} in the later experiments as indicated in Table 2. Since we used the lower learning rate, we increased the number of training epochs to 500 to ensure sufficient time for the loss to decay.

Table 2. CNN-LSTM parameterization and experimental environment.

Parameter	Value
Number of filters	128
Filter size	11
Optimizer	Adam
Epochs	100,500
Learning rate	1×10^{-4}
Number of blocks	6
Drop out	25×10^{-2}
CPU	2.6 GHz 6-Core Intel Core i7
Platform	Python

Increasing the learning rate also resulted in a noisier loss curve at low loss values. From Table 3, we can confirm that the Adam optimizer with a learning rate of 1×10^{-3} achieved a minimum validation loss of -12.62. Second was Adam with a learning rate 3×10^{-4} and a validation loss of -12.58, but with the advantage of having a smoother loss curve.

Table 3. Minimum validation loss achieved by each combination of optimizer and learning rate.

	Min Validation Loss		
Learning Rate	**Adam**	**SGD**	**RMSprop**
1×10^{-4}	-12.47	-8.84	-12.33
3×10^{-4}	-12.58	-9.20	-12.30
6×10^{-4}	-12.56	-8.88	-12.39
1×10^{-3}	-12.62	-8.63	-12.24

Table 4 summarizes the test errors averaged across all test trajectories for each learning rate and optimizer combination. Since we are interested in the position estimation accuracy, we reported the standard for the task *RMSE* and *MAE* measures. For completeness, we added the R^2 metric too. As expected, the models trained with Adam and RMSprop optimizers significantly outperformed the models trained with SGD. Adam optimizer models again performed slightly better than RMSprop models. Surprisingly, the minimum *MAE* and *RMSE* errors were achieved by Adam with a learning rate 1×10^{-4}, despite a validation loss for learning rates of 1×10^{-3} and 3×10^{-4} being lower. The reported mean R^2 metric values for the Adam optimizer were affected by bad trajectory results in two out of eleven trajectories.

Table 4. Comparison of the test errors for different learning rate and optimizer combinations. Test error metrics averaged across all test trajectories for each learning rate and optimizer combination.

Optimizer	Learning Rate	*MAE*	*RMSE*	*RMSE_x*	*RMSE_y*	R^2_x	R^2_y
Adam	1×10^{-4}	3.86	4.47	2.15	2.76	0.92	-3.33
	3×10^{-4}	5.75	6.66	2.98	4.08	0.81	-7.09
	6×10^{-4}	4.00	4.56	2.43	2.33	0.91	0.06
	1×10^{-3}	4.45	5.00	2.34	1.96	0.92	-0.29
SGD	1×10^{-4}	29.72	33.79	22.68	14.63	0.94	-0.32
	3×10^{-4}	27.14	30.35	19.48	17.86	0.87	-1.69
	6×10^{-4}	26.79	29.29	19.13	15.97	0.89	0.46
	1×10^{-3}	30.30	34.82	18.95	19.03	-0.16	-22.559
RMSprop	1×10^{-4}	4.36	4.83	1.97	2.47	-2.72	-38.338
	3×10^{-4}	4.84	5.50	2.81	3.50	-1.86	-22.61
	6×10^{-4}	4.56	5.16	2.47	1.94	-2.69	-18.86
	1×10^{-3}	17.04	20.76	13.53	12.22	-3.52	-32.82

4.5. Evaluation of Model Performance

Deep learning architecture was explored to improve accuracy. Using a low-cost IMU-embedded smartphone and LiDAR scanner data, qualitative and quantitative analyses were performed for evaluation.

Root mean square error (RMSE): The *RMSE* was used as a standard statistical metric to measure model performance. The underlying assumption when presenting the *RMSE* was that the errors were unbiased and followed a normal distribution. In the pose prediction stage, a loss function was employed to determine the predicted error over the training samples. This error shows where the output was different from what was anticipated. To investigate the system performance on the tradeoff between ground truth and estimated position, we defined a cost function that covered the *RMSE* metrics as follows.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left|(p_i - \hat{p}_i)^2\right|}{N}} \tag{4}$$

where n is the reading timestamps, p_i is the actual position, and $\hat{p}_i$ is the predicted position.

Mean absolute error (MAE): The *MAE* is another useful measure widely used in model evaluations. The *MAE* is widely used to evaluate the performance of a guidance or navigation system and represents the estimated position's global accuracy. The *MAE* for a regression problem is the average of the absolute difference between the predicted and actual values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\|p_i - \hat{p}_i\|^2 \tag{5}$$

where p_i is the actual value, $\hat{p}_i$ is the predicted value, and n is the number of forecast values. These metrics are error rates; thus, a lower *MAE* result represents a more accurate prediction.

4.6. Experimental Results and Analysis

Four users conducted experiments using five time sequences of IMU data collected by each individual, which were trained with the same hyperparameter settings in both the testing and training phases. The *RMSE* described in Section 4.5 was used for evaluation purposes. Table 4 shows the empirical results of each user in detail. Table 4 shows *RMSE*, which is better to evaluate the performance of the model. Compared with the traditional step and heading system-based pedestrian dead reckoning method, our end-to-end model on the lightweight 6-axis IMU provided an important enhancement.

Table 5 summarizes the different subject walk sequence evaluation results of the proposed method. The position estimation accuracy in each sequence was summarized with *RMSE*. The best result with the lowest *RMSE* was achieved on the female collected data, user 2. For the first scenario dataset (Table 5), the estimated *RMSE* varied between a maximum of 1.99 and a minimum of 0.51, and, similarly, estimated *MAE* varied between a maximum of 1.56095 and a minimum of 0.1210. From the table, the female subject *RMSE* and *MAE* could be assumed to be very small even if all users intended to perform the experiment in the same walking mode.

Table 5. The *RMSE* and *MAE* comparison for each sequence measurement by each user (scenario 1).

Sequence	Subject							
	User 1		User 2		User 3		User 4	
	RMSE	***MAE***	***RMSE***	***MAE***	***RMSE***	***MAE***	***RMSE***	***MAE***
01	0.98	1.0637	0.84	0.0495	1.34	0.9614	1.82	0.9207
02	1.96	0.1210	0.51	0.0682	1.64	1.4756	1.69	1.4594
03	1.16	1.0484	0.54	0.0333	1.32	0.1385	0.87	0.7786
04	1.99	1.1553	0.74	0.0496	0.68	1.3156	1.51	1.56095
05	1.92	0.1851	0.67	0.1371	0.51	0.9278	1.13	1.0377

Figure 5 shows the box plot for the *RMSE* mean of each user in Table 5. The average root means square error ranged from 0.51 to 1.61 m for the three datasets. As it is demonstrated in Table 5, the lowest error mean was user 1. The comparison was between the users' collected sequence in scenario 1 with the ground truth provided by the LIDAR dataset. The *RMSE* of user 1 was the largest because user one was the tallest among those subjects asked to collect the dataset, and his stride was very fast during normal walking, while the second user was the only female who participated in dataset collection and her *MAE* was relatively very small.

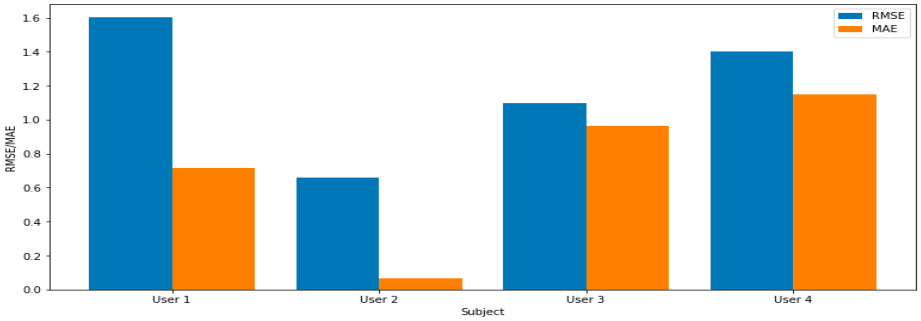


Figure 5. *RMSE* vs. *MAE* of different users for scenario 1.

Table 6 summarizes the different subject walk sequence evaluation results of the proposed methods. We evaluated the *RMSE* mean and *MAE* mean of each sequence to demonstrate the efficiency of each sequence. The best result with the lowest *RMSE* was the user 2 who was a female collecting data. For the first scenario dataset (Table 5), the estimated *RMSE* varied between a maximum of 0.346 and a minimum of 0.258, and similarly the estimated *MAE* varied between a maximum of 0.999 and a minimum of 0.0761.

Table 6. The *RMSE* and *MAE* comparison for each sequence measurement by each user (scenario 2).

Sequence	Subject							
	User 1		User 2		User 3		User 4	
	RMSE	*MAE*	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*	*MAE*
01	0.8961	0.0761	0.87128	0.0963	0.2053	0.0359	0.205345	0.2669
02	0.461	0.2275	0.48741	0.0867	0.2206	0.0945	0.220611	0.2302
03	0.237	0.1963	0.15797	0.2314	0.1579	0.2143	0.119556	0.1192
04	0.48741	0.3235	0.2139	0.0420	0.21398	0.0869	0.085189	0.3687
05	0.999	0.0413	0.22340	0.1530	0.22347	0.0746	0.125689	0.16309

Figure 6 shows a boxplot summarizing the *RMSE* and *MAE* value for different subject. The box plot provides the visualization of statistical estimated position max-min.

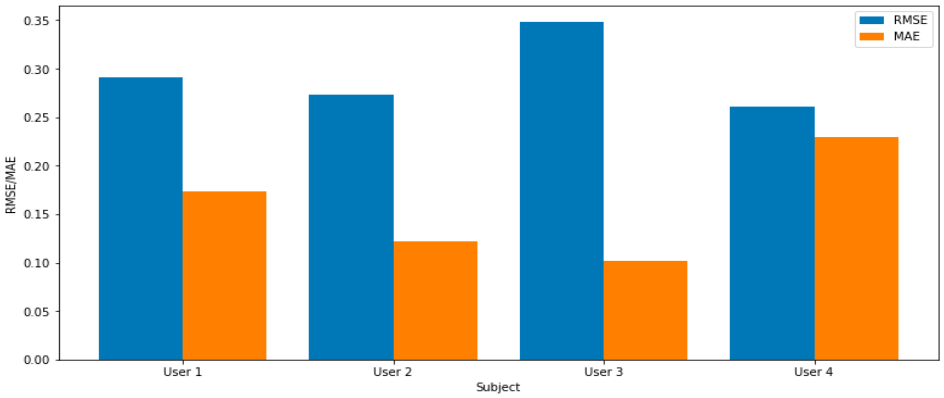


Figure 6. *RMSE* vs. *MAE* of different users for scenario 2.

Figure 7 illustrates the estimated RMSEs of each coordinate x, y, and z, and the total trajectory. The coordinates for x, y, and z showed increased errors because the user direction changed. The test trajectories' length along the x-axis was twice the length along the y-axis, which might be related to the observed large *RMSE*.

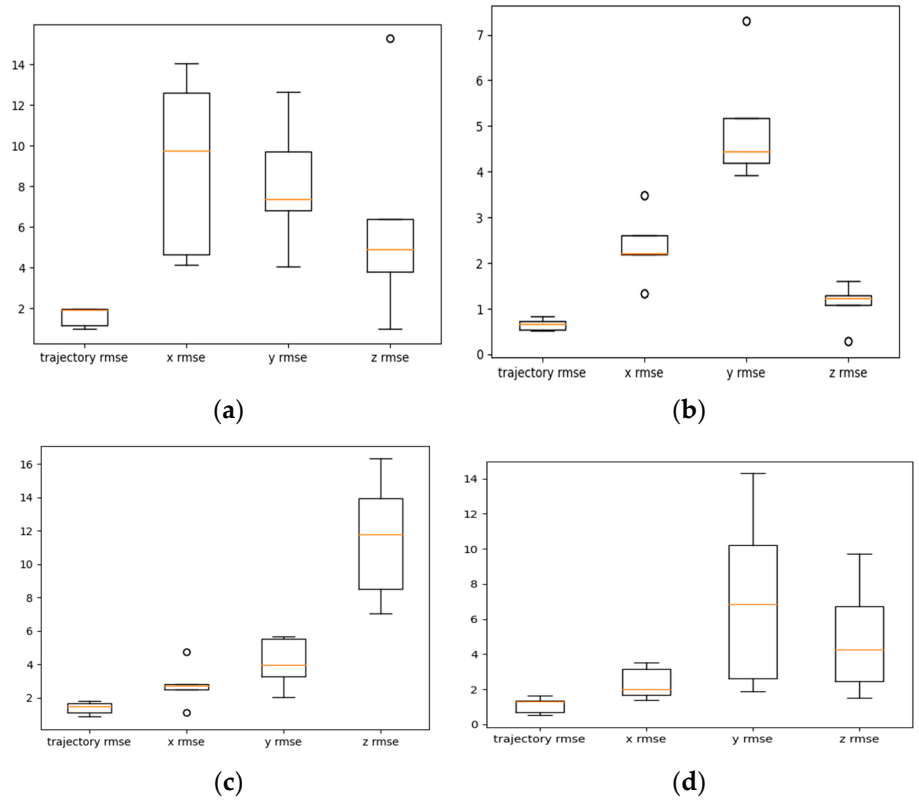


Figure 7. Boxplot of user trajectory RMSEs with CNN-LSTM model where (**a**) is user 1; (**b**) is user 2; (**c**) is user 3; (**d**) is user 4.

The trajectory of each sequence pose estimated by the model in our datasets was shown in Figure 8.

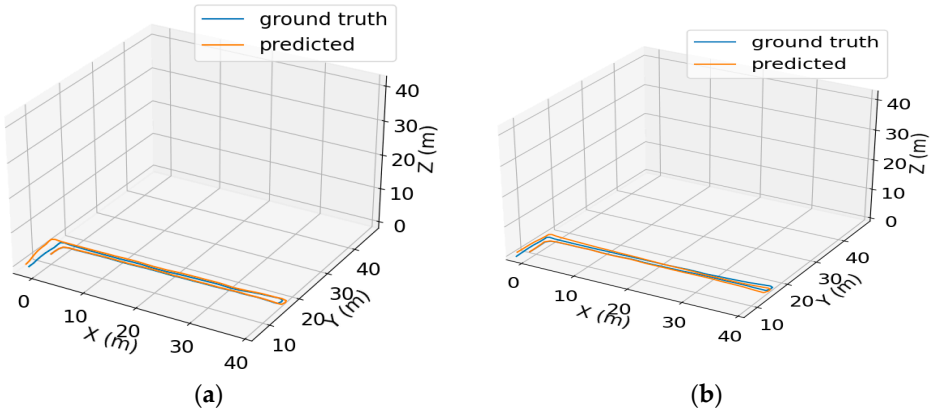


Figure 8. *Cont.*

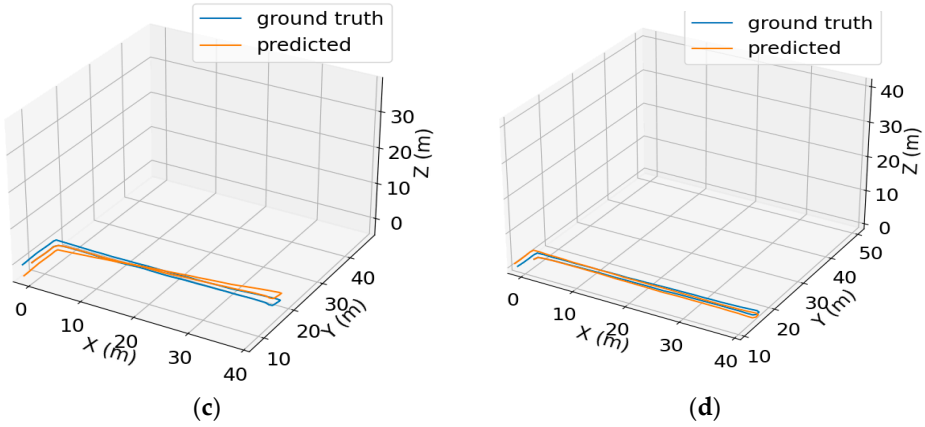


Figure 8. The estimated trajectory of the proposed method for scenario 1: (**a**) user 1; (**b**) user 2; (**c**) user 3; (**d**) user 4.

Figure 9 shows a comparison of each user's trajectory of scenario 2. For our results, we displayed the root mean square error for each sequence from the four runs plotted in Figure 9a–d.

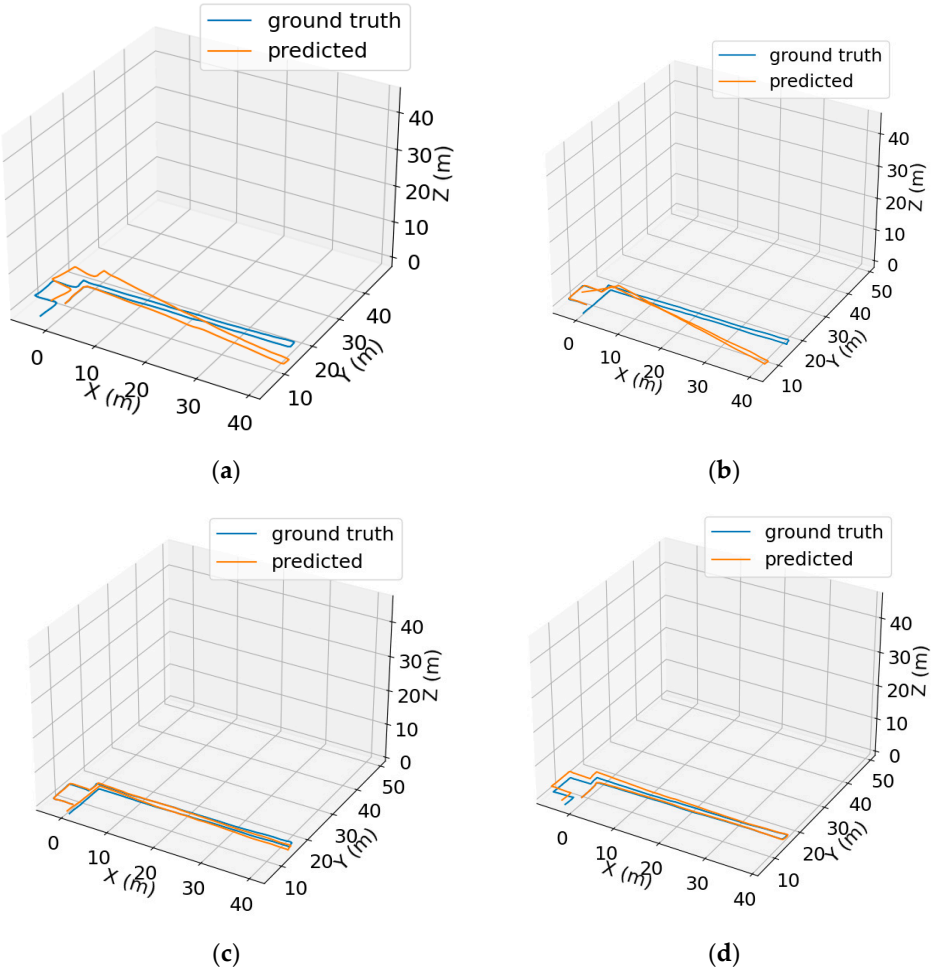


Figure 9. The estimated trajectory of the proposed method for scenario 2 (**a**) user 1 (**b**) user 2 (**c**) user 3 (**d**) user 4.

Figure 10 shows the sample sequence 01 of user 1. The orientations shown here were achieved by converting the quaternion to a Euler angle. As indicated in Figure 10, the orientation impacted the position error by using the phone orientation and the ground

truth orientation. We can see that the orientation module improved the performance of all other position models (quite significantly for CNN-LSTM), but it also nearly reached the theoretical maximum performance where ground truth orientations were directly provided.

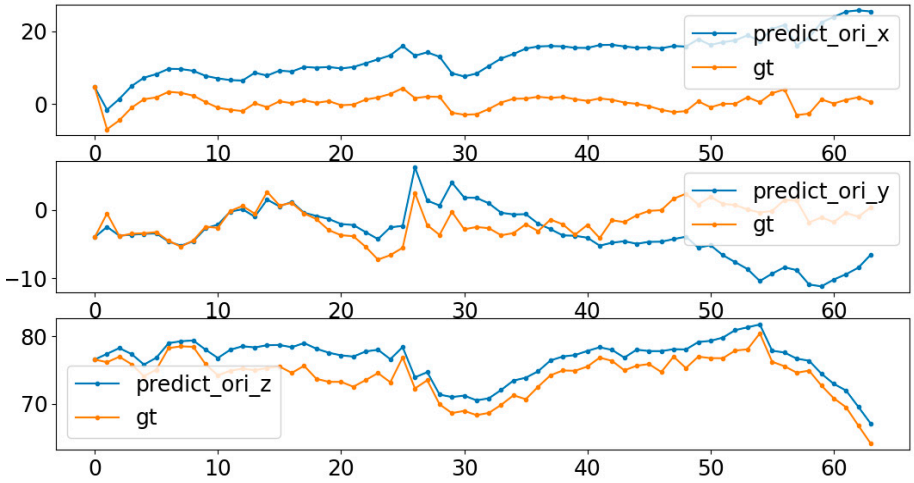
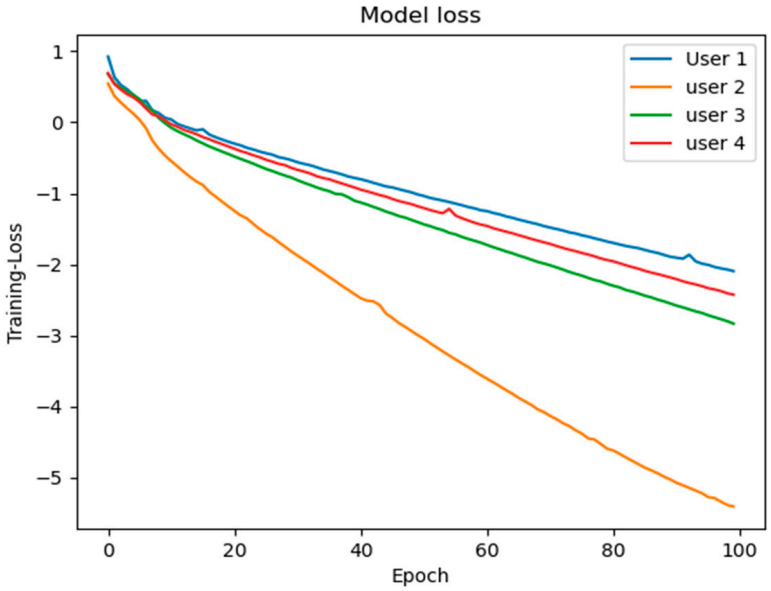


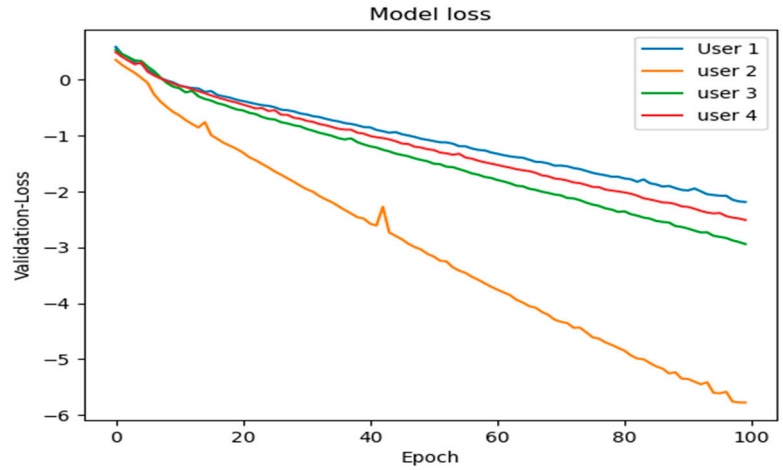
Figure 10. Sample relative orientation obtained from user 1, sequence 01.

The training loss and validation loss vs. epochs in Figure 11 show how the epochs were distributed among the different users. The training loss and validation loss for large epochs were not smooth because of the variation in the learning rate. There was the probability of finding bad data for learning in large epochs and batch sizes. Even though the training loss and validation loss were very smooth in 100 epochs, the trajectory estimation was not fairly consistent within each user. The results of the various user's specific validation and learning rate calculations were used to select the optimal number of epochs and sensitivity of parameters for the learning model. First, we picked out the suitable number of epochs for model training. Figure 11c,d shows the training and validation loss results for the model. In each case, the training and validation loss of the model reached its lowest level when epochs between 100 and 500 were used.

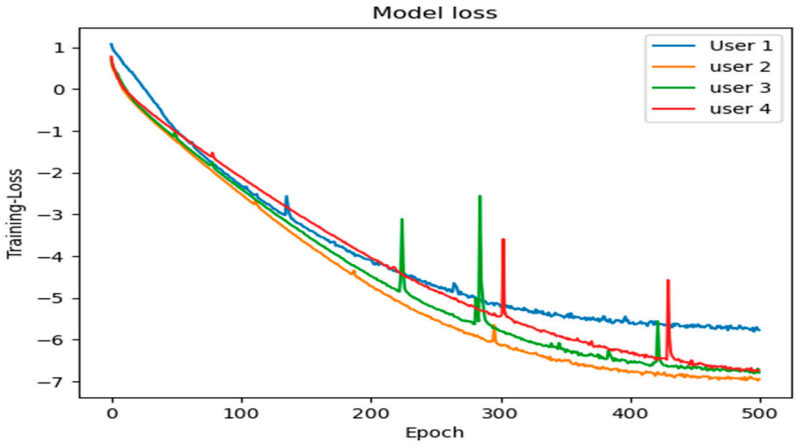


(**a**)

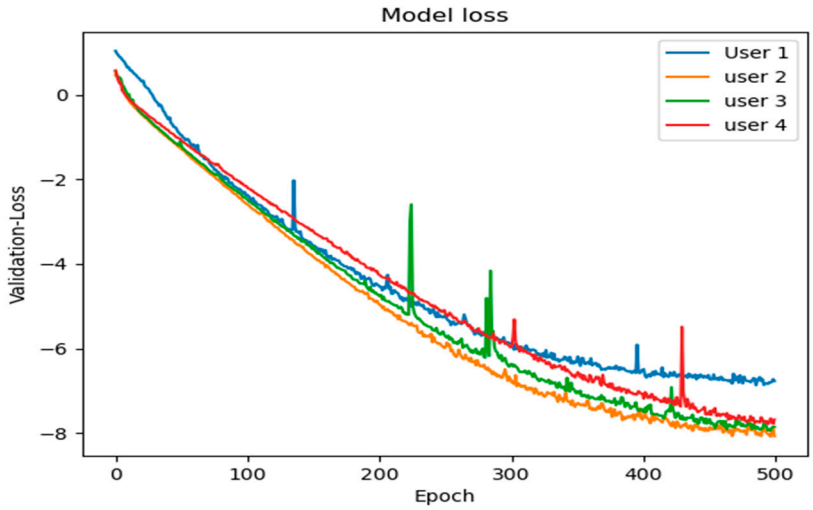
Figure 11. *Cont.*



(**b**)



(**c**)



(**d**)

Figure 11. Each user's training and validation loss on selected epochs for the model. (**a**) Training loss for 100 epochs; (**b**) validation loss for 100 epochs; (**c**) training loss for 500 epochs; (**d**) validation loss for 500 epochs.

4.7. Conventional PDR Trajectory Estimation

By taking the sample trajectory from the collected sequence of IMU data for scenario 2, we estimated the position using conventional SHS-based PDR. First, we estimated the step and step length using biomechanical methods. Second, we used the attitude and heading reference system-based heading estimation. Finally, positions were estimated by multiplying the step length with heading estimation techniques.

Figure 12 shows the trajectory of the estimated position of conventional SHS PDR. The Blue line is the estimated position and the yellow line is the Ground truth from the LIDAR pose.

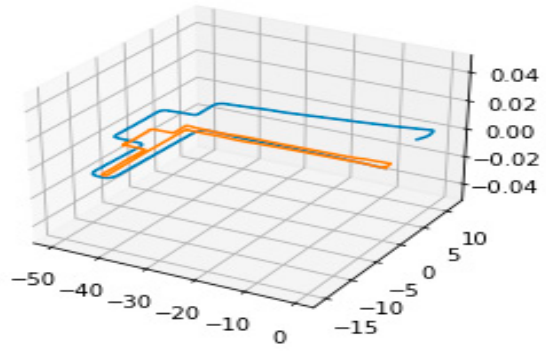


Figure 12. The trajectory of conventional SHS-PDR. The return position error of the two scenarios evaluated for long-range distance for comparison.

The heading estimation has influences on the position estimation accuracy. Thus, it is very important to select an algorithm with better heading estimator accuracy to simulate the process of the conventional pedestrian dead reckoning. Figure 13 compares error accumulation as the trajectory increases in the traditional PDR SHS method. The CNN-LSTM PDR was used to compromise the resulting position drift.

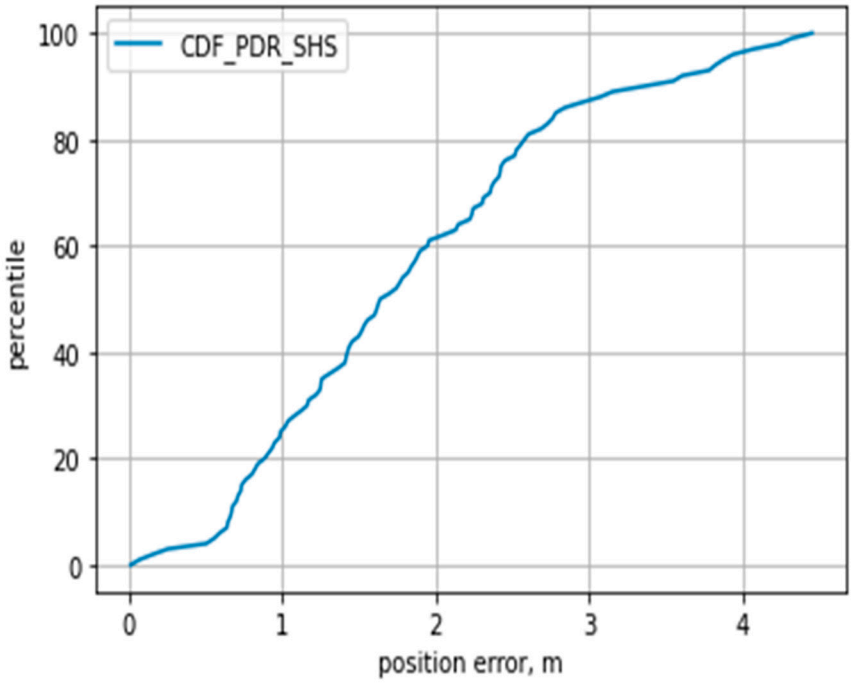


Figure 13. Cumulative distribution function (CDF) of the conventional SHS PDR horizontal position error. (The office building scenario 2).

Figure 13 shows the CDF of the location errors in the experiment corresponding to one of the motion types and scenario 2. We evaluated the horizontal position error results using the CDF to compare with the degree of divergence within a very short distance. It showed that the CDF of the 2D horizontal position error increased with time. At the same time, it was found that the probability of the error range was more than 80% within 3 m as indicated in Figure 13. The position error range was greatly reduced by the use of the deep learning-based PDR algorithm, which improved the system's position performance.

This work was inspired by state-of-the-art deep learning that has been used to improve PDR. Previous efforts in the study of PDR for indoor positioning were done using the model-based method. These approaches commonly include the INS and SHS methods. However, employing each method for indoor positioning has the limitation of error drifts. Nevertheless, currently, there are data-driven methods that constrain system error drift to predict the position and orientation of pedestrians without any handcrafted engineering. In this work, therefore, we presented a learning end-to-end inertial odometry to outperform previous model-based approaches.

We used a combination of CNN-LSTM to extract features of a multi-channel low-cost IMU from backpack devices. A CNN and LSTM combination was used to extract effective features for motion estimation. In addition, we evaluated the traditional SHS-based PDR that relied on a three-heading estimation mechanism to compare the performance between the two PDR approaches. As a result, we found that a CNN-LSTM-based PDR accuracy improvement was obtained for the PDR compared with SHS PDR. To the best of our knowledge, there were no previous efforts to use either aforementioned method to evaluate qualitatively and quantitatively method using a self-collected odometry dataset. Finally, this study concludes that the deep learning-based PDR outperforms the traditional PDR.

5. Comparison with Other Studies

Deep learning-based PDR is an active topic of research and the previous studies provide a base for performance comparison on one hand and hints for unexplored sub-problems worth pursuing. Previous studies have selected comparison baselines from conventional PDR such as SHS-PDR and INS-based PDR. In this section, we compare our result with other contemporary PDR methods based on machine learning and deep learning techniques with reference to the accuracy observed for the odometry dataset in the context of trajectory length.

Table 7 shows that methods based on the hybrid CNN-SVM showed significantly a better performance than using CNN method alone, based on the accuracy values. In our case, a hybrid CNN-LSTM network trained on different users showed that in most of the cases, *MAE* and *RMSE* were less than traditional PDR methods as shown in Table 8. It is important to notice that, a small error here demonstrated a significant improvement compared to the PDR result. Looking at the average *MAE* and *RMSE* on all users showed that there was an improvement in the value of *MAE*. Regarding *RMSE*, the effect of considerable noise was not too much, and still, the *RMSE* of each user was less than existing methods.

Table 7. Comparison of CNN-LSTM localization accuracy with traditional PDR trajectory (first scenario).

Method	*MAE* Error	*RMSE* Error
SHS PDR	2.021	3.558
CNN-LSTM PDR	0.0237	0.48741

As an additional qualitative comparison, in Figure 14 we show the ground truth and estimated trajectories on a dataset from the public repository provided by [17]. Our results are shown in the right column and are obtained from a model trained with an Adam optimizer with a learning rate of 3×10^{-4} (see Table 4).

Table 8. Comparison of different deep learning-based and machine learning techniques used for IMU data-based PDR.

No.	Methods	*RMSE* [m]	Trajectory Length [m]	Type
1	Regression network [50]	2.11	Rectangular route	ML
2	Convolutional neural network [51]	3.42	670	DL
3	DNN-SAN-LTE-ZUPT [52]	1.97	600	DL
4	DualCNN-LSTM [53]	3.83	70	DL
5	ours	0.52	120	DL

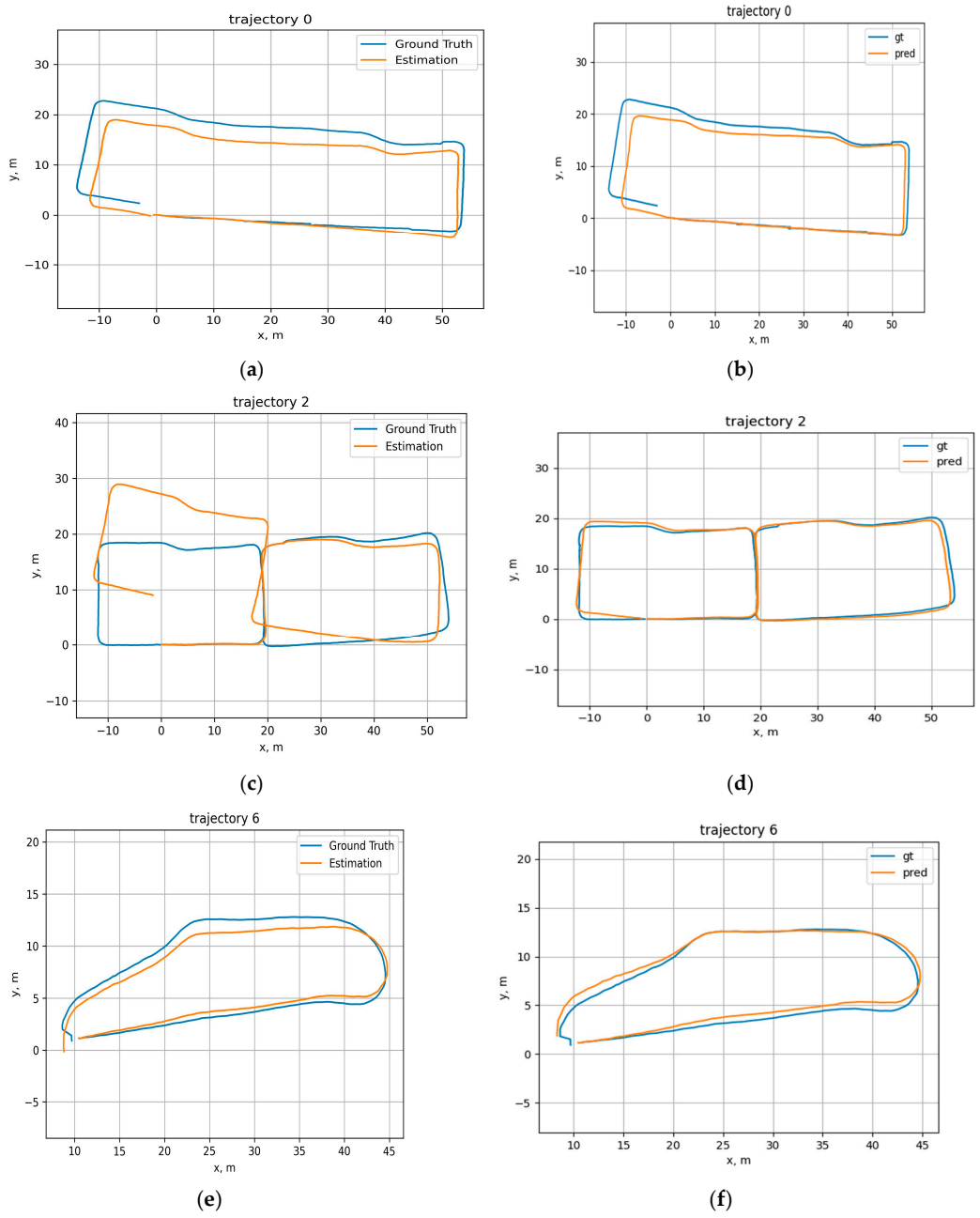


Figure 14. Ground truth and estimated trajectories on a dataset from the public repository provided by [17]. (**a**,**c**,**e**) shows estimated trajectory with normal training data. (**b**,**d**,**f**) shows the estimated trajectory with tuned better optimizer and learning rate. The result in the left column is from the publicly released model [17]. Our results are shown in the right column and are obtained from a model trained with an Adam optimizer with a learning rate of 3×10^{-4} (see Table 4).

6. Discussion

In this paper, we used a backpack LiDAR system with laser scanners and a handheld smartphone IMU to illustrate its ability for enhancing the efficiency of a deep learning-based robust positioning PDR system. Collecting a low-cost inertial measurement unit (IMU) from a handheld smartphone and backpack LIDAR pose is a crucial method for estimating navigation solutions and using them as the ground truth, respectively. While the acquisition of such data in indoor environments is relatively difficult to obtain GPS signal denied environment, it is very important to attain ground truth for the pedestrian dead reckoning algorithm evaluation. We also introduced a new dataset to encourage the community to adapt deep learning-based PDR method evaluation for indoor environments when external observers are unavailable. The proposed approach was trained and tested on our smartphone dataset from a backpack multisensorial system collected by five individuals. We inherited settings from 2-layer BiLSTM for the position and orientation estimation mechanisms. The input was a sequence of IMU-form handheld smartphone measurements and LiDAR poses for ground truth in the world frame. We evaluated CNN-LSTM on five individually collected datasets against conventional filtering-based research works.

By comparing our data-driven approach results with model-based PDR or filtering-based PDR without any pre-installed infrastructure research, we found that a CNN-LSTM neural network accuracy improvement was obtained for the PDR compared with conventional SHS PDR. These results showed that the proposed deep learning-based pedestrian dead reckoning methodology is a useful tool for the position and orientation method for IMUs captured from handheld smartphones and backpacks. Often times in many scenarios, it is reasonable for IMUs to be noisy, thereby providing imperfect data. To determine the accuracy of the two approaches in our scenarios and evaluate the influence of the neural network, we prepared two datasets with inertial data (consisting of accelerometer, gyroscope, and LIDAR pose data).

7. Conclusions

The experiments described in this paper aimed to evaluate the deep learning-based pedestrian dead reckoning under discussion across the different sequences of smartphone-embedded IMU and LiDAR ground truth pose data. We evaluated how a deep learning, hybrid CNN-LSTM system is able to efficiently estimate an accurate position even when working under a largely GPS-denied environment. A hybrid CNN-LSTM model with end-to-end training mechanisms for pedestrian dead reckoning is proposed. The proposed pedestrian dead reckoning-based indoor positioning system using CNN and LSTM algorithms extracted the spatial and temporal features from the input IMU data. The CNN module was used to extract spatial features from the IMU measurements, followed by the two bi-directional LSTM with SoftMax scoring alignment, which was applied to further capture the temporal features. Our experiments validated the effectiveness of the proposed CNN-LSTM-based PDR in terms of pose estimation accuracy.

The limitation of this study is that only normal walking was considered. Other modes of motion such as side stepping, running, climbing, and descending stairs are to be considered at a later stage. In addition to unconstrained smartphone-based PDR, in our future work, we will consider the addition of a self-attention mechanism to increase robustness by mitigating noise spikes and missing measurements, and by improving generalization over a variety of smartphone models. Another promising approach to consider is structured state-space sequence modeling that targets solving the problem of long-range dependencies. It will help in capturing rich building interior contexts and improve performance on trajectories specific to the given building.

Author Contributions: Conceptualization, F.W. and V.B.I.; methodology, F.W.; software, V.B.I.; validation, V.B.I., S.L. and S.P.; formal analysis, F.W.; investigation, V.B.I.; resources, S.P.; data curation, S.L.; writing—original draft preparation, F.W.; writing—review and editing, S.L.; visualization, S.P.; supervision, S.P.; project administration, S.P.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported and funded by the ETRI Research and Development Support Program of MSIT/IITP, Republic of Korea. [Project Title: Development of Beyond X-verse Core Technology for Hyper-realistic Interactions by Synchronizing the Real World and Virtual Space/Project Number:RS−2023−00216821].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

Abbreviations

MU	Inertial measurement unit
PDR	Pedestrian dead reckoning
MEMS	Micro-electromechanical system
LIDAR	Light detection and ranging
INS	Inertial navigation system
SHS	Step-and-heading system
GPS	Global positioning system
RMSE	Root mean squared error
MAE	Mean absolute error
CNN	Convolutional neural network
LSTM	Long short-term memory
DL	Deep learning
ML	Machine learning
ZARU	Zero angular rate update

References

1. Zhou, B.; Tu, W.; Mai, K.; Xue, W.; Ma, W.; Li, Q. A novel access point placement method for wifi fingerprinting considering existing aps. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 1799–1802. [CrossRef]
2. Kim, J.W.; Jang, B. Workload-aware indoor positioning data collection via local differential privacy. *IEEE Commun. Lett.* **2019**, *23*, 1352–1356. [CrossRef]
3. Zhuang, Y.; Syed, Z.; Georgy, J.; El-Sheimy, N. Autonomous smartphone-based WiFi positioning system by using access points localization and crowdsourcing. *Pervasive Mob. Comput.* **2015**, *18*, 118–136. [CrossRef]
4. Paul, A.S.; Wan, E.A. Wi-Fi based indoor localization and tracking using sigma-point Kalman filtering methods. In Proceedings of the 2008 IEEE/ION Position, Location and Navigation Symposium, Monterey, CA, USA, 5–8 May 2008; pp. 646–659.
5. Husen, M.N.; Lee, S. Indoor location sensing with invariant Wi-Fi received signal strength fingerprinting. *Sensors* **2016**, *16*, 1898. [CrossRef] [PubMed]
6. Saab, S.S.; Nakad, Z.S. A standalone RFID indoor positioning system using passive tags. *IEEE Trans. Ind. Electron.* **2011**, *58*, 1961–1970. [CrossRef]
7. House, S.; Connell, S.; Milligan, I.; Austin, D.; Hayes, T.L.; Chiang, P. Indoor localization using pedestrian dead reckoning updated with RFID-based fiducials. In Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, MA, USA, 30 August–3 September 2011; pp. 7598–7601.
8. Ruiz, A.R.J.; Granja, F.S.; Honorato, J.C.P.; Rosas, J.I.G. Accurate pedestrian indoor navigation by tightly coupling foot-mounted IMU and RFID measurements. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 178–189. [CrossRef]
9. Huang, J.; Yu, X.; Wang, Y.; Xiao, X. An integrated wireless wearable sensor system for posture recognition and indoor localization. *Sensors* **2016**, *16*, 1825. [CrossRef]
10. Zhou, Y.; Law, C.L.; Guan, Y.L.; Chin, F. Indoor elliptical localization based on asynchronous UWB range measurement. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 248–257. [CrossRef]

11. Jimenez, A.; Seco, F. Comparing decawave and bespoon UWB location systems: Indoor/outdoor performance analysis. In Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcala de Henares, Spain, 4–7 October 2016; pp. 1–18.
12. Mainetti, L.; Patrono, L.; Sergi, I. A survey on indoor positioning systems. In Proceedings of the 2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 17–19 September 2014; pp. 111–120.
13. Wu, Z.; Wen, M.; Peng, G.; Tang, X.; Wang, D. Magnetic-assisted initialization for infrastructure-free mobile robot localization. In Proceedings of the 2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), Bangkok, Thailand, 18–20 November 2019; pp. 518–523.
14. Blanco, J.L.; González, J.; Fernández-Madrigal, J.A. Mobile robot ego-motion estimation by proprioceptive sensor fusion. In Proceedings of the 2007 IEEE 9th International Symposium on Signal Processing and Its Applications, Sharjah, United Arab Emirates, 12–15 February 2007.
15. Kelly, J.; Saripalli, S.; Sukhatme, G.S. Combined visual and inertial navigation for an unmanned aerial vehicle. In *Field and Service Robotics: Results of the 6th International Conference*; Springer: Berlin/Heidelberg, Germany, 2008.
16. Chen, C. Learning Methods for Robust Localization. Ph.D. Thesis, University of Oxford, Oxford, UK, 2020.
17. Silva do Monte Lima, J.P.; Uchiyama, H.; Taniguchi, R.I. End-to-end learning framework for imu-based 6-dof odometry. *Sensors* **2019**, *19*, 3777. [CrossRef]
18. Chen, C.; Lu, C.X.; Wahlstrom, J.; Markham, A.; Trigoni, N. Deep neural network based inertial odometry using low-cost inertial measurement units. *IEEE Trans. Mob. Comput.* **2019**, *20*, 1351–1364. [CrossRef]
19. Scaramuzza, D.; Fraundorfer, F. Visual odometry [tutorial]. *IEEE Robot. Autom. Mag. (RAM)* **2011**, *18*, 80–92. [CrossRef]
20. Groves, P. Navigation using inertial sensors [tutorial]. *IEEE Aerosp. Electron. Syst. Mag.* **2015**, *30*, 42–69. [CrossRef]
21. Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings of the Proceedings Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152.
22. Wang, X.; Jiang, M.; Guo, Z.; Hu, N.; Sun, Z.; Liu, J. An indoor positioning method for smartphones using landmarks and PDR. *Sensors* **2016**, *16*, 2135. [CrossRef] [PubMed]
23. El-Sheimy, N.; Hou, H.; Niu, X. Analysis and modeling of inertial sensors using Allan variance. *IEEE Trans. Instrum. Meas.* **2007**, *57*, 140–149. [CrossRef]
24. Haq, M.A. CDLSTM: A novel model for climate change forecasting. *Comput. Mater. Contin.* **2022**, *71*, 2363–2381.
25. Haq, M.A.; Rahim Khan, M.A.; AL-Harbi, T. Development of PCCNN-Based Network Intrusion Detection System for EDGE Computing. *Comput. Mater. Contin.* **2022**, *71*, 1769–1788. [CrossRef]
26. Haq, M.A.; Rahim Khan, M.A. DNNBoT: Deep neural network-based botnet detection and classification. *Comput. Mater. Contin.* **2022**, *71*, 1729–1750.
27. Chen, B.; Zhang, R.; Wang, S.; Zhang, L.; Liu, Y. Deep-Learning-Based Inertial Odometry for Pedestrian Tracking Using Attention Mechanism and Res2Net Module. *IEEE Sens. Lett.* **2022**, *6*, 6003804. [CrossRef]
28. Hannink, J.; Kautz, T.; Pasluosta, C.F.; Barth, J.; Schulein, S.; Gassmann, K.-G.; Klucken, J.; Eskofier, B.M. Mobile Stride Length Estimation with Deep Convolutional Neural Networks. *IEEE J. Biomed. Health Inform.* **2017**, *22*, 354–362. [CrossRef]
29. Wu, X.; Zhao, L.; Guo, S.; Zhang, L. Pedestrian inertial navigation based on CNN-SVM gait recognition algorithm. *J. Phys. Conf. Series* **2021**, *1903*, 012043. [CrossRef]
30. Cao, X.; Zhou, C.; Zeng, D.; Wang, Y. RIO: Rotation-equivariance supervised learning of robust inertial odometry. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 6614–6623.
31. Wang, Y.; Cheng, H.; Meng, M.Q.H. A2DIO: Attention-Driven Deep Inertial Odometry for Pedestrian Localization based on 6D IMU. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 819–825.
32. Sun, S.; Melamed, D.; Kitani, K. Idol: Inertial deep orientation-estimation and localization. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 6128–6137.
33. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* **2018**, *4*, e00938. [CrossRef] [PubMed]
34. Chen, C.; Lu, X.; Markham, A.; Trigoni, N. Ionet: Learning to cure the curse of drift in inertial odometry. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
35. Klein, I. Data-driven meets navigation: Concepts, models, and experimental validation. In Proceedings of the 2022 DGON Inertial Sensors and Systems (ISS), Braunschweig, Germany, 13–14 September 2022; pp. 1–21.
36. Wang, C.; Bo, Y.; Jiang, C. A New Efficient Filtering Model for GPS/SINS Ultratight Integration System. *Math. Probl. Eng.* **2020**, *2020*, 9158185. [CrossRef]
37. Harle, R. A survey of indoor inertial positioning systems for pedestrians. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1281–1293. [CrossRef]
38. Jiménez, A.R.; Seco, F.; Zampella, F.; Prieto, J.C.; Guevara, J. PDR with a foot-mounted IMU and ramp detection. *Sensors* **2011**, *11*, 9393–9410. [CrossRef] [PubMed]

39. Ho, N.-H.; Truong, P.H.; Jeong, G.-M. Step-detection and adaptive step-length estimation for pedestrian dead-reckoning at various walking speeds using a smartphone. *Sensors* **2016**, *16*, 1423. [CrossRef]
40. Lee, M.-W.; Khan, A.M.; Kim, J.-H.; Cho, Y.-S.; Kim, T.-S. A single tri-axial accelerometer-based real-time personal life log system capable of human activity recognition and exercise information generation. *Pers. Ubiquitous Comput.* **2011**, *15*, 887–898. [CrossRef]
41. Mahdi, A.E.; Azouz, A.; Abdalla, A.E.; Abosekeen, A. A machine learning approach for an improved inertial navigation system solution. *Sensors* **2022**, *22*, 1687. [CrossRef]
42. Sabzevari, D.; Chatraei, A. INS/GPS Sensor Fusion based on Adaptive Fuzzy EKF with Sensitivity to Disturbances. *IET Radar Sonar Navig.* **2021**, *15*, 1535–1549. [CrossRef]
43. Wahlstrom, J.; Kok, M. Three Symmetries for Data-Driven Pedestrian Inertial Navigation. *IEEE Sens. J.* **2022**, *22*, 5797–5805. [CrossRef]
44. Grekov, A.N.; Kabanov, A.A.; Alekseev, S.Y. Support Vector Machine for Determining Euler Angles in an Inertial Navigation System. *arxiv* **2022**, arXiv:2212.03550.
45. Li, Y.; Zeng, G.; Wang, L.; Tan, K. Accurate Stride-Length Estimation Based on LT-StrideNet for Pedestrian Dead Reckoning Using a Shank-Mounted Sensor. *Micromachines* **2023**, *14*, 1170. [CrossRef] [PubMed]
46. Kone, Y.; Zhu, N.; Renaudin, V.; Ortiz, M. Machine learning-based zero-velocity detection for inertial pedestrian navigation. *IEEE Sens. J.* **2020**, *20*, 12343–12353. [CrossRef]
47. Zhao, L.; Pingali, G.; Carlbom, I. Real-time head orientation estimation using neural networks. In Proceedings of the International Conference on Image Processing, Rochester, NY, USA, 22–25 September 2002; Volume 1, p. I.
48. Deng, J.; Xu, Q.; Ren, A.; Duan, Y.; Zahid, A.; Abbasi, Q.H. Abbasi, Machine Learning Driven Method for Indoor Positioning Using Inertial Measurement Unit. In Proceedings of the 2020 International Conference on UK-China Emerging Technologies (UCET), Glasgow, UK, 20–21 August 2020; pp. 1–4.
49. Golroudbari, A.A.; Sabour, M.H. End-to-end deep learning framework for real-time inertial attitude estimation using 6DoF IMU. *arXiv* **2023**, arXiv:2302.06037.
50. Asraf, O.; Shama, F.; Klein, I. PDRNet: A deep-learning pedestrian dead reckoning framework. *IEEE Sens. J.* **2021**, *22*, 4932–4939. [CrossRef]
51. Lin, C.Y.; Lu, Y.E.; Huang, C.H.; Chiang, K.W. A Cnn-Speed-Based Gnss/Pdr Integrated System For Smartwatch. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2023**, *48*, 235–241. [CrossRef]
52. Abdallah, A.A.; Jao, C.-S.; Kassas, Z.M.; Shkel, A.M. A pedestrian indoor navigation system using deep-learning-aided cellular signals and ZUPT-aided foot-mounted IMUs. *IEEE Sens. J.* **2021**, *22*, 5188–5198. [CrossRef]
53. Kawaguchi, N.; Nozaki, J.; Yoshida, T.; Hiroi, K.; Yonezawa, T.; Kaji, K. End-to-end walking speed estimation method for smartphone PDR using DualCNN-LSTM. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation, Pisa, Italy, 30 September–3 October 2019.

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.