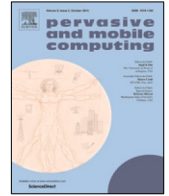




Contents lists available at ScienceDirect

## Pervasive and Mobile Computing

journal homepage: [www.elsevier.com/locate/pmc](http://www.elsevier.com/locate/pmc)

# Passive infrared sensor dataset and deep learning models for device-free indoor localization and tracking

Kan Ngamakeur<sup>a,\*</sup>, Sira Yongchareon<sup>a,1</sup>, Jian Yu<sup>a</sup>, Saiful Islam<sup>b,1</sup>

<sup>a</sup> Department of Computer Science and Software Engineering, Auckland University of Technology, Auckland, 1010, New Zealand

<sup>b</sup> School of Information and Communication Technology, Griffith University, Australia

## ARTICLE INFO

### Article history:

Received 18 May 2022

Received in revised form 26 September 2022

Accepted 28 October 2022

Available online 14 November 2022

### Keywords:

PIR

Location estimation

Indoor localization

Device-free localization

PIR dataset

## ABSTRACT

Location estimation or localization is one of the key components in IoT applications such as remote health monitoring and smart homes. Amongst device-free localization technologies, passive infrared (PIR) sensors are one of the promising options due to their low cost, low energy consumption, and good accuracy. However, most of the existing systems are complexly designed and difficult to deploy in real life, in addition, there is no public dataset available for researchers to benchmark their proposed localization and tracking methods. In this paper, we propose a system and a dataset collected from our PIR system consisting of commercial-of-the-shelf (COTS) sensors without any modification. Our dataset includes profile data of 36 classes that have over 1,000 samples of different walking directions and test data consisting of multiple scenarios with a sequence length of over 2,000 timesteps. To evaluate our system and dataset, we implement various deep learning methods such as CNN, RNN, and CNN-RNN. Our results prove the applicability and feasibility of our system and illustrate the viability of deep learning methods for PIR-based localization and tracking. We also show that our dataset can be converted for coordinate estimation so that deep learning methods and particle filter approaches can be applied to estimate coordinates. As a result, the best performer achieves a distance error of 0.25 m.

© 2022 Elsevier B.V. All rights reserved.

## 1. Introduction

Location is one of the key information for many IoT applications such as remote health monitoring, smart building, energy management, and security management [1,2]. It is so important that location information must be precise and processed promptly to ensure that critical incidents such as medical and security alerts can be handled efficiently without any delay [2–4]. Over the past decade, various surveillance technologies have been developed to provide location estimation in an indoor environment. Video camera offers sequential images of persons but can be used only in a public area due to privacy concerns. Wearable and mobile devices, such as smartwatches, smartphones, etc., can be employed to estimate a person's location but they are considered intrusive and uncomfortable because a person is required to wear or hold such devices all the time [5]. On the other hand, passive sensing technology such as Wifi (e.g., [6]) and PIR sensors can overcome the drawbacks of intrusive monitoring devices and provide accurate localization and tracking.

\* Corresponding author.

E-mail addresses: [kan.ngamakeur@aut.ac.nz](mailto:kan.ngamakeur@aut.ac.nz) (K. Ngamakeur), [sira.yongchareon@aut.ac.nz](mailto:sira.yongchareon@aut.ac.nz) (S. Yongchareon), [jian.yu@aut.ac.nz](mailto:jian.yu@aut.ac.nz) (J. Yu), [saiful.islam@griffith.edu.au](mailto:saiful.islam@griffith.edu.au) (S. Islam).

<sup>1</sup> Senior Member IEEE.

PIR sensors and thermopile array sensors require very low energy in a unit of  $\mu\text{W}$  and are not affected by electromagnetic interference [7]. Although these kinds of sensors have a low to moderate coverage distance and are susceptible to changes in their surrounding environment, the accuracy of most infrared sensor systems is still within one meter thus making the infrared technology quite attractive. Most of the existing PIR-based systems utilized binary outputs for analyzing human movements and locations and they rely on sophisticated sensor nodes or a huge number of sensors that are impractical to be set up in real-life settings [8–10]. Some other works leveraged characteristics of the PIR's analog outputs [11] to reduce the number of sensors that are used for location estimation [12–14], however, the existing systems are still too complex and difficult to deploy in the real world. Most importantly, there are no datasets made publicly available for researchers to reproduce the results or benchmark their proposed method with the others.

In this paper, we propose a compact PIR device built using commercial-of-the-shelf (COTS) PIR sensors and a dataset to address the above challenges. Our dataset contains a set of profile data that have over 1,000 walking samples in different directions and a set of test data that contains both simple and complex scenarios and has more than 2,000 timesteps for each scenario. The data is collected in a controlled environment without any specific modification to ensure the correctness and unbiasedness of the collected data. For localization techniques, we study various deep learning methods that can achieve promising results based on RF signals [15–17]. With well-trained deep learning models, we can achieve an average of 77% accuracy and maximum accuracy of 85%. In addition, we show that both deep learning methods and particle filters can achieve accurate localization with an averaged distance error of 0.4 meters and a minimum distance error of 0.25 m.

We believe that our work facilitates and benefits researchers that are interested in PIR-based localization and tracking by providing our ready-to-use dataset with ground truths to explore new challenges and develop new techniques while being able to evaluate and compare their techniques with existing techniques. We summarize our contributions as follows.

- We propose a compact PIR-based localization system that is low-cost and easy to install.
- We provide a comprehensive PIR dataset that is publically available for researchers to reproduce the results and benchmark their methods.
- We empirically evaluate our dataset based on well-known deep learning models and particle filters to show the applicability of the dataset.

The rest of this paper is organized as follows. Section 2 discusses our related works. Section 3 describes the proposed system, the data collection process and the characteristics of our dataset. Section 4 discusses the procedure and candidate methods that we use to evaluate and validate our dataset. The results of the experiments are discussed in Section 5. Finally, the summary of our work is presented in Section 6.

## 2. Related work

In this section, we review existing works for device-free location estimation for non-infrared-based technology and infrared-based technology in Section A and Section B, respectively.

### 2.1. Non-infrared technology

Radiofrequency (RF) technology such as Zigbee/IEEE 802.15.4, WiFi, RFID, UWB, and mmWave can be employed as promising solutions. The main advantage of RF technology is that it can cover a long distance and can penetrate a wall [18]. Zigbee/IEEE 802.15.4-based, and RFID-based solutions require a dense deployment of transmitter nodes and receiver nodes, which makes these solutions impractical in real-life conditions [19]. The common localization techniques for both Zigbee and RFID technology are the imaging method [20–22], proximity method [23], and model-based method [4,24–27]. WiFi-based localization systems are in the mainstream because we can reuse existing WiFi routers to perform localization or location estimation and take advantage of their large coverage. Thus, equipment and installation costs are significantly reduced [28].

Recently, many researchers promoted the use of WiFi channel state information (CSI) because it can provide more information and features than radio signal strength (RSS) [29–31]. However, CSI is accessible to some specific routers [17]. Thus, The WiFi CSI-based solutions may be less favorable because new hardware must be invested. The main drawback of WiFi-based solutions is that they need to have a constant supply of energy to maintain their coverage [5]. As a result, they incur a high energy cost. Fingerprinting methods [32–34] are commonly used in WiFi-based solutions. Ultra-wide bandwidth (UWB) is not interfered with by a conventional narrowband and carrier wave transmission in the same frequency band and has superior time resolution, which allows a static person and a moving person to be differentiated [35]. Thus, UWB is also suitable for locating a person.

mmWave-based localization takes advantage of mmWave's high-frequency radio wave and directional beamforming to provide very accurate location estimation [36,37]. However, mmWave devices are very expensive, difficult to procure and have a high energy usage. Both UWB and mmWave solutions usually use the triangulation-based technique to localize a person [36,38]. Besides RF technology, capacitive floor mats can be employed for location estimation. This type of technology can achieve accuracy of location estimation within 1 meter, but the installation of floor mats can be difficult if an area is already fully furnished [39].

Acoustic-based localization can estimate a location of a person accurately by detecting a footstep and measuring the angle of arrival (AOA) between a footstep and a microphone [40,41] but it may be susceptible to background noise. On the other hand, inertial sensors are adopted in [4] to measure human lower limb locomotion. Thus, the orientation and position of a person can be estimated. However, errors in inertial sensor measurements could be accumulated over time and affect localization accuracy.

Visible light-based localization utilizes light sources and light sensors. When a person is located within the vicinity of light sensors, these light sensors can measure changes in light level or illuminance passively. As a result, we can infer a person's location by analyzing measurements from these light sensors [42]. One advantage of VL-based localization is that we can leverage the existing lighting infrastructure. In addition, it is immune to RF interference. However, it cannot operate without a light source and visible light can be obstructed by walls.

## 2.2. Infrared technology

Passive infrared technology can be employed to detect a movement of a person or an object. In general, passive infrared devices such as passive infrared (PIR) sensors and thermopile sensors are commonly used for this task. The basic principle of both types of sensors is that a movement of a person causes changes in temperature on the sensing elements of both sensors and an electric current is generated due to the thermal-to-electric conversion of the sensing element's material. Then, the electric current is converted to voltage or binary as the final output. By utilizing voltage outputs (a.k.a. analog outputs) and binary output, it is possible to adopt both PIR and thermopile sensors for location estimation. Compared with other technologies, both passive infrared sensors consume much less energy, usually in a unit of  $\mu\text{W}$ , have a low price, and are simple to employ.

Although the infrared sensors are less accurate than some RF technologies such mmWave, they still can achieve the distance error within a sub-meter unit. For these reasons, PIR and thermopile sensors are also viable options for location estimation in an indoor environment such as home and office buildings. There are a substantial number of papers that proposed passive infrared-based location estimation.

We classify these papers according to their sensor deployment including *floor-mounted*, *wall-mounted*, and *ceiling-mounted*.

For the floor-mounted category, Hao, et al. [43] developed infrared sensor modules consisting of 8 PIR sensors to detect a movement in 360 degrees. FOVs of 8 sensors are overlapped to form 16 detection regions and a visibility coding scheme is used to associate these regions to the detection of sensors. In their experiment, 4 sensor nodes were placed at each side of a  $9\text{ m} \times 9\text{ m}$  monitored area. Particle filtering and Kalman filtering are employed to track a person, which achieved an approximate distance error of 1 meter. Yang, et al. [44] used a similar approach and developed a PIR node consisting of six sensors that are arranged to form 12 overlapped detection zones around a sensor node. 9 sensor nodes are deployed uniformly on a  $10\text{ m} \times 10\text{ m}$  area and they obtained an average distance error of 0.5 m. Narayana, et al. [11] analyzed characteristics of PIR analog output that are relating to location estimation and develop a sensor tower consisting of 4 sensors for height classification and location estimation. The distance between a person and the sensor tower can be estimated by adjusting variable gains for different PIR to create various detection regions. They placed two sensor towers perpendicular to each other to monitor an area of  $8\text{ m} \times 8\text{ m}$  and achieve a distance error of 0.3 m.

For the wall-mounted category, Kemper and Hauschildt [45] utilized 4 thermopile sensors and deployed 4 of them at all corners of a  $4.9\text{ m} \times 6.2\text{ m}$  area. The Probability Hypothesis Density (PHD) filter was implemented to perform location estimation and obtained a mean distance error of 0.25 m. Mukhopadhyay, et al. [12] adopted the peak-to-peak amplitude measurement from [11] and proposed two models based on hyperbolic function and Piecewise linear function to express a relationship between the peak-to-peak amplitude and the distance from a sensor. Then, they used multi-lateration and Support Vector Regression (SVR) based techniques to estimate a location of a person. They set up four PIR sensors on each side of a  $7\text{ m} \times 7.5\text{ m}$  area and obtained a distance error of 0.65 m. Alternatively, Liu, et al. [13] proposed azimuth change measurement which is the difference in the azimuth of a person with respect to a PIR sensor at two locations and can be easily determined by a law of cosine. They set up a monitored area of  $8\text{ m} \times 8\text{ m}$  that has 4 sensors located at 4 corners of the area. Particle filtering was employed to determine a person's location and achieve an accuracy of 0.67 m. Wei-Han and Hsi-Pin [46] mounted two thermopile sensors that are located 3.3 meters away from each other and are oriented at a 30-degree angle on a wall to monitor an area of  $2.35\text{ m} \times 3\text{ m}$ . The foreground of the human body is extracted by subtracting background images. Then, the angle of arrival (AOA) is determined using the foreground from two sensors. The location is estimated by AOA based positioning method. Then, a regression model is used to improve the estimation accuracy. Thus, the mean distance error is achieved at 0.13 m. Narayana, et al. [47] developed a custom sensor module called LOCI consisting of one thermopile and one PIR sensor. By combining both sensors, 3-dimension locations can be estimated. The thermopile can provide location information across the FoV cone axis and the distance between the sensor module and a person is determined by a PIR sensor. Machine learning models, i.e., KNN, can be trained to estimate a possible location of a person. Only one sensor is mounted on wall at 1.2 meters height to monitor a  $9\text{ m} \times 8\text{ m}$  area and they obtain the best accuracy at 0.12 m.

For the ceiling-mounted category, Yang, et al. [48] deployed 9 PIR sensors to cover an entire apartment. A practical binary sensor model and an accessibility map were proposed to improve the estimation of the particle filter. As a result, They achieved the distance error within 0.6 m. Tao, et al. [49] used a binary sensor network consisting of 43 ceiling-based PIR sensors to monitor an office room with a size of  $15.0\text{ m} \times 8.5\text{ m}$ . Information such as desk's locations and

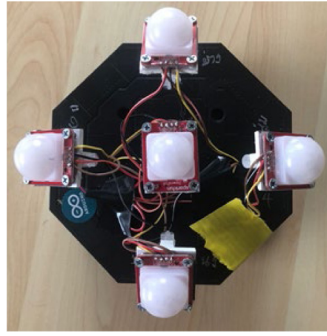


Fig. 1. A photo of developed PIR sensor node.

the moving directions to improve location estimation. Wu, et al. [14] developed the rotationally shuttered PIR (Ro-PIR) sensor to enable the detection of a stationary person. The shuttered PIR sensor is mounted at a height of 2.8 meters and can cover a circle area with a radius of 2 m. They adopted machine learning algorithm such as KNN and SVM to localize and track a person and achieved the accuracy within 0.44 m.

Based on the existing work, there are abundant public localization & tracking datasets for video camera and radio frequency technology such as the UCI wireless indoor localization dataset [50], WiFi RSSI indoor localization dataset [51], etc. These public datasets allow new researchers to improve or invent different techniques at ease, which accelerates several improvements in localization and tracking for their respective technologies. For passive infrared technology, a substantial number of works have been proposed but none of these passive infrared works has made their datasets available to the public. There are still some challenges related to certain areas, such as signal processing, feature engineering, signal modeling, etc., that should be further explored. However, it is not possible due to the lack of a dataset. Therefore, we construct a public PIR-based location estimation dataset for researchers to explore and address challenges.

### 3. Our proposed system and dataset

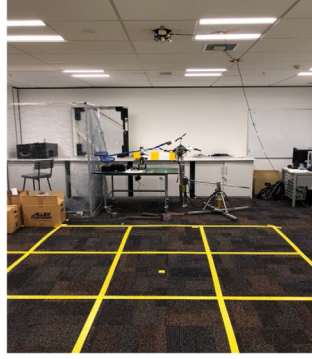
In this section, we discuss our proposed PIR-based localization system and a process for constructing our PIR dataset. Section A introduces our hardware and its setup in an environment. Section B discusses the process of collecting training data and testing data. Last, Section C explains the description and format of collected PIR data.

#### 3.1. Hardware and environment setup

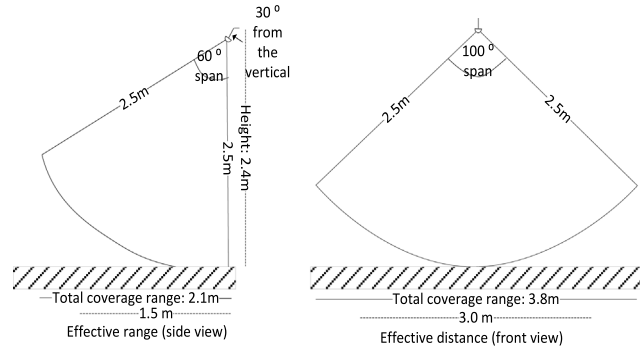
There are three types of sensor deployment: floor-mounted, wall-mounted, and ceiling-mounted. A floor-mounted sensor node is quite simple to deploy in indoor and outdoor areas. However, indoor locations, especially offices or houses, can be cluttered with furniture that obstructs the FOV of a floor-mounted node. In addition, a floor-mounted sensor node can cause tripping hazards. A preoccupied room can lead to difficulty in a wall-mounted sensor deployment. For example, furniture such as a wardrobe can block an installation point. Furthermore, the size and shape of a room can add additional difficulty, especially when we need to arrange sensors in a specific way. In contrast, a ceiling-mounted node does not face these issues. Although ceiling heights may vary for different buildings, we can adjust a detection range of a PIR sensor to address such a variable. For this reason, in this research, we choose a *ceiling-based sensor node*.

We studied various designs of a PIR sensor node from existing works. The hardware designs similar to [7,45] require the dense deployment of PIR sensor nodes and they are not practical in real-life. Some works such as [40,49] need to apply complicated masking, which results in a bulky node and is overly complex. We focus on simplifying a design of a PIR sensor node to be more compact and with improved area coverage. For these reasons, we develop a ceiling-mounted sensor node consisting of 5 SEN-13968 PIR sensors (Model: SEN-13968) from SparkFun. We arrange 5 PIR sensors as shown in Fig. 1. The middle sensor points perpendicularly to the ground and the surrounding sensors point outwards at 30 degrees from vertical as shown in Fig. 3. The front FOV and side FOV of each PIR sensor span at 100 degrees and 60 degrees respectively. As a result, overlapping detection zones are formed as illustrated in Fig. 4. The advantage of the overlapping zones is that they can make a location of a person easier to be distinguished because unique signal patterns are generated from 5 PIR sensors when a person moves in different locations in the monitored area. In addition, the FOVs are aligned symmetrically as much as possible to ensure that we have consistent patterns. All PIR sensors are connected to Arduino MEGA 2560 microcontroller to synchronize the transmission cycle of the five sensors and collect the most recent measurements received from these sensors.

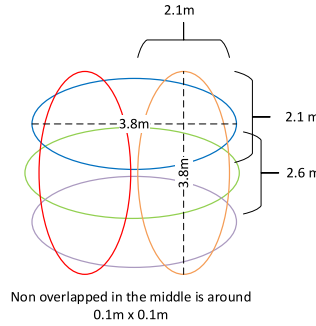
Next, we attach the sensor node to the ceiling in an indoor open office area where its ambient temperature is controlled regularly to ensure that PIR analog outputs are not affected by changes in temperature as much as possible. Note that the environmental condition that can affect the reading of a PIR sensor is mainly the room temperature because the difference



**Fig. 2.** A photo of a monitored area with a sensor node attached on ceiling at 2.4 meters height.



**Fig. 3.** An orientation of surrounding PIR sensors.



**Fig. 4.** An overlapped detecting area of a PIR node.

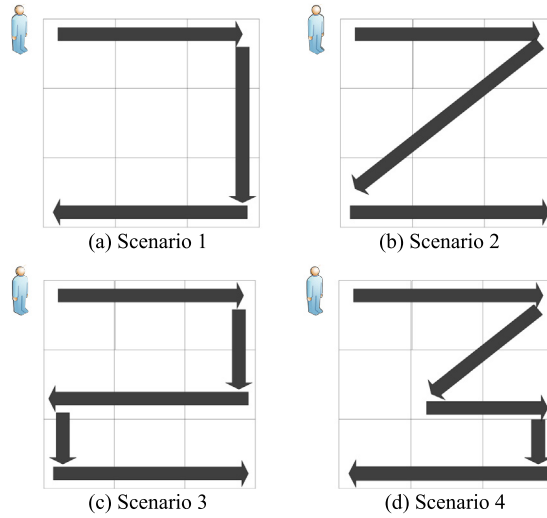
between a body temperature and the temperature can increase or decrease the amplitude of PIR analog output. Then, we set up a monitored area in a square shape as shown in Fig. 2 that consists of 9 grids and each grid has a size of  $1\text{ m} \times 1\text{ m}$ . We use an area size of  $3\text{ m} \times 3\text{ m}$  because a person's movement at the edge of the monitored area is too far from a sensor and is not sufficient enough voltage to trigger a sensor activation. The sensor node is linked to a desktop computer via a serial communication port for processing and storing the collected data. In our work, we set the sampling rate to 60 samples per second because we want to ensure that we do not miss any important features in PIR raw signals, and this sampling rate also simplifies time synchronization for data labeling.

### 3.2. Data collection

This section discusses the construction of a profile dataset. We further divide each grid in a monitored area into 4 square cells that have 50 cm in width and 50 cm in length. Each cell should be large enough to hold one person because the average shoulder of males and females is less than 50 cm. As a result, we have 36 cells in total. Each grid is labeled using numeric numbers from 1 to 9 and its cells are labeled using numbers from 1 to 4. Fig. 5 shows numeric labels for

1-1	1-2	2	3
1-3	1-4		
4	5	6	
7	8	9	

**Fig. 5.** Grids and cells of the monitored area with labeling numbers.



**Fig. 6.** An illustration of 4 test scenarios.

all grids and cells in the monitored area. Then, a participant is instructed to walk vertically, horizontally, and diagonally across a monitored area. To be specific, a participant starts walking from the left end to the right end of the monitored area. There are 6 starting locations for a horizontal walking direction. For the vertical walking direction, a person starts from the top end to the bottom end of the monitored area and there are 6 starting locations. For the diagonal walking directions, there are two directions including diagonal left and diagonal right. For the former direction, A participant walks from the top left end to the bottom right end of the monitored area. On the other hand, a participant walks from the top right end to the bottom left end of the monitored area for the latter direction. For each diagonal walking direction, there are 11 starting points. To complete one sample of the profile dataset, a person needs to walk forth and back in every direction. As a result, we have 24 sequences for the horizontal and vertical directions. In addition, we have 44 sequences for the diagonal directions. According to [11], different walking speeds can affect the amplitude and frequency of PIR analog output. However, it is not feasible to collect training data for all walking speeds. To reduce the effect of speed on the PIR output signal, we assume that a participant needs to walk at a constant speed. We employ a metronome that can generate a steady pulse to help a participant control his walking speed to be constant as much as possible. We employ a metronome that can generate a steady pulse to help a participant control his walking speed to be constant as much as possible.

To construct a test dataset, we develop 4 different walking scenarios in the real world as shown in Fig. 6. Similar to the profile data, we ask a participant to walk at a constant speed. For each scenario, we increase the complexity of walking paths to validate the collected PIR profile. In the first scenario, a participant walks from cells 1-1 to 3-2. Then, he turns and walks from cells 3-2 to 9-4. Finally, he walks from cell 9-4 and finishes at cell 7-3. This first scenario is the simplest one because the chosen paths generate signal patterns that are similar to the profile dataset.

In the second scenario, a participant walks from cells 1-1 to 3-2. Next, he walks diagonally from cells 3-2 to 7-3. Finally, he walks from cell 7-3 and finishes at cell 9-4. The second scenario is slightly more complicated than the first scenario due to the diagonal walking path. We observe that the signal patterns generated by the diagonal path are significantly varied so it can affect the accuracy of location estimation.



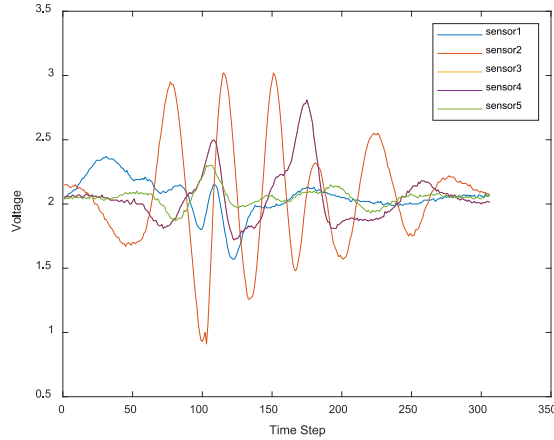


Fig. 7. A sample of PIR analog signal captured by a PIR node.

In the third scenario, a participant walks from cells 1-1 to 3-2. Next, he walks down from cells 3-2 to 6-4. Then, he continues walking from cells 6-4 to 4-3. Then, he takes another turn from cells 4-3 to 7-3. Finally, he walks from cell 7-3 and finishes at cell 9-4. In this scenario, we include a few shorter paths to introduce the partial signal patterns because we want to evaluate how well deep learning models can handle the partial pattern.

In the fourth scenario, a participant walks from cells 1-1 to 3-2. Next, he walks diagonally from cells 3-2 to 5-3. Then, he walks from cells 5-3 to 6-4. Then, he walks from cells 6-4 to 9-4. Finally, he walks from cell 9-4 and finishes at cell 7-3. In this fourth scenario, we not only include the partial pattern but also introduce some unexpected patterns generated by a person turning to change his direction under the sensor node. For both datasets, we record the ground truth using a video camera using 60 frames per second and the PIR sensors' signals are synchronized with each recorded frame using a local PC desktop time.

### 3.3. Dataset description and format

Our dataset has 36 classes in total and each class label represents a location of a cell in a monitored area. For example, cell 1-1 is presented as label 11 represents, cell 3-2 is presented as label 32. Training data contains 1020 sequences of raw PIR output, and each PIR sequence in training data has different lengths depending on the walking distance. Testing data comprises four scenarios with more than 2,000 timesteps. In addition, all PIR sequences have five channels for both training and testing data. The amplitude of raw PIR signals ranges from 0 to 5 V. When a PIR sensor detects no movement, the output voltage hovers around 2.05 V. To train machine learning models, we can either utilize raw data directly or extract features such as standard deviation, root-mean-square, peak-to-peak amplitude, etc.

For this dataset, we focus on a sequence-to-sequence (s2s) classification. To construct our profile data, we divide all collected PIR samples based on walking directions such as left-right, top-down, etc. For each direction, we crop and align the samples using a cross-correlation method because machine learning or deep learning model can learn the discriminative features from the samples correctly if the samples are aligned properly [52]. From one PIR sensor, a collected data is a sequence of output voltage  $VT = \{vt_1, vt_2, vt_i, \dots, vt_n\}$  where  $vt_i$  represents the  $i$ th voltage value in a sequence and  $n$  is a length of a sequence. We require data sequences from multiple PIR sensors to complete one sample for each walking direction  $VT_{sample} = \{VT_1, ; VT_2; \dots; VT_N\}$  where  $N$  is the total number of PIR sensors. In this paper, we use 5 PIR sensors to collect data and an example of analog signals is shown in Fig. 7. Therefore, one sample is a 2-dimensional matrix with a size of  $5 \times n$  where  $n$  is a length of a sequence. Next, a sample is associated with a ground truth  $L = \{l_1, l_2, l_i \dots l_n\}$  where  $l_i$  denotes the  $i$ th label in a sequence and  $n$  is a length of a sequence.

For test data, we use the same format as the profile data. Therefore, the test data is also a matrix with a size of  $5 \times n_{test}$  and is associated with a ground truth with a size of  $1 \times n_{test}$  where  $n_{test}$  is the length of the test sequence.

Our dataset is publicly available for download at <https://github.com/KanNgamakeur/PIR-localization-dataset>

## 4. Test methods

In this section, we apply localization methods to evaluate our system and dataset. Section A discusses various deep learning methods for location classification. Section B discusses CNN-based methods for cartesian coordinates estimation. Finally, Section C discusses the standard evaluation metrics used.

#### 4.1. Deep learning methods for classifying a location

For image classification, there are many state-of-the-art architectures such as Alexnet, Inception, ResNet, VGG, etc [53]. For the time series classification, we observe that there is no such gold standard architecture because the characteristics of time series data vary from application to application. Therefore, we review several related papers and find that the reviewed papers commonly adopt an architecture based on Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), or a combination of CNN and RNN [17,31,54–56]. Therefore, we evaluate our dataset using the above-mentioned architectures.

For CNN-based architectures, we test a 1D Convolutional Neural Network (1D-CNN) that has been modified from the standard convolutional neural networks for image classification (2D-CNN) to operate on 1D signals. In general, 1D CNNs consist of 1D convolution layers that extract discriminative features by sliding a set of 1D filters over the entire sequence of signals and fully connected (dense) layers that use the extracted features to perform a classification task. For the past years, several studies reported that 1D-CNNs give good results, are fast to train, and require low computational power [57]. We also test Temporal convolutional networks (TCNs) which are a variant of convolutional neural networks. They are designed to handle sequential data similar to recurrent neural networks (RNNs). TCNs use causal convolutions to convolve an output at time  $t$  with inputs up to time  $t$  in the previous layer [58]. However, they require many layers or larger filters to achieve the long effective history sizes. Therefore, dilated convolutions and residual connections are incorporated to address these problems without sacrificing the performance and stability of the networks. Several studies showed that TCNs' performance can match the performance of recurrent networks or even outperform them [59].

For RNN-based architectures, we test three recurrent networks including Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM), and Gated Recurrent Unit (GRU). LSTM can regulate the flow of information entering or exiting a memory cell using the input gate, output gate, and forget gate. These gates enable LSTM to choose what information to be maintained and removed. Thus, it can handle long sequences better than a traditional RNN [17]. Bi-LSTM can use both past and future information to achieve a better prediction for the sequence classification because it consists of two LSTMs that are trained using an original input sequence and a reversed copy of the input sequence [31]. GRU is a lightweight version of LSTM because a cell state is removed and there are only two gates including a reset gate and an update gate. As a result, GRU is faster to train compared with LSTM [60]. We observe that three RNNs can be used as a single layer or can be stacked on top of another. Therefore, we include single-layer RNNs and two stacked RNNs in our tested architectures.

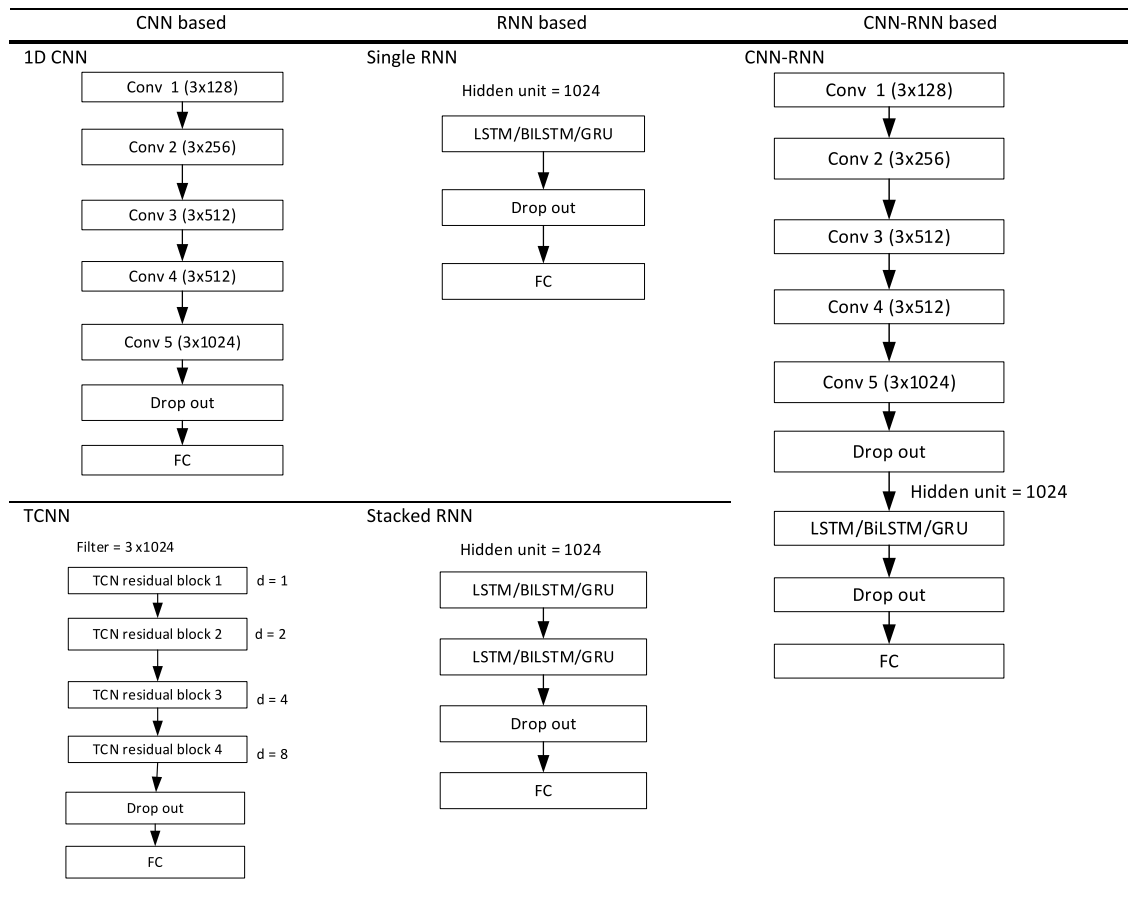
CNN-RNN-based architectures are composed of 1D Convolutional Neural Network (CNN) layers and Recurrent Neural Networks (RNNs). 1D-CNN layers perform feature extraction on input data. Then, extracted features are fed to an RNN layer that is capable of learning temporal dependency between extracted features. The hybrid models have been employed successfully in various applications such as activity recognition [61], Natural language processing [62], speech recognition [63], etc. We test CNN-LSTM, CNN-BiLSTM, and CNN-GRU respectively.

The configurations of our tested architectures are listed in Fig. 8. 1D CNN consists of 5 CNN layers with filter sizes of 128, 256, 512, 512 and 1024. Each CNN layer performs a 1D convolution operation on PIR signals using a filter size of 3 and a stride length of 1, followed by batch normalization and dropout. Next, the normalized output is passed to the Relu activation function. Finally, a 1D max pooling with a window size of 2 and a stride length of 1 is performed. For both convolution and max pooling, a zero-padding is applied to maintain the length of the input sequence. For TCNN, we set the number of filters to 1024 for all TCNN blocks. Each block uses filter size of 3 and a stride length of 1. The dilation factors are set to 1, 2, 4 and 8 respectively. For RNN based, we set the number of hidden units to 1024 for both single and stacked RNN. For CNN-RNN, we use a similar CNN architecture as 1D-CNN architecture and use a single RNN with 1024 hidden units. For all tested architectures, we set a dropout factor to 0.3 and a fully connected layer uses 36 neuron units to generate weighted sum outputs for 36 classes. Lastly, the Softmax function is used to convert the outputs into probabilities that sum to one.

To train all tested models, we use 10-fold cross-validation to ensure that the trained models are not under-fitted or overfitted. We employed an ADAM optimizer that is more effective than other optimizers such as RMSprop or SGD [64]. Cross-entropy is chosen as a loss function for this multiclass classification. We train each tested architecture for 100 epochs and set a mini-batch size to 32. A learning rate is set to 0.0001. To reduce an overfitting issue, L2 Regularization is set to 0.0001. All models are implemented using the Keras library and trained on an i5 3.2 GHz Intel quad-core CPU, Ram 16 Gb, and a GPU of NVIDIA RTX2070 with 8Gb video memory.

Once all models are trained, test data is inputted to these trained deep learning models to classify the locations of a person. However, our test data contains long sequences of analog signals that are generated by a PIR sensor node when a person walks or stops in different walking scenarios. To localize a person using PIR sensors, we are only interested in parts of the signals that are generated by the movements of a person. As a result, we need to perform movement detection. A threshold method is commonly used to indicate a movement or presence of a person because it is a simple yet effective method [29,65,66]. This detection method relies on a threshold value to determine whether signal events are caused by a movement of a person or not. We set a threshold value at  $2.1 \pm 0.2$  V to select only subsequences that pass a threshold value from test data. As a result, these subsequences contain only PIR analog outputs that are generated by a person's movements. Thus, we can feed the proper subsequences into the trained model to classify a person's location.





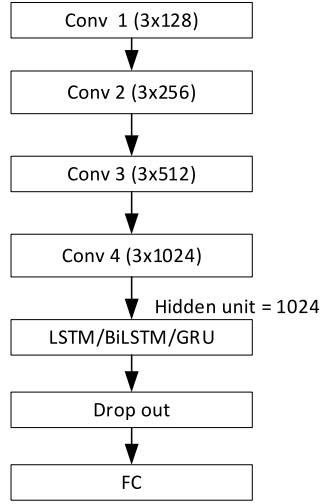
**Fig. 8.** The configuration of deep learning architectures for classification.

#### 4.2. Location estimation in cartesian coordinates

We not only want to test our dataset to estimate locations as a classification problem, but we also want to demonstrate an alternative way to utilize our dataset by estimating locations as cartesian coordinates. To achieve this idea, we first need to convert location labels into 2D coordinates for both training data and test data. We recorded ground truth using a video camera. For each frame, we draw a bounding box around a moving person. Then, the coordinates of the box are used to determine a cell that is occupied by a person. Since we synchronize our PIR data and video data using a local PC desktop time, we assign a cell number as a location label for each time step of PIR data appropriately. We then use the center coordinate of each cell as ground truths in our data. Then, we use deep learning architectures and particle filtering to estimate 2D coordinates.

For the deep learning architecture, we employ CNN-RNN-based architecture to estimate cartesian coordinates including CNN-LSTM, CNN-BiLSTM, and CNN-GRU in the demonstration because we have tested a variety of CNN-based, RNN based, and a hybrid CNN-RNN based architecture and found that a hybrid one achieves the best results. The configuration of deep learning architecture for cartesian coordinate estimation shown in Fig. 9, has 4 CNN layers and a single RNN layer. We set the number of filters in the CNN layer to 128, 256, 512 and 1024. Similarly, we use filter size of 3 and move the filter across the signal using a stride length of 1. We use the same structure of the CNN layer as in the 1D CNN model for classification. i.e., each CNN layer is followed by batch normalization, dropout, Relu and max pool. In addition, a dropout layer is added before a fully connected layer to reduce overfitting. The value of a dropout factor is set to 0.3. Instead of outputting only one label like in classification, a fully connected layer produces two outputs which are  $x$  and  $y$  coordinates respectively. Therefore, we set the number of neuron units to 2. We train the deep learning models with the ADAM optimizer and use the same hyperparameter setting as in the previous subsection. For this regression task, we select Mean Squared Error (MSE) loss function to be minimized by the ADAM optimizer.

To test the trained models, we follow the same method as described in the previous section to select subsequences that are generated by the movements of a person and feed them to the trained models to estimate the locations of a person in 2d coordinates.



**Fig. 9.** The configuration of deep learning architecture for cartesian coordinate estimation.

For Particle filtering, we select this algorithm because it can represent an arbitrary probability distribution and can handle nonlinear processes and measurement models [67]. To use particle filtering with our dataset, we adopt an azimuth change measurement proposed in [13]. According to their paper, we implement a system model that expresses a state transition of a moving person and an observation model that describe a relationship between a person's movement and observed azimuth measurement. For each loop of particle filtering, a non-overlapped window of length 60 is slid across test data and a threshold method is used to check if there is a person's movement or not. If yes, an azimuth change measurement is determined from PIR data in the sliding window. Then, we use it to correct the estimated state i.e., a location of a person.

#### 4.3. Evaluation metrics

To validate the performance of classification models, five performance metrics are implemented. The first commonly-used metric is the overall accuracy (Acc) which is the proportion of correct location estimations in the whole dataset:

$$Acc = \frac{TP}{Total\ number\ of\ locations}$$

where  $TP$  (true positive) is the total number of the correctly classified locations. However, we cannot rely on overall accuracy to determine the performance of the classification models because the calculation of the overall accuracy may be influenced by an imbalanced dataset. To give a fair comparison between the tested models, we employ the macro averaged recall, the macro averaged precision, the macro F1 score, and Cohan's Kappa Score (Kappa) because these additional metrics can provide the performance analysis of all tested models from different perspectives. The macro averaged recall can measure the overall ability of the tested models to identify all the true positives for each location, while the macro averaged precision can determine the overall trustworthiness of the models when they predict a particular location as a true positive. Both macro recall and macro precision are calculated by averaging the recall and the precision of each location as follows:

$$macro\_Re = \frac{\sum_{i=1}^n Re_i}{n}$$

$$macro\_Pr = \frac{\sum_{i=1}^n Pr_i}{n}$$

where  $Re_i$  and  $Pr_i$  are the recall and the precision of the  $i$ th location, and can be calculated as  $Re_i = TP_i / (TP_i + FN_i)$  and  $Pr_i = TP_i / (TP_i + FP_i)$  respectively. We denote  $TP_i$ ,  $FN_i$ , and  $FP_i$  as a true positive, false negative, and false positive of  $i$ th location accordingly. We also use the macro F1 score which is the harmonic mean of Macro-Precision and Macro-Recall. The macro F1 can provide a performance indicator that takes precision and recall into account and can be calculated as follows:

$$Macro\_F1\ score = 2 \times \frac{macro\_Pr \times macro\_Re}{macro\_Pr + macro\_Re}$$

where  $macro\_Pr$  and  $macro\_Re$  are the macro precision and the macro recall respectively. In general, the calculated F1 score is in the range of 0 to 1. A higher macro F1 score indicates that a model has low false positives and low false negatives for

**Table 1**  
Overall performance for all the tested DL architectures.

	Accuracy	Macro f1 score	Macro recall	Macro precision	Kappa
1D CNN	0.710	0.522	0.538	0.540	0.694
TCNN	0.780	0.631	0.660	0.650	0.763
LSTM	0.743	0.605	0.626	0.613	0.729
BiLSTM	0.844	0.700	0.722	0.690	0.835
GRU	0.754	0.549	0.567	0.583	0.740
Stacked LSTM	0.74	0.578	0.595	0.592	0.724
Stacked BiLSTM	0.847	0.727	0.751	0.726	0.840
Stacked GRU	0.722	0.578	0.606	0.579	0.707
CNN-LSTM	0.767	0.637	0.656	0.661	0.754
CNN-BiLSTM	0.849	0.770	0.812	0.760	0.841
CNN-GRU	0.734	0.606	0.628	0.630	0.719

all classes. Otherwise, a model performs poorly when the F1 score is low. The Kappa score aims to measure the agreement between a model's prediction and ground truth. Compared with the F1 score, it can provide a better understanding of the performance of the tested models for both multiclass and imbalance class problems. The Kappa score can be calculated as follows:

$$Kappa = \frac{P_o - P_e}{1 - P_e}$$

where  $P_o$  is the observed agreement and  $P_e$  is the expected agreement. Kappa values are in the range of  $-1$  and  $1$ . A model's prediction completely agrees with a ground truth when a kappa score is  $1$ . On the other hand, the value is less than or equal to  $0$  when they do not agree. For 2d coordinate estimation, we employ two popular metrics including mean distance error (MDE) and cumulative distance function (CDF) of distance errors. The mean distance error can be calculated by averaging the Euclidean distance between the estimated coordinates and the ground truth coordinates as follows:

$$MDE = \frac{\sum_{i=1}^N \sqrt{(x_e - x_g)^2 + (y_e - y_g)^2}}{N}$$

where  $(x_e, y_e)$  is the estimated coordinate,  $(x_g, y_g)$  is the ground truth coordinate, and  $N$  is the total number of locations. The cumulative probability function of a distance error is normally used to measure the precision of location estimation. It describes the probability that the distance error is within a particular value and can be expressed as:

$$F(x) = \int_{-\infty}^x f(t) dt$$

where  $f$  is the probability density function.

## 5. Results and discussions

This section discusses our experimental results on different aspects. Section A discusses cell-level classification results. Section B discusses estimation results based on 2D coordinates. Lastly, Section B provides discussions on the remaining challenges.

### 5.1. Cell-level classification results

The results of all tested models are shown in [Tables 1 and 2](#). The average accuracy of all models is 77%. For CNN based model, the performance of the 1D-CNN model is the lowest in all metrics compared with the other models. Although the accuracy and kappa score of 1D-CNN is around 71% and 0.694. However, the model suffers from a low f1 score, precision, and recall which are 0.522, 0.540, and 0.538 respectively. We observe that false-positive locations that are not part of ground truth locations are the cause of low f1 score, precision, and recall, especially for Scenarios 2 and 4. The TCNN model can perform equal to or even outperform some RNN models such as LSTM and GRU. It achieves an accuracy of 78% and a kappa score of 0.768. In addition, the f1 score, precision, and recall are better than the 1D-CNN model because the dilated causal convolution allows features to be extracted from the past time step up to the current time step without any leakage of future data and can cover a long data sequence. Thus, it performs better than 1D-CNN to model sequential data. TCNN achieves fairly good results in Scenarios 1 to 3 but performs poorly in Scenario 4 because there are a considerable number of false negatives for cell 53 where a person changes his direction under a sensor node.

For RNN based model, LSTM and GRU achieve accuracy of 74% and 75% respectively but obtain a low recall, precision, and f1 score which is around or less than 0.6 as shown in [Table 1](#). LSTM and GRU achieve satisfactory outcomes for Scenarios 1 to 3 but obtains poor results in Scenario 4 for the similar reason as TCNN. BiLSTM achieves an accuracy of 84% and a kappa score of 0.835 which are the best results compared with that of LSTM and GRU. The f1 score, precision,

**Table 2**

Detailed performance results for all the tested DL architectures.

	Accuracy				Macro f1 score				Macro recall				Macro precision				Kappa			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1D CNN	0.83	0.70	0.79	0.56	0.68	0.51	0.70	0.39	0.69	0.53	0.71	0.43	0.69	0.54	0.71	0.42	0.82	0.68	0.78	0.53
TCNN	0.87	0.85	0.83	0.61	0.79	0.75	0.84	0.44	0.81	0.76	0.86	0.46	0.78	0.78	0.84	0.53	0.86	0.83	0.82	0.59
LSTM	0.86	0.81	0.78	0.58	0.77	0.74	0.79	0.46	0.79	0.74	0.81	0.53	0.80	0.76	0.80	0.49	0.85	0.80	0.77	0.56
BiLSTM	0.91	0.86	0.88	0.75	0.91	0.82	0.85	0.62	0.92	0.80	0.86	0.65	0.90	0.85	0.85	0.61	0.91	0.85	0.85	0.73
GRU	0.91	0.80	0.77	0.60	0.82	0.64	0.79	0.43	0.83	0.64	0.82	0.45	0.81	0.68	0.80	0.46	0.91	0.79	0.76	0.57
Stacked LSTM	0.91	0.80	0.78	0.55	0.78	0.73	0.75	0.41	0.78	0.73	0.78	0.45	0.79	0.79	0.75	0.46	0.90	0.78	0.77	0.52
Stacked BiLSTM	0.91	0.87	0.87	0.77	0.85	0.86	0.88	0.60	0.85	0.84	0.89	0.65	0.88	0.87	0.90	0.61	0.90	0.86	0.86	0.75
Stacked GRU	0.89	0.74	0.79	0.52	0.81	0.61	0.76	0.38	0.82	0.61	0.78	0.43	0.81	0.63	0.78	0.40	0.88	0.72	0.78	0.50
CNN-LSTM	0.90	0.74	0.86	0.61	0.81	0.62	0.83	0.48	0.82	0.63	0.85	0.50	0.81	0.63	0.83	0.54	0.90	0.72	0.85	0.58
CNN-BiLSTM	0.96	0.87	0.87	0.75	0.95	0.83	0.85	0.66	0.96	0.82	0.86	0.70	0.95	0.87	0.86	0.67	0.95	0.86	0.86	0.73
CNN-GRU	0.90	0.72	0.81	0.57	0.77	0.51	0.83	0.46	0.77	0.51	0.86	0.50	0.76	0.54	0.84	0.48	0.89	0.69	0.80	0.54

and recall are around 0.70. the use of past and future information allows a better understanding of the time dependency of PIR outputs. Thus, misclassified locations are reduced for Scenarios 1 to 4. When two RNN networks are stacked, we can see that it is not much different in all metrics for all stacked RNN models except stacked BiLSTM which has an improvement in f1 score, precision, and recall. For Scenarios 1 to 3, stacked BiLSTM can perform equally or slightly better than BiLSTM. For Scenario 4, Stacked BiLSTM can improve the number of true positives for cell 53 compared with BiLSTM but the f1 score, precision, and recall do not improve because the number of the misclassified locations that are not part of ground truth locations is relatively unchanged.

For the CNN-RNN-based model, CNN-BiLSTM obtains an accuracy of 85% and a kappa score of 0.841. In addition, precision, recall, and f1 score are over 0.75. CNN-BiLSTM can perform better than stacked BiLSTM in Scenario 1 while its performance is nearly identical to or slightly lower than stacked BiLSTM in Scenarios 2 and 3. In Scenario 4, CNN-BiLSTM achieves a better f1 score, precision, and recall than stacked BiLSTM because the CNN-BiLSTM model obtains perfect precision and recall for more locations and has fewer misclassified locations. The overall performance of CNN-LSTM and CNN-GRU is slightly higher than the stacked version of LSTM and GRU. For Scenarios 1,3 and 4, CNN-LSTM and CNN-GRU have a better performance than stacked LSTM and stacked GRU. However, CNN-LSTM and CNN-GRU perform worse than stacked LSTM and stacked GRU in Scenario 2. We observe that both CNN-LSTM and CNN-GRU entirely misclassify some locations such as cell 33 as other locations. The extracted features may be similar between these cells but LSTM and GRU can use only the past information to determine locations. Thus, there is not enough information to make a correct estimation.

Based on all result, architectures containing BiLSTM performs better than other RNNs. Scenario 4 is quite problematic for all architecture due to random patterns of PIR output.

## 5.2. 2D coordinate estimation results

The results of 2D Coordinate estimation are shown in Table 3 and Fig. 10. CNN-LSTM obtains the mean distance error of 0.2359 m and around 75% of the distance error is less than 0.4 m, which is the best result compared with the other methods. For CNN-BiLSTM, it achieves a 28% lower mean distance error and 70% of the distance error is less than 0.4 m. BiLSTM can use past and future information to improve the understanding of sequence data. This characteristic of BiLSTM greatly improves the accuracy of certain applications such as language translation and speech recognition. However, it seems that we cannot fully take advantage of this characteristic in this case. CNN-GRU performs considerably worse than both CNN-LSTM and CNN-BiLSTM. It achieves the mean distance error of 0.5198 meters which is 75% higher than CNN-LSTM and 49% higher than CNN-BiLSTM. CDF of CNN-GRU also indicates that around 70% of the distance error is within 0.6 m. The bad performance of CNN-GRU may be due lack of a cell state. Particle filtering obtains the worst results compared with the deep learning methods. The mean distance error of the particle filter is 0.5482 m, and its CDF is slightly worse than GRU. The azimuth change measurement is quite ambiguous in some locations in a monitored area which affects the accuracy of the location estimation. From our observation, the ambiguity of azimuth change measurement causes particles to converge at the wrong location and there is no particle at the other locations in a monitored area. Thus, an estimated location cannot be corrected in the correction step. To mitigate this issue, we reinitialize 10% of particles and place them uniformly across the monitored area before we correct an estimated location. According to the above results, features extracted by deep learning methods can provide a better discriminative power than the handcrafted feature used in the particle filter.

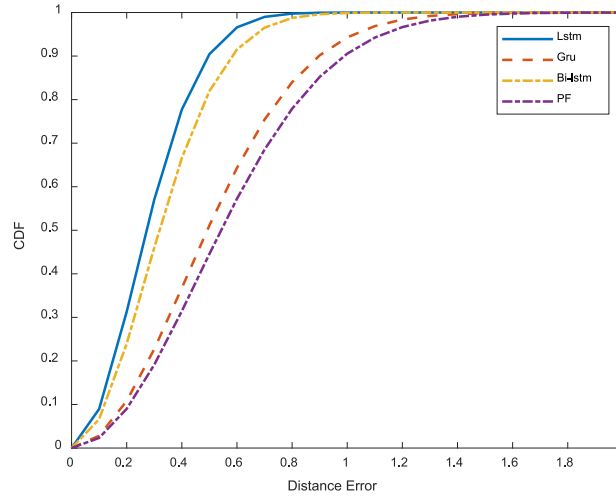
## 5.3. Discussions and challenges

Although the feasibility of our proposed system is proven and the performance of deep learning models is quite promising, several challenges need to be explored.

**Table 3**

Mean distance error and standard deviation for all the tested methods.

	Overall		Scenario 1		Scenario 2		Scenario 3		Scenario 4	
	Mean	Standard deviation	Meam	Standard deviation	Meam	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
CNN-LSTM	0.2359 m	0.2255 m	0.1388 m	0.0970 m	0.2005 m	0.1546 m	0.1860 m	0.1383 m	0.3708 m	0.3129 m
CNN-BiLSTM	0.3131 m	0.2812 m	0.2497 m	0.1411 m	0.2466 m	0.1439 m	0.2548 m	0.1660 m	0.4524 m	0.2740 m
CNN-GRU	0.5198 m	0.2189 m	0.4548 m	0.2638 m	0.5576 m	0.3023 m	0.4628 m	0.2713 m	0.5932 m	0.2661 m
Particle filter	0.5482 m	0.3265 m	0.4230 m	0.2083 m	0.5429 m	0.3268 m	0.5652 m	0.3984 m	0.6617 m	0.3049 m

**Fig. 10.** CDF of the distance error for all tested method.

**Signal ambiguity:** Although the deep learning models are capable of extracting features automatically, they do misclassify some locations due to the ambiguity of the extracted features or patterns. Patterns of the PIR signal in some locations in the monitored area are quite similar. As a result, a trained model cannot differentiate these patterns completely. As a result, a trained model often misclassifies these locations and this issue affects the accuracy of localization. Therefore, it may be necessary to include some extra handcrafted features to improve the differentiation of extracted features and patterns in different locations.

**Infeasible location:** Location estimations in some time steps can result in infeasible locations. Therefore, it would be preferable to include the motion information in the classification models to enhance their accuracy and reduce infeasible estimations.

**Unexpected signal patterns:** It is difficult for a trained model to predict a location of a person from unexpected patterns of the PIR signal because the model has not been trained to handle these patterns. Thus, the model usually produces an inaccurate localization result in this case. It is also not feasible to collect every combination of PIR output signals because it is time consuming and requires huge human efforts. It is preferable to have models that can handle unexpected signal patterns caused by some events, such as a person turning or a person standing still, to some extent to reduce errors in the location estimation.

**Length of sequence:** We use a full sequence of our test data to evaluate the feasibility of our dataset for deep learning models and the results are quite positive. However, sensor data is sent partially from time to time. Partial sensor data make it difficult for a trained model to predict a location of a person accurately because partial PIR sensor data do not provide enough information for a trained model to differentiate a location of a person accurately.

**Possibility for other methods of location estimation:** so far, we only explore location estimation using sequence-to-sequence classification, cartesian coordinates, and particle filters. However, there are a few other possibilities to be explored to use our dataset for location estimation. One can covert PIR signals to images using methods such as Gramian Angular Field (GAF) and Markov Transition Field (MTF) [68]. Then, it is possible to apply state-of-the-art deep learning models for image classification such as ResNet, VGGnet, etc.

**Scalability of the proposed system:** we focus on only a small-scale deployment of sensor nodes to evaluate the feasibility of our proposed system in this work. To improve the scalability of our proposed system, we need to increase the number of installed sensor nodes. It is a common practice to have a sink node to collect data from different sensor nodes. However, there are some factors to be considered such as a reporting frequency, a communication protocol, time synchronization of sensor nodes, etc. These factors can affect our systems in terms of energy consumption, localization accuracy, and real-time capability. A trade-off between these factors must be further explored.

## 6. Conclusion & future work

In this paper, we propose a Passive Infrared sensor system and a dataset for indoor localization and tracking. Our sensor node is developed based on commercial-off-the-shelf sensors. The dataset consists of 1,000 walking samples, 4 scenarios, and 36 classes in total. We evaluate our dataset using different deep learning methods such as CNN, RNN, and CNN-RNN. The results confirm the feasibility of our dataset and show that the best deep learning method can achieve an accuracy of 85%, a kappa score of 0.84, and an f1 score of 0.77. Furthermore, our dataset can be converted for a problem of coordinate estimation. To validate this idea, we apply both deep learning methods and the particle filter to estimate coordinates. The best performer achieves a distance error of 0.25 m.

In the future, we aim to improve the scalability of the proposed system. In addition, we will increase the variety of our dataset. more scenarios will be included, especially with multiple persons. Different topologies of sensor deployment will be included.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## References

- [1] L. Zhao, H. Huang, C. Su, S. Ding, H. Huang, Z. Tan, Z. Li, Block-sparse coding-based machine learning approach for dependable device-free localization in IoT environment, *IEEE Internet Things J.* 8 (5) (2021) 3211–3223.
- [2] G. Acampora, D.J. Cook, P. Rashidi, A.V. Vasilakos, A survey on ambient intelligence in health care, *Proc. IEEE. Inst. Electr. Electron. Eng.* 101 (12) (2013) 2470–2494.
- [3] A.L. Ballardini, I. Ferretti, S. Fontana, A. Furlan, D.G. Sorrenti, An indoor localization system for telehomecare applications, *IEEE Trans. Syst. Man Cybern. Syst.* 46 (10) (2016) 1445–1455.
- [4] S. Qiu, Z. Wang, H. Zhao, K. Qin, Z. Li, H. Hu, Inertial/magnetic sensors based pedestrian dead reckoning by means of multi-sensor fusion, *Inf. Fusion* 39 (2018) 108–119, 2018.
- [5] K. Ngamakeur, S. Yongchareon, J. Yu, S.U. Rehman, A survey on device-free indoor localization and tracking in the multi-resident environment, *ACM Comput. Surv.* 53 (4) (2020) 71.
- [6] S. Xu, R. Chen, Y. Yu, G. Guo, L. Huang, Locating smartphones indoors using built-in sensors and Wi-Fi ranging with an enhanced particle filter, *IEEE Access* 7 (2019) 95140–95153.
- [7] B. Yang, M. Zhang, Credit-based multiple human location for passive binary pyroelectric infrared sensor tracking system: Free from region partition and classifier, *IEEE Sens. J.* 17 (1) (2017) 37–45.
- [8] H. Qi, H. Fei, X. Yang, Multiple human tracking and identification with wireless distributed pyroelectric sensor systems, *IEEE Syst. J.* 3 (4) (2009) 428–439.
- [9] S. Tao, M. Kudo, H. Nonaka, J. Toyama, Person localization and soft authentication using an infrared ceiling sensor network, *Comput. Anal. Images Patterns* 122–129.
- [10] F. Alam, N. Faulkner, B. Parr, Device-free localization: A review of non-RF techniques for unobtrusive indoor positioning, *IEEE Internet Things J.* 8 (6) (2021) 4228–4249.
- [11] S. Narayana, R.V. Prasad, V.S. Rao, T.V. Prabhakar, S.S. Kowshik, M.S. Iyer, PIR sensors: Characterization and novel localization technique, in: *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, New York, NY, USA, 2015, pp. 142–153.
- [12] B. Mukhopadhyay, S. Sarangi, S. Srirangarajan, S. Kar, Indoor localization using analog output of pyroelectric infrared sensors, in: *2018 IEEE Wireless Communications and Networking Conference, WCNC*, 2018, pp. 1–6.
- [13] X. Liu, T. Yang, S. Tang, P. Guo, J. Niu, From relative azimuth to absolute location: pushing the limit of PIR sensor based localization, in: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, Vol. 1, Association for Computing Machinery, 2020.
- [14] L. Wu, Y. Wang, H. Liu, Occupancy detection and localization by monitoring nonlinear energy flow of a shuttered passive infrared sensor, *IEEE Sens. J.* 18 (21) (2018) 8656–8666.
- [15] X. Wang, L. Gao, S. Mao, S. Pandey, DeepFi: Deep learning for indoor fingerprinting using channel state information, in: *2015 IEEE Wireless Communications and Networking Conference, WCNC*, 2015, pp. 1666–1671.
- [16] J. Wang, X. Zhang, Q. Gao, H. Yue, H. Wang, Device-free wireless localization and activity recognition: A deep learning approach, *IEEE Trans. Veh. Technol.* 66 (7) (2017) 6258–6267.
- [17] Z. Chen, H. Zou, J. Yang, H. Jiang, L. Xie, WiFi fingerprinting indoor localization using local feature-based deep LSTM, *IEEE Syst. J.* 14 (2) (2020) 3001–3010.
- [18] S. Palipana, B. Pietropaoli, D. Pesch, Recent advances in RF-based passive device-free localisation for indoor applications, *Ad Hoc Netw.* 64 (2017) 80–98.
- [19] J. Wang, D. Fang, Z. Yang, H. Jiang, X. Chen, T. Xing, L. Cai, E-HIPA: An energy-efficient framework for high-precision multi-target-adaptive device-free localization, *IEEE Trans. Mob. Comput.* 16 (3) (2017) 716–729.
- [20] B. Wagner, N. Patwari, D. Timmermann, Passive RFID tomographic imaging for device-free user localization, in: *9th Workshop on Positioning, Navigation and Communication*, Dresden, 2012, pp. 120–125.
- [21] J. Wilson, N. Patwari, Radio tomographic imaging with wireless networks, *IEEE Trans. Mob. Comput.* 9 (5) (2010) 621–632.
- [22] J. Wilson, N. Patwari, See-through walls: Motion tracking using variance-based radio tomography networks, *IEEE Trans. Mob. Comput.* 10 (5) (2011) 612–621.
- [23] D. Zhang, J. Zhou, M. Guo, J. Cao, T. Li, TASA: Tag-free activity sensing using RFID tag arrays, *IEEE Trans. Parallel Distrib. Syst.* 22 (4) (2011) 558–570.



- [24] D. Lieckfeldt, J. You, D. Timmermann, Characterizing the influence of human presence on bistatic passive RFID-system, in: 2009 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Marrakech, Morocco, 2009, pp. 338–343.
- [25] D. Lieckfeldt, J. You, D. Timmermann, Passive tracking of transceiver-free users with RFID, *Intell. Interact. Assist. Mob. Multimed. Comput.* (2009) 319–329.
- [26] D. Zhang, J. Ma, Q. Chen, L.M. Ni, An RF based system for tracking transceiver-free objects, in: Fifth Annual IEEE International Conference on Pervasive Computing and Communications, PerCom'07, White Plains, NY, USA, 2007, pp. 135–144.
- [27] V. Rampa, M. Nicoli, C. Manno, S. Savazzi, EM model-based device-free localization of multiple bodies, *Sensors* 21 (5) (2021).
- [28] M. Youssef, M. Mah, A. Agrawala, Challenges device-free passive localization for wireless environment, in: The 13th Annual ACM International Conference on Mobile Computing and Networking, Montréal, Québec, Canada, 2007, pp. 222–229.
- [29] J. Xiao, K. Wu, Y. Yi, L. Wang, L.M. Ni, Pilot: Passive device-free indoor localization using channel state information, in: 2013 IEEE 33rd International Conference on Distributed Computing Systems, Philadelphia, PA, USA, 2013, pp. 236–245.
- [30] X. Wang, X. Wang, S. Mao, Deep convolutional neural networks for indoor localization with CSI images, *IEEE Trans. Netw. Sci. Eng.* 7 (1) (2020) 316–327.
- [31] S. Fan, Y. Wu, C. Han, X. Wang, A structured bidirectional LSTM deep learning method for 3D Terahertz indoor localization, in: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, 2020, pp. 2381–2390.
- [32] M. Youssef, A. Agrawala, The horus WLAN location determination system, in: Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, Seattle, Washington, 2005, pp. 205–218.
- [33] I. Sabek, M. Youssef, A.V. Vasilakos, ACE: An accurate and efficient multi-entity device-free WLAN localization system, *IEEE Trans. Mob. Comput.* 14 (2) (2015) 261–273.
- [34] C. Xu, B. Firner, W. Trappe, R.S. Moore, R. Howard, F. Zhang, Y. Zhang, NingAn, SCPL: Indoor device-free multi-subject counting and localization using radio signal strength, in: 2013 ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN, Philadelphia, 2013, pp. 79–90.
- [35] B. Gulmezoglu, M.B. Guldogan, S. Gezici, Multiperson tracking with a network of ultrawideband radar sensors based on Gaussian mixture PHD filters, *IEEE Sens. J.* 15 (4) (2015) 2227–2237.
- [36] F. Lemic, J. Martin, C. Yarp, D. Chan, V. Handziski, R. Brodersen, G. Fettweis, A. Wolisz, J. Wawrzyn, Localization as a feature of mmWave communication, pp. 1033–1038.
- [37] T. Wei, X. Zhang, Mtrack: High-precision passive tracking using millimeter wave radios, in: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, Paris, France, 2015, pp. 117–129.
- [38] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrani, M.A. Al-Ammar, H.S. Al-Khalifa, Ultra wideband indoor positioning technologies: Analysis and recent advances, *Sensors (Basel)* 16 (5) (2016).
- [39] A. Braun, H. Heggen, R. Wichert, CapFloor – a flexible capacitive indoor localization system, evaluating AAL systems through competitive benchmarking, in: S. Chessa, S. Knauth (Eds.), Indoor Localization and Tracking: International Competition, Competition in Valencia, Spain, July (2011) 25–29, and Final Workshop in Lecce, Italy, September 26, 2011, in: EvAAL 2011, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 26–35, Revised Selected Papers.
- [40] X. Bian, G.D. Abowd, J.M. Rehg, Using sound source localization in a home environment, in: In Proceedings of the Third International Conference on Pervasive Computing, Berlin, Heidelberg, 2005, pp. 19–36.
- [41] C. Cai, H. Pu, P. Wang, Z. Chen, J. Luo, We hear your PACE: Passive acoustic localization of multiple walking persons, *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5 (2) (2021) 55.
- [42] H. Yang, W.D. Zhong, C. Chen, A. Alphons, P. Du, Qos-driven optimized design-based integrated visible light communication and positioning for indoor IoT networks, *IEEE Internet Things J.* 7 (1) (2020) 269–283.
- [43] Q. Hao, D.J. Brady, B.D. Guenther, J.B. Burchett, M. Shankar, S. Feller, Human tracking with wireless distributed pyroelectric sensors, *IEEE Sens. J.* 6 (6) (2006) 1683–1696.
- [44] B. Yang, J. Luo, Q. Liu, A novel low-cost and small-size human tracking system with pyroelectric infrared sensor mesh network, *Infrared Phys. Technol.* 63 (2014) 147–156.
- [45] J. Kemper, D. Hauschildt, Passive infrared localization with a probability hypothesis density filter, in: 2010 7th Workshop on Positioning, Navigation and Communication, Dresden, Germany, 2010, pp. 68–76.
- [46] C. Wei-Han, M. Hsi-Pin, A fall detection system based on infrared array sensors with tracking capability for the elderly at home, in: 2015 17th International Conference on E-Health Networking, Application & Services, HealthCom, Boston, MA, USA, 2015, pp. 428–434.
- [47] S. Narayana, V. Rao, R.V. Prasad, A.K. Kanthila, K. Managundi, L. Mottola, T.V. Prabhakar, LOCI: Privacy-aware, Device-free, Low-power Localization of Multiple Persons using IR Sensors, pp. 121–132.
- [48] D. Yang, W. Sheng, R. Zeng, Indoor human localization using PIR sensors and accessibility map, in: The 5th Annual IEEE International Conference on Cyber Technology in Automation, in: Control and Intelligent Systems, Shenyang, China, 2015, pp. 577–581.
- [49] S. Tao, M. Kudo, B.-N. Pei, H. Nonaka, J. Toyama, Multiperson locating and their soft tracking in a binary infrared sensor network, *IEEE Trans. Hum.-Machine Syst.* 45 (5) (2015) 550–561.
- [50] J.G. Rohra, B. Perumal, S.J. Narayanan, P. Thakur, R.B. Bhatt, User Localization in an Indoor Environment Using Fuzzy Hybrid of Particle Swarm Optimization & Gravitational Search Algorithm with Neural Networks. in: Proceedings of Sixth International Conference on Soft Computing for Problem Solving, pp. 286–295.
- [51] M.T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, K. Reddy, Recurrent neural networks for accurate RSSI indoor localization, *IEEE Internet Things J.* 6 (6) (2019) 10639–10651.
- [52] G. Trigeorgis, M.A. Nicolaou, B.W. Schuller, S. Zafeiriou, Deep canonical time warping for simultaneous alignment and representation learning of sequences, *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (5) (2018) 1128–1138.
- [53] Z. Luo, F. Branchaud-Charron, C. Lemaire, J. Konrad, S. Li, A. Mishra, A. Achkar, J. Eichel, P. Jodoin, MIO-tcd: A new benchmark dataset for vehicle classification and localization, *IEEE Trans. Image Process.* 27 (10) (2018) 5129–5141.
- [54] E. Maiorana, Deep learning for EEG-based biometric recognition, *Neurocomputing* 410 (2020) 374–386, 2020.
- [55] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.-A. Muller, Deep learning for time series classification: a review, *Data Min. Knowl. Discov.* 33 (4) (2019) 917–963, 2019.
- [56] X. Huang, G.C. Fox, S. Serebryakov, A. Mohan, P. Morkisz, D. Dutta, Benchmarking deep learning for time series: Challenges and directions, in: 2019 IEEE International Conference on Big Data, Big Data, Los Angeles, CA, USA, 2019, pp. 5679–5682.
- [57] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D.J. Inman, 1D convolutional neural networks and applications: A survey, *Mech. Syst. Signal Process.* 151 (2021) 107398, 2021.
- [58] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018, CoRR, abs/1803.01271.
- [59] Y. Wang, F. Liu, L. Yang, EEG-based depression recognition using intrinsic time-scale decomposition and temporal convolution network, in: The Fifth International Conference on Biological Information and Biomedical Engineering, Hangzhou, China, 2021, p. 5.

- [60] J. Chung, Ç. Gülçehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, ArXiv, abs/1412.3555.
- [61] R. Mutegeki, D.S. Han, A CNN-LSTM approach to human activity recognition, in: 2020 International Conference on Artificial Intelligence in Information and Communication, ICAIIC, Fukuoka, Japan, 2020, pp. 362–366.
- [62] S. Zhang, Language processing model construction and simulation based on hybrid CNN and LSTM, *Comput. Intell. Neurosci.* 2021 (2021) 2578422, 2021.
- [63] J. Zhao, X. Mao, L. Chen, Speech emotion recognition using deep 1D & 2D CNN LSTM networks, *Biomed. Signal Process. Control* 47 (2019) 312–323, 2019.
- [64] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2015, CoRR, abs/1412.6980.
- [65] L. Gong, W. Yang, Z. Zhou, D. Man, H. Cai, X. Zhou, Z. Yang, An adaptive wireless passive human detection via fine-grained physical layer information, *Ad Hoc Netw.* 38 (2016) 38–50.
- [66] P. Zappi, E. Farella, L. Benini, Tracking motion direction and distance with pyroelectric IR sensors, *IEEE Sens. J.* 10 (9) (2010) 1486–1494.
- [67] M.F. Bugallo, T. Lu, P.M. Djuric, Target tracking by multiple particle filtering, in: 2007 IEEE Aerospace Conference, Big Sky, MT, USA, 2007, pp. 1–7.
- [68] Z. Wang, T. Oates, Imaging time-series to improve classification and imputation, in: Proceedings of the 24th International Conference on Artificial Intelligence, Buenos Aires, Argentina, 2015, pp. 3939–3945.