


Article

A Robust and Accurate Indoor Localization Using Learning-Based Fusion of Wi-Fi RTT and RSSI

Hamada Rizk ^{1,2}, Ahmed Elmogy ^{3,4,*}  and Hirozumi Yamaguchi ²¹ Computers & Control Engineering Department, Tanta University, Tanta 31527, Egypt; hamada_rizk@f-eng.tanta.edu.eg² Graduate School of Information Science and Technology, Osaka University, Suita 565-0871, Japan; hamada_rizk,h-yamagu@ist.osaka-u.ac.jp³ Faculty of Computer Engineering & Sciences, Prince Sattam Ibn Abdelaziz University, Al-Kharj 16273, Saudi Arabia⁴ Faculty of Engineering, Tanta University, Tanta 31527, Egypt

* Correspondence: a.elmogy@psau.edu.sa

Abstract: Great attention has been paid to indoor localization due to its wide range of associated applications and services. Fingerprinting and time-based localization techniques are among the most popular approaches in the field due to their promising performance. However, fingerprinting techniques usually suffer from signal fluctuations and interference, which yields unstable localization performance. On the other hand, the accuracy of time-based techniques is highly affected by multipath propagation errors and non-line-of-sight transmissions. To combat these challenges, this paper presents a hybrid deep-learning-based indoor localization system called *RRLoc* which fuses fingerprinting and time-based techniques with a view of combining their advantages. *RRLoc* leverages a novel approach for fusing received signal strength indication (RSSI) and round-trip time (RTT) measurements and extracting high-level features using deep canonical correlation analysis. The extracted features are then used in training a localization model for facilitating the location estimation process. Different modules are incorporated to improve the deep model's generalization against overtraining and noise. The experimental results obtained at two different indoor environments show that *RRLoc* improves localization accuracy by at least 267% and 496% compared to the state-of-the-art fingerprinting and ranging-based-multilateration techniques, respectively.

Keywords: indoor localization; deep learning; fingerprinting; round-trip time; canonical correlation analysis



Citation: Rizk, H.; Elmogy, A.; Yamaguchi, H. A Robust and Accurate Indoor Localization Using Learning-Based Fusion of Wi-Fi RTT and RSSI. *Sensors* **2022**, *22*, 2700. <https://doi.org/10.3390/s22072700>

Academic Editor: David Plets

Received: 24 February 2022

Accepted: 30 March 2022

Published: 31 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

User location becomes one of the most valuable contexts in human-centric environments. This context can be used to enhance a wide range of applications and services such as tracking, navigation, healthcare, emergency, etc. [1–3]. For instance, an improved localization accuracy, which reduces emergency response time by one minute, saves over 10,000 lives annually in the USA alone [4]. Since people spend most of their time indoors, immense attention has been paid to indoor localization. Despite the fact that GPS is the standard localization system, it cannot be leveraged in indoor settings. This is due to the high levels of signal interference and reflection [5] that block the line of sight to the reference satellites. Therefore, indoor localization has been an active research topic to find a ubiquitous and accurate alternatives to GPS in indoor settings [6–9].

Various technologies have been investigated, including Wi-Fi, radio frequency identification (RFID), Bluetooth, ultrawideband (UWB), cellular, zigbee, IMU, etc. [10–13]. Each technology has its own advantages which support its adoption in specific types of applications. Wi-Fi has been widely adopted due to its ubiquitous coverage and the support of the IEEE 802.11 standard by the majority of mobile devices [8,14–22].

Various localization techniques have been proposed to overcome the challenges related to indoor localization. These techniques include multilateration, fingerprinting, angle of arrival, and time-based techniques [23,24]. Fingerprinting and time-based techniques are the most researched ones in the field. Fingerprinting [25–27] is widely adopted due to its relatively good performance especially with deep learning. Fingerprinting technique builds a fingerprint database that involve signatures of Wi-Fi signals collected at different reference locations covering the area of interest. The fingerprint database is then used to define a model that can be used to estimate the user location, given the received signals at the run-time. Specifically, the defined model can be classified as deterministic [3] or probabilistic [28], or even as a machine-learning-based model [29]. Probabilistic solutions in general have better ability to mitigate the inherent random noise of wireless signals compared to deterministic solutions [24]. However, probabilistic solutions usually assume access points (APs)-independency to avoid a dimensionality problem [30], which is practically incorrect and also leads to information loss [24]. Therefore, deep learning has been widely adopted to learn the underlying joint distribution of signals received from the installed APs, leading to a superior localization performance. While many researchers have also examined fingerprinting techniques as a solution for localization issues, these techniques usually suffer from RSSI fluctuations and signal interference due to their sensitivity to obstacles, multipath fading, and indoor radio noise and/or hardware.

To overcome the above fingerprinting challenges, time-based techniques have been investigated. These techniques determine the distance of a mobile unit (e.g., phone) to APs using the measurement of the signal's propagation time and the known signal's velocity. Different approaches have been proposed for measuring the propagation time, including time of arrival (ToA) [31,32], time difference of arrival (TDoA) [33], and RTT [34]. The problem with ToA and TDoA is that they require a precise time synchronization of all devices. In contrast, RTT uses one clock to measure the time needed for the signal to travel to a destination node and return, and thus the synchronization problem is mitigated. Unlike RSSI-based approaches, RTT is more resilient to the challenges of cluttered indoor environments, including multipath interference, signal attenuation, transmission power variation, and radio interference. The fine time measurement (FTM) protocol, which can measure the RTT between the mobile phone and the APs, has recently been introduced by the IEEE 802.11-2016 standard. This protocol has increasing support from commercial APs and consumers' mobile phones, making time-based techniques a promising solution for enabling practical indoor localization.

Nevertheless, RTT does not eliminate the significant indoor localization errors due to multipath propagation and latency as well as non-line-of-sight transmissions. Thus, signals traverse longer indirect paths, resulting in longer travel distance estimation (distance overestimation) [17]. To mitigate this issue, some solutions were proposed based on map matching or filtering [34]. However, they only slightly improve the localization performance [17].

In this paper, we present *RRLoc*: a hybrid fingerprinting-based indoor localization system that combines the advantages of both RSSI-based and time-based techniques. Specifically, the system constructs a fingerprint map of both RSSI and RTT at different discrete reference points in the area of interest. Then, the fused fingerprints are utilized for training a deep-learning-based classifier to track the user's location.

This approach enables *RRLoc* to overcome the shortcomings associated with each modality. Nevertheless, fusing the RSSI and RTT to obtain a discriminative signature that facilitates the location estimation process is challenging, since they represent different modalities that are intrinsically dissimilar in nature. Thus, the error of the modality joint combination usually confuses/deceives the localization model, leading to more errors than those of each individual modality [14]. Therefore, *RRLoc* proposed a novel fusion and feature extraction method that automatically projects the two modalities into hyperspace where their correlation is maximum. To obtain that space, *RRLoc* employs a deep version of canonical correlation analysis (DCCA) [35,36] that yields a coordinated representation of the

input modalities. The DCCA is trained to maximize the correlation between different RSSI and RTT, leading to separable representations facilitating user location estimation. In this vein, the developed hybrid model can cope with the localization challenges such as NLOS (non-line of sight), interference, clock synchronization, and signal attenuation. Unlike data integration methods [14], which tend to leverage information from one modality to improve the other, the proposed fusion method incorporates both modalities in a combined analysis, thus allowing for true interaction between them that maximizes their benefit.

The proposed system leverages the power of deep learning [4,29] in learning complex mapping functions to boost the robustness of the localization system. Furthermore, we ensure the system's generalization by training models on different types of noisy signals during the location estimation stage. Additionally, *RRLoc* is able to avert the overfitting that may occur during the training phase by incorporating several regularization techniques.

We implemented and evaluated *RRLoc* using different Android phones on two different cluttered environments, a large environment of an area of 629 m² and a small one of 141 m². Seven commercial Google Wi-Fi APs were installed in each environment that already included other traditional non-RTT-enabled APs that could act as an added source of interference. Our results show that *RRLoc* achieved a submeter localization accuracy for both indoor environments with a median localization error of 0.42 m and 0.32 m for the two environments, respectively. These results reveal an improvement over the traditional RSSI fingerprinting accuracy by at least 267% and outperform the ranging-based multilateration localization approach accuracy by at least 496%. This accuracy was maintained under heterogeneous devices which qualified *RRLoc* as a robust and accurate indoor localization technique.

The contribution of this paper is threefold. First, we present a novel deep-learning-based indoor localization system exploiting the availability of the FTM protocol on consumer devices. Second, a novel data-driven-based fusion and feature learning method is designed for extracting a correlated representation of RTT and RSSI. Third, we train a robust localization model to enable pinpointing the user in the continuous space. We experimentally evaluate the performance of the proposed system, demonstrating its capability to localize with fine-grained accuracy.

The rest of this paper is structured as follows. Section 2 discusses the most relevant work of the proposed system. Section 3 provides a brief introduction to the IEEE802.11-2016 FTM protocol and the canonical correlation analysis approach. In Section 4, we provide a general overview of the *RRLoc* system architecture and present its different components. In Section 5, the modules of the *RRLoc* are introduced in details. Section 6 evaluates the different parameters of the system and shows its overall performance compared to the other approaches. Finally, Section 7 concludes the paper and discusses future work.

2. Related Work

Towards constructing smart buildings, many researchers have considered the problem of indoor localization. GPS has been known as an excellent technology used for localization, but it is not suitable for indoor positioning because indoor environments have great loss in signal propagation. Thus, many other sensors and technologies have been examined. Examples include Bluetooth, ultrasound, RFID, etc. However, the use of these sensors and technologies is restricted due to their limited energy, high cost, and/or constrained bandwidth. On the other hand, Wi-Fi technology has been recently given considerable attention with the great and continuous use of smart phones in almost all life activities. Thus, many Wi-Fi-based systems can be easily developed with reasonable cost as no special infrastructure is needed. The RSSI and time-based techniques are among the most applicable Wi-Fi techniques. Some discussions about these techniques and their relevance to our work are introduced in this section.

2.1. RSSI-Based Techniques

RSSI-based localization works by estimating the distance between two nodes and measuring their received signal strength. The RSSI positioning techniques are burdened by their poor accuracy that is due to many reasons such as NLOS, fading, noise data, etc. [37,38]. The simplicity of RSSI techniques has motivated many researchers to work on mitigating these issues. The RSSI-fingerprinting technique is one of the successful techniques toward this goal [25–27]. Fingerprinting localization approaches are common examples of probabilistic localization techniques. The RSSI-fingerprinting systems work on determining object location using two stages: offline and online phases. In the offline phase, the fingerprints of objects (objects' RSSI) are measured at predefined reference positions to build a fingerprint database. These fingerprints will be used in the online phase to estimate the objects' positions at new locations. Different features can be implemented as system fingerprints. Correspondingly, channel state information (CSI) techniques work on providing detailed information about the signal information between two communicated nodes. Both the RSSI and CSI localization approaches are highly affected by the changes in the power of Wi-Fi nodes which are very common. The heterogeneity of the used Wi-Fi devices may also degrade their performance. Although fingerprinting approaches are extensively used in developing good indoor localization systems overcoming the abovementioned challenges, the associated fingerprints/signatures are sensitive to signal interference, diffraction, and fading. In addition, to achieve efficient localization with fingerprinting, a homogeneous distribution of APs is highly recommended.

2.2. Time-Based Techniques

Alternatively, the time-based approaches are popularly used for indoor localization. They work to determine the objects' positions based on time measurements and the known velocity of the transmitted signal. The ToA [31,32], TDoA [33], and RTT [34] are among the most popular techniques used for this purpose. The ToA technique works by measuring the time the signal takes to reach the receiver station (timestamp). In order to obtain accurate time estimation, a strict synchronization between the two sides is necessary [31,32]. The TDoA techniques work by transmitting signals from three or more stations and then measuring the difference between the signals propagation times, which are then used to estimate the user location. Again, this requires a type of time synchronization but with transmitters only, unlike the ToA technique [33]. The ToA, and TDoA techniques are considered one-way measurement techniques.

On the other hand, RTT is a two-way measurement technique that works by measuring the round-trip time, which is the time the signal takes to travel from the transmitter to the receiver and back. The most important advantage of the RTT technique over the ToA and TDoA techniques is that it does not require synchronization between the transmitter and receiver as only one clock is used. However, traditional multilateration systems that incorporate the RTT measurements suffer from poor accuracy due to NLOS and multipath effects [34].

Numerous Wi-Fi-RTT approaches have been developed to mitigate the effect of NLOS and multipath effects. In [39] for example, a real-time ranging Wi-Fi-RTT model was developed to reduce the error caused by multipath, and NLOS effects. In addition, in [40], the authors proposed a calibration model that works by eliminating the transmitter RTT range offset and thus improving accuracy. In [41], a Wi-Fi FTM geomagnetic positioning approach was proposed to mitigate the effect of NLOS. An enhanced mind evolutionary algorithm (EMEA) was incorporated in the developed approach to ensure the localization accuracy, whereas in [42], a Wi-Fi-RTT-based approach was developed by line-of-sight identification and range calibration. Some other works depend on identifying the NLOS and multipath signals and categorizing them to low- and high-quality signals using support vector machines [43].

For the sake of obtaining better positioning accuracy, a RTT-RSSI technique was proposed in [9]. This hybrid technique uses a new and simple multilateration model that combines RTT and RSSI techniques to improve the localization accuracy. However, this

cannot be continually achieved since some signal attenuation may occur due to NLOS conditions. In addition, in [14], a hybrid RTT-RSSI fingerprinting localization approach was proposed. However, the deduced results show that the proposed approach is not able to achieve the expected accuracy as the correlation between the different modalities is not taken into consideration.

In this paper, a novel hybrid RTT-RSSI fingerprinting approach is presented. The introduced model is able to cope with the localization challenges mentioned above. The deep learning is incorporated in the designed model to increase its robustness and generalization. The details of the presented approach are given below.

3. Background

3.1. Round-Trip Time (RTT)

The round-trip time is a time-based technique mainly used for calculating distance. It is used in the current study to measure the distance between two Wi-Fi stations: in our case, the user's mobile device and AP. The great advantage of using the RTT technique is its ability to measure the distance between two stations without necessitating explicit synchronization. It is worth noting that synchronization is one of the most important challenges of time-based localization. The RTT technique has been supported recently by the development of the FTM protocol in IEEE 802.11-2016.

The user's mobile device (the initiator) starts the process by sending a Wi-Fi signal to the AP (the receiver) to check its availability. The receiver confirms its availability by sending an acknowledge signal. A two-way communication is thus started between the two stations to measure the distance. This communication can be repeated several times for the sake of obtaining a more accurate distance estimation. The RTT distance estimation is performed for all APs lying in the range of the mobile device. Another feature of using the RTT technique is its ability to compute the distance at the edge side and thus the user's privacy is well preserved.

As shown in Figure 1, the process starts by sending a FTM request from the mobile device to the access point to see if it is available or not. The access point replies by the ACK signal if it is available and then the mobile device can compute the round-trip time by sending multiple FTM packets. The processing time at the mobile device side can be computed as follows:

$$T_p = t_3 - t_2 \quad (1)$$

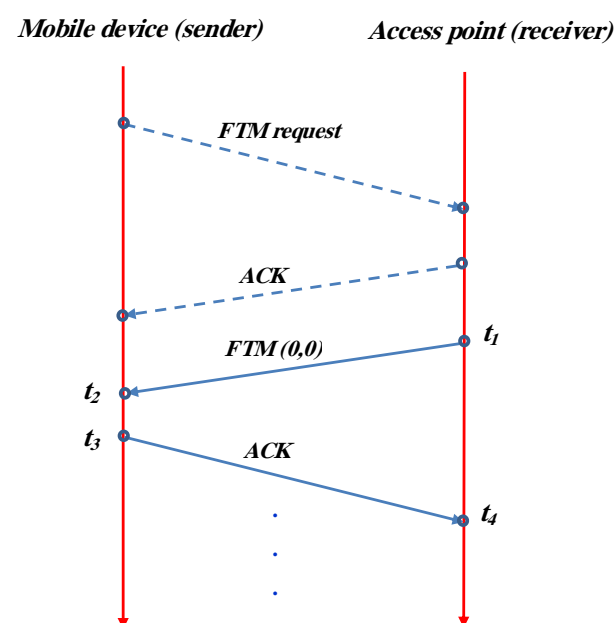


Figure 1. FTM protocol.

The round-trip time (RTT) can be computed as:

$$\text{RTT} = t_4 - t_1 - T_p \quad (2)$$

The distance (D) between the mobile device and the access point can be computed as:

$$D = \frac{1}{2} \text{RTT} \times C \quad (3)$$

where C is light speed which equals 3×10^5 km/s.

It is noteworthy that the mobile device performs RTT ranging to all RTT-capable APs in the vicinity. Different from multilateration approaches [17,44,45], *RRLoc* harnesses the collected RTT values (via the FTM protocol) as fingerprints (signatures for each location) as described in the following section.

3.2. Canonical Correlation Analysis

In this section, we provide a brief background on the traditional canonical correlation analysis (CCA) on which the DCCA is built. The details of our DCCA algorithm are given in Section 5.

CCA [35,36] is a standard highly versatile statistical method for finding common correlation for two multivariate sets of variables (vectors) having the same situations. In particular, CCA **linearly** projects the input sets into another lower-dimensional space in which these sets are maximally correlated. This helps in studying the strength of the relationship between two quantitative variables and how they are related. An appealing property of CCA for prediction tasks is that if there is noise in either set, the learned representations should not contain that noise in the new space.

More formally, assume $S = [s_1, s_2, \dots, s_N] \in \mathbb{R}^{d_x \times N}$ and $Y = [r_1, r_2, \dots, r_N] \in \mathbb{R}^{d_y \times N}$ are two different multivariate variable sets of N samples and feature space of dimensions d_x and d_y , respectively. The goal of CCA is to find K pairs of linear projections (canonical vectors) $W_s = [w_{s,1}, w_{s,2}, \dots, w_{s,K}] \in \mathbb{R}^{d_x \times K}$ and $W_r = [w_{r,1}, w_{r,2}, \dots, w_{r,K}] \in \mathbb{R}^{d_y \times K}$, so that the correlations between $W_s^T S$ and $W_r^T R$ are maximized. Specifically, CCA aims at finding the projection matrix that maximizes the correlation coefficient α between $W_s^T S$ and $W_r^T R$ as:

$$\alpha(W_s^T S, W_r^T R) = \frac{W_s^T S R^T W_r}{W_s^T S S^T W_s W_r^T R R^T W_r} \quad (4)$$

That is, we want to find:

$$(W_1^*, W_2^*) = \underset{W_s, W_r}{\operatorname{argmax}} \frac{W_s^T S R^T W_r}{W_s^T S S^T W_s W_r^T R R^T W_r} \quad (5)$$

Since α is scaling-invariant, we can rewrite the correlation as:

$$(W_1^*, W_2^*) = \underset{W_s, W_r}{\operatorname{argmax}} W_s^T S R^T W_r \quad (6)$$

$$s.t. \quad W_s^T S S^T W_s = 1, \quad W_r^T R R^T W_r = 1$$

To find the optimum solution for Equation (6), one has to solve the general eigenvalue problem of the form [46]:

$$\begin{bmatrix} \mathbf{0} & \Sigma_{sr} \\ \Sigma_{rs} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{W}_s \\ \mathbf{W}_r \end{bmatrix} = \lambda \begin{bmatrix} \hat{\Sigma}_{ss} & \mathbf{0} \\ \mathbf{0} & \hat{\Sigma}_{rr} \end{bmatrix} \begin{bmatrix} \mathbf{W}_s \\ \mathbf{W}_r \end{bmatrix} \quad (7)$$

where $\hat{\Sigma}_{ss}, \hat{\Sigma}_{rr}$ are the covariance matrices. Σ_{sr} and Σ_{rs} are defined as: $\Sigma_{sr} = \frac{1}{N} S R^T$ and $\Sigma_{rs} = \frac{1}{N} R S^T$.

By solving Equation (7), we obtain K eigenvectors $\{[\mathbf{W}_{s,k}; \mathbf{W}_{r,k}]\}_{k=1}^K$ and the corresponding K th eigenvalue that is equal to the correlation coefficient in Equation (4). Therefore, the aimed projection matrix W is the set of obtained eigenvectors.

In this paper, we adopt a deep-learning-based version of CCA, denoted DCCA [47], that can be viewed as a **nonlinear extension** of the traditional CCA.

4. System Overview

Figure 2 shows the *RRLoc* system architecture. *RRLoc* works in two stages: an offline calibration and training stage and an online localization stage. During the calibration stage, Wi-Fi data is collected at predefined reference points that uniformly cover the area of interest (This data can be transparently collected without the burden of site surveying using our earlier work in [48]). Typically, the collected data constitutes the fingerprints of each reference point involving the RTT and RSSI measurements from the overheard APs. This fingerprint map is constructed using the **Fingerprint Recorder** App running on a mobile phone and leverages the Android RTT API [49] to scan for RTT and RSSI readings. The collected fingerprint map is uploaded to an online running service for further processing. The **preprocessor** module is used to construct pairs of fixed size vectors (i.e., RSSI and RTT vectors) that are simultaneously captured from the APs overheard in the area of interest. Each pair of vectors is then forwarded to the **feature extractor** module to extract high-level location discriminative features. Specifically, this module learns the complex nonlinear transformation of the original low-level features to a new feature space where the RSSI and RTT projections are highly correlated as described in Section 5.2. Then, the obtained features are fed to the **localization model creator** module that is responsible for training a localization model for estimating the location of the mobile device. The output of this calibration stage is two trained models (i.e., the deep canonical correlation analysis model and the localization model) that are stored for later retrieval in the online stage.

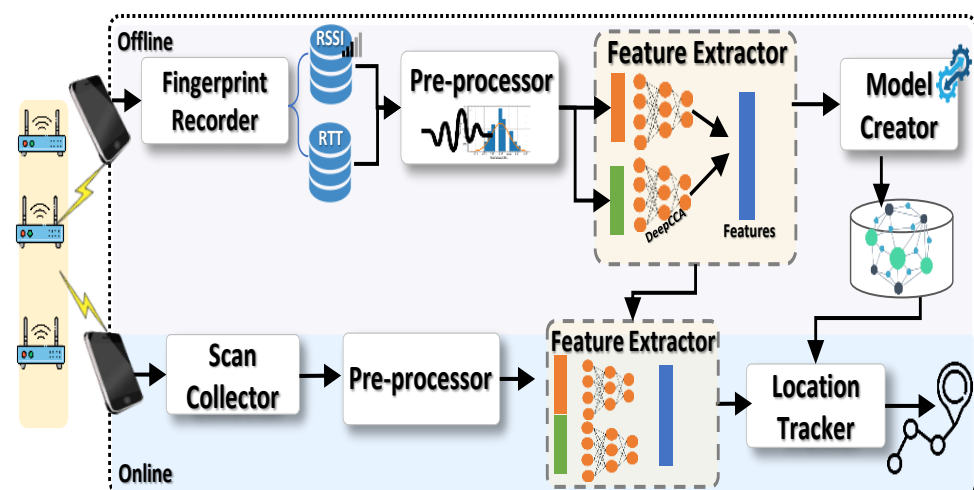


Figure 2. *RRLoc* system architecture.

During the online phase, users are tracked in real-time. When carrying their mobile phones at unknown locations, the phones scan for the APs in the vicinity. Each scan includes the RSSI and RTT from the overheard APs which are forwarded to the *RRLoc* server. This data is first handled by the **preprocessor** module to form the pairs of unified length vectors. Thereafter, these pairs are fed to the trained **DCCA model** to extract the desired features. Finally, the **location estimation model** feeds the extracted features to the localization model trained in the calibration stage to estimate the most probable reference locations where the user may be located. Based on these probabilities, the system obtains the user's location in the continuous spatial space.

5. The *RRLoc* System

Figure 2 shows the different modules of the *RRLoc* system. In the balance of this section, we describe the details of each module. Table 1 summarizes the notations used in this section.

Table 1. Notations used in the paper.

Notation	Description
q	The total number of access points covering the area of interest.
m	The number of access points detected in an arbitrary scan $\leq q$.
n	The total number of scans collected in the area of interest (i.e. at all locations).
l	The total number of reference location in the area of interest.
s	RSSI measurement vector which is composed of q entries, each of them represents the RSSI corresponding to one AP.
r	RTT measurement vector which is composed of q entries, each of them represents the RTT corresponding to one AP.
w	Matrix of weights.
b	Vector of biases.
z	The extracted feature vector as generated from the DCCA.
θ	The model parameters.
v	The size of the extracted feature vector.

5.1. The Preprocessor Module

The preprocessor module is responsible for mapping the RTT and RSSI measurements to a pair of fixed-length feature vectors. Each entry in the feature vector represents a measurement from an AP such that an AP entity in the RTT vector has a corresponding value in the RSSI vector. It is worth noting that all the installed APs cannot be overheard at every scan due to the range. Thus, only a subset of the APs may be detected in an arbitrary scan leading to variable-length feature vectors. To resolve this issue, nonheard APs in a specific scan are substituted by the RTT value of 0.2×10^{-3} ms, which is equivalent to a 60 m distance. This value is larger than any RTT value for the APs in the scanning range. Similarly, the RSSI value of -100 dBm is assigned to any unheard AP as it is lower than all RSSI values received from within the range of APs in the collected scans. Thus, a short RTT/low RSSI value is assigned to any AP lying far away from the mobile device carried by the user. It is also observed that when the mobile device is very close to an arbitrary AP, a negative distance is reported by the Android API [49]. This can be explained due to the internal configuration and calibration of the Wi-Fi cards or the multipath compensation algorithms that process the measurements in firmware before the driver receives them. RTT may also suffer from some latency when used with fast moving mobile devices. The presence of such negative values (former case) or latency (latter case) usually leads to a significant drop in the performance of traditional multilateration approaches [17]. However, this event cannot affect the *RRLoc*'s performance as it is a fingerprinting-based technique, and such negative values or delay can be considered a strong signature of particular locations.

Finally, normalization is employed to rescale the input values of each modality to be in the range between $[0, 1]$. Normalization has been empirically verified to speed up model convergence during training [50].

5.2. The Feature Extractor Module

This module aims to transform the preprocessed RSSI feature vectors and the corresponding RTT vectors to a latent space in which they are highly correlated. This is a more flexible feature-based fusion approach of different modalities given their spatial dependency while avoiding spurious measurements. In other words, both the RTT and RSSI of an arbitrary AP are different representations of how far the mobile device is from that AP.

Therefore, projecting the two modalities into another hyperspace where intercorrelation is maximum leads to more separable location signatures and thus better localization.

To discover that latent space, we adopt DCCA [47]. Traditional techniques, e.g., [14] obtain a joint multimodal representation by concatenating the individual RSSI and RTT vectors. Despite the simplicity of that approach, it loses essential information about the correlation (i.e., spatial dependency) between the input modalities. Moreover, compared to a single modality, the joint representation usually leads to a worse performance due to the presence of different types of noise and latency associated with data of varying nature. Compared to the classical CCA [35,36], which linearly transforms the input views into highly correlated projections, the DCCA solves the same objective function by realizing more powerful **nonlinear** projections in a new latent space using deep neural networks. These projections are learned via the gradient descent technique. The intuition behind leveraging the deep version of CCA is the ability of the deep neural network to learn complex relations from such noisy Wi-Fi data automatically. Unlike traditional deep-learning methods that are trained to maximize the likelihood of target class (location) given the RSSI scan alone [29] or the RTT scans alone [34], *RRLoc* combines both modalities using the correlation-based objective function of the DCCA. This empowers the **system robustness and learning ability** compared to just concatenating the noisy raw measurements that may deceive the localization model (as evaluated in Section 6.2.3).

Figure 3 shows the schematic structure of the proposed DCCA feature extraction model. As shown in the figure, the DCCA consists of two independent deep neural networks (DNNs), one for each type of measurements (RSSI and RTT). Each DNN consists of cascaded fully connected layers. The input layer of the DNN *A* and DNN *B* are the RSSI and RTT vectors which are captured simultaneously by the mobile device. These DNNs are then trained to encode these inputs to a fixed-size subspace where the corresponding output vectors (z_A and z_B) are maximally correlated. Specifically, let S_A be a set of RSSI input vectors, S_B is the corresponding set of RTT vectors which are collected simultaneously at the same set of reference points. These modalities are fed to the DCCA twin networks to obtain the aimed latent representations that leverage the advantages of both modalities. For instance, the output of the first layer of network *A* is $h_1^A = \sigma(W_1^A S + b_1^A)$, where σ is a nonlinear activation function (e.g., logistic Sigmoid) applied component-wise, W_1^A is a matrix of weights and b_1^A is a vector of biases. The output of each layer is used to calculate the output of the next layer until the final layer *d* whose output is calculated based on the output of the previous layer h_{d-1} as $f_A(S_A) = \sigma(W_d^A h_{d-1} + b_d^A)$ which is the intended latent representation (z), i.e., the spatially correlated feature vector. Similarly, the representation obtained by the second DNN of *g* layers is $f_B(S_B) = \sigma(W_g^B h_{g-1} + b_g^B)$ with different parameters W_g^B , b_g^B and *g*. The objective of the DCCA is to jointly learn the parameters θ_A and θ_B for both neural networks such that the correlation between z_A and z_B is maximum. Therefore, the objective function of the DCCA is defined as follows:

$$(\theta_A^*, \theta_B^*) = \underset{(\theta_A, \theta_B)}{\operatorname{argmax}} \operatorname{Corr}(f_A(S_A; \theta_A), f_B(S_B; \theta_B)) \quad (8)$$

To achieve this, we compute the correlation and its gradient with respect to the output layers. Then the back-propagation is used to update the parameters of both networks. This process is repeated until convergence is obtained (Given that, in general, optimization of deep models may not achieve the best performance if the model parameters are initialized randomly. Therefore, we adopt the Xavier initialization approach in [51] for better initialization of the feature extraction module).

After the training of the twin networks, the transformed feature vectors z_A and z_B become in the coordinated hyperspace. Then, these vectors are fed to the localization model as input for weighting with respect to their contribution in the location recognition process (as discussed in Section 5.3).

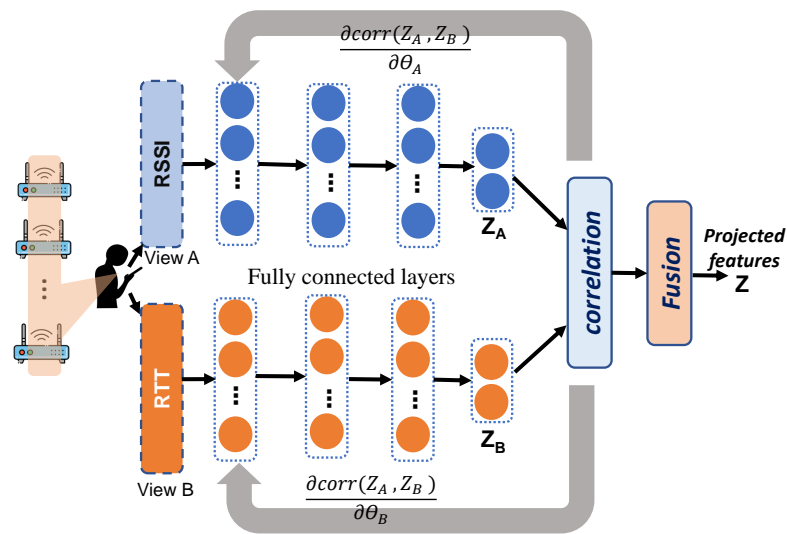


Figure 3. The network structure of the DCCA-based feature extraction module. It consists of two deep networks learned so that the output layers (topmost layer of each network) are maximally correlated. A correlation layer is stacked on top of a fully connected layer to calculate the correlation between the views.

5.3. Location Estimation Module

This module is responsible for utilizing the correlation features (z) extracted from the DCCA network to train the localization model and find its optimal parameters. The trained model is used in the online phase by the *online location predictor* module.

Figure 4 shows the structure of the considered deep neural network for localization. Specifically, *RRLoc* adopts a fully connected feed forward neural network. The hierarchical representation of *RRLoc* is obtained by cascaded hidden layers of nonlinear processing units. The rectified linear unit (ReLU) (the state of the art of nonlinearity) is used as the activation/transfer function for the hidden layers due to its sparsity and immunity to vanishing gradient problems [52].

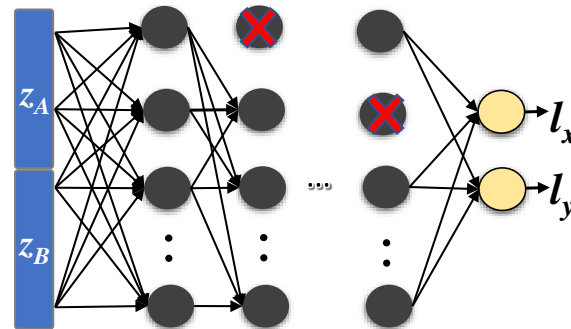


Figure 4. The network structure of the location estimator module. Crossed neurons represent dropped out units to avoid overfitting during training.

The input layer of the network is a vector z of length v which is obtained from the *feature extraction* module (described in Section 5.2). The network is trained to operate as a regression model having an output layer consisting of two neurons (corresponding to the 2D spatial coordinates (l_x, l_y)). Therefore, the selected network can be classified as a many-to-one, i.e. the model will learn a function that maps the latent representation z of the RSSI and RTT to an output location.

One advantage of designing the *localization model* to operate as a regressor rather than a classifier is the requirement to estimate the user location in the continuous space. Classification models can only estimate the user locations at one of the predefined few

discrete reference points. This usually leads to a bad user experience as the predicted locations will be spaced out even with a very accurate model. To ensure the required smooth tracking of the users in the continuous spatial space (*RRLoc* can locate the user anywhere, even on locations different from reference points.), *RRLoc* models the localization process as a deep-learning-based regressor that is trained to estimate the user location coordinates in the environment (even the nonsurveyed ones). We utilized the Adam optimizer [53] and mean square error (MSE) as a loss function.

DNN are known to have a tendency to overfit the training data, reducing their predictive skill [54]. Therefore, we utilize two regularization techniques: First, we use *dropout* to probabilistically exclude neurons and their connections from activation and weight updates while training a network. Second, we leverage *early stopping* so that training would terminate once the validation set no longer obtains performance improvements [55].

5.4. Online Phase

The goal of this phase is to track the users' locations in the environment of interest. Initially, each user device captures the RSSI and RTT from the detectable APs in the environment, and forward the scan to our running service to preprocess and extract the coordinated feature vectors from the trained DCCA, as described in Section 5.2. These vectors are then fed to the trained localization model to obtain a location estimate in the continuous space.

6. Evaluation

In this section, the data collection setup and tools used are described first. Then, we show how the system performs by varying the different system parameters. Finally, we compare the performance of *RRLoc* to the state-of-the-art techniques.

6.1. Collection Setup and Tools

For analyzing and evaluating the *RRLoc* system performance, we deployed the system in two realistic indoor testbeds. Table 2 summarizes the characteristics of the two testbeds. The first one, denoted as "Lab", is a full floor in our university campus which spans an area of 629 m² and contains nine rooms of different sizes and a long corridor as shown in Figure 5. The second testbed, denoted "Office", as shown in Figure 6 is an administrative building of 141 m² area consisting of a large meeting room, a long corridor, and five rooms. We used a wireless network setup of seven Google Wi-Fi APs uniformly distributed to cover the whole area of interest in both testbeds. The area of interest in both testbeds was uniformly discretized into different reference points distributed over the area one meter apart from each other. (We evaluate the effect of changing the spacing between reference points later in this section.) The Lab testbed had 143 different reference locations, while the second testbed included 76 locations. Each reference location was ensured to be covered by at least one Google Wi-Fi AP.

Data were collected with an Android application installed on different Android phones including Google Pixel XL and a Pixel 2XL. The application continuously scanned for the nearby APs. To facilitate ground-truth profiling, our data collector application ran synchronously on all mobile devices with one device dedicated to controlling ground-truth collection for all devices. The user input the coordinates of his current location (ground-truth) and launched the data collection process. At each reference location, at least 100 samples were captured in 3 min for training purposes. Hold-out test sets were collected independently, including 21 and 30 locations (different from the training points) in the Office testbed and the Lab testbed, respectively. This was completed over several days during working hours (to consider the time variation of signals indoors).

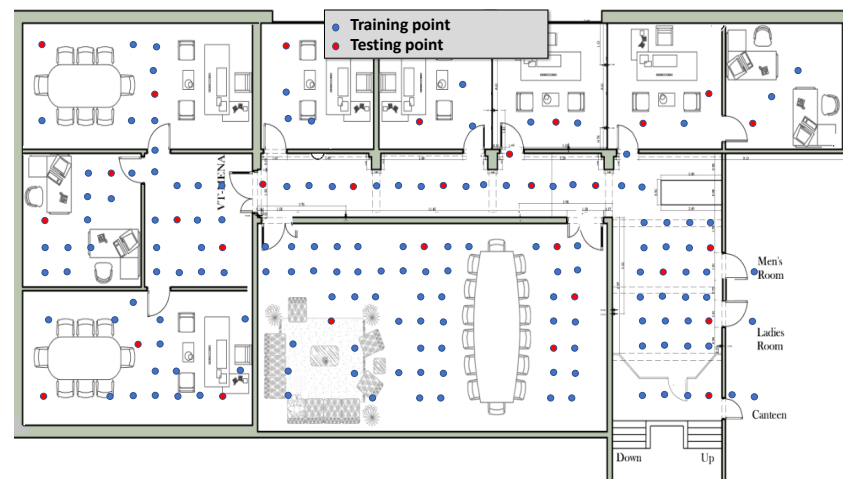


Figure 5. Layout of Lab testbed. Blue and red circles represent training and testing points, respectively.

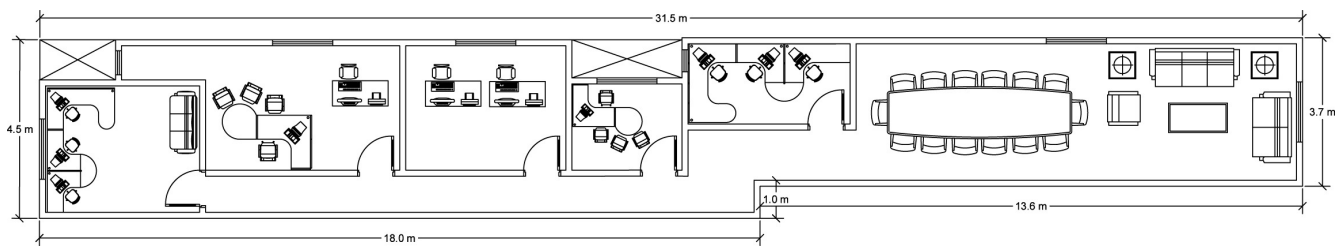


Figure 6. Layout of the Office testbed.

Table 2. Summary of the testbeds considered in evaluating *RRLoc*.

Criteria	Lab Testbed	Office Testbed
Area (m ²)	17×37	4.5×31.5
Number of training points	143	76
Number of testing points	30	21
Spacing of seed points (m)	1	1
Building Material	Brick	Brick& Wood
Number of APs	7	7
Total fingerprinting time (hrs:mins)	~08:39	~04:51
Training time (hrs:mins)	~01:05	~ 0:43

6.2. Effect of Changing *RRLoc* Parameters

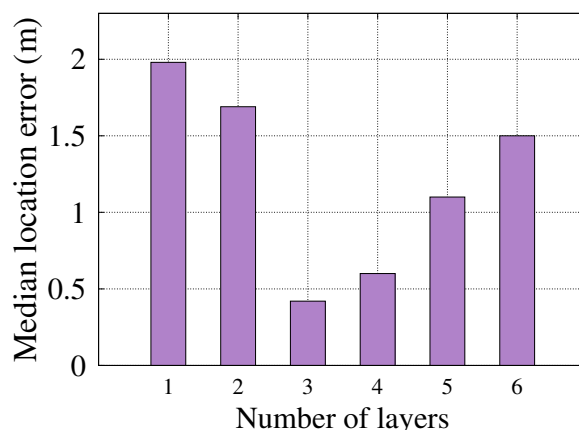
In this section, we study the impact of the different system parameters including the deep model's hyperparameters on the *RRLoc* performance and how much they enable learning the nonlinear transformations for achieving the maximum correlation between all modalities and thus better localization accuracy. These parameters include the number of layers, the effect of the feature extraction method, and the size of the feature vector. In the following subsections, we show the effect of varying these parameters only on the Lab testbed for clarity of presentation. However, we present how *RRLoc* performs in both testbeds in Subsection 6.3.1. Table 3 summarizes the default values of system parameters that are used throughout the evaluation section.

Table 3. Default parameters values used in the evaluation.

Parameter	Range	Default
Learning rate	0.0001–0.2	0.001
Number of hidden neurons	20–1000	300
Batch size	1–Dataset size	128
Number of layers	1–30	3
Early Stopping Patience (epochs)	1–10	40
Number of samples per reference point	20–100	100
Number of epochs	Automatic by Early stopping	
Used devices	Google Pixel XL, Google Pixel 2XL	
Number of users	3	
Update rate (scan/sec)	2	

6.2.1. Number of Layers in the Network

Figure 7 shows the effect of changing the number of layers on *RRLoc* performance. As shown, the more hidden layers to consider, the better the accuracy (i.e., less localization error) *RRLoc* can achieve until it reaches an optimal value at three layers. This can be justified due to two reasons. First, increasing the number of layers increases the distributed learning ability of the localization model. In this vein, the model has enhanced computing power enabling the better fitting of the underlying function (without underfitting). Second, as few as three layers are enough to allow the localization model to learn the user location from the latent extracted features. It is worth noting that the extracted features obtained by the DCCA radically simplify the classification problem. Beyond three layers, the model tends to overfit the training data, reducing its flexibility and thus its accuracy. As a result, *three* layers is set as the default number of layers in the *RRLoc* model to achieve a balance between underfitting and overfitting.

**Figure 7.** Effect of changing the number of layers on *RRLoc* localization error.

6.2.2. Dropout Percentage

The effect of increasing the percentage of dropout is shown in Figure 8. It can be observed from the figure that at a rate of 0.1 dropout, the best performance of *RRLoc* is achieved (even better than the case of no dropout). This confirms the significant regularization role of dropout in boosting the network learning to generalize rather than overfit the training data.

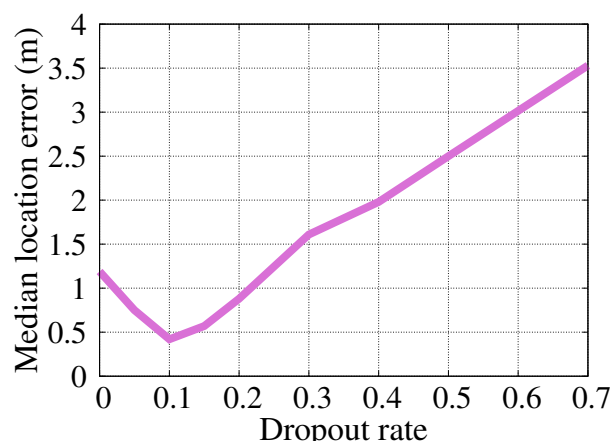


Figure 8. Effect of changing the dropout rate on *RRLoc* localization error.

6.2.3. Feature Extraction Method

In this section, we study the influence of the different feature extraction techniques on the overall system performance. Figure 9 compares the effect of using DCCA for extracting discriminative features to either using joint representation (i.e., concatenated RSSI and RTT features) or feature projection using the classic CCA [36]. The figure confirms the favorable performance of the DCCA of *RRLoc* as compared to both the joint representation and classic CCA. Specifically, *RRLoc* gives an improvement of 83%, and 186% in estimating the correct user location as compared to the joint representation and classic CCA, respectively. This can be explained by noting that the classifier tries to learn the underlying distribution of the input and map it to the output. However, in the case of the joint representation, the classifier is supplied with mixed input of two different distributions, which generally yields an information loss of the correlation between the two modalities. On the other hand, the classical CCA assumes that a linear transformation of the inputs improves correlation between RTT and RSSI in the new space, which is not always the case in practice due to the propagation challenges in indoor environments. These results highlight the efficacy of using the DCCA in capturing the nonlinear correlated signatures of the two Wi-Fi input modalities, facilitating the accurate tracking of the user location.

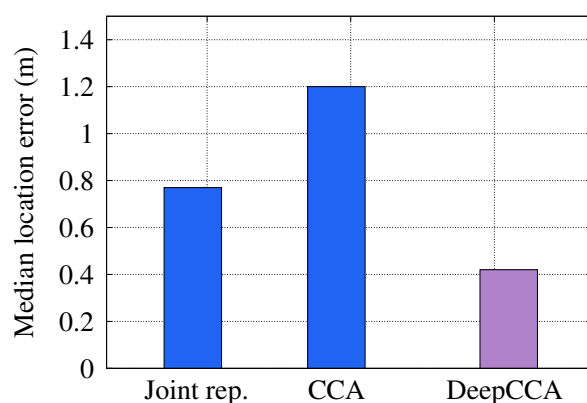


Figure 9. Effect of feature extraction module on *RRLoc* performance.

6.2.4. Feature Vector Length

Figure 10 shows the location estimation accuracy of *RRLoc* as a function of the latent space dimension size obtained by the DCCA network. It is clear from the figure that increasing the size (dimensions) of the latent feature vector z improves the *RRLoc* performance. The figure also shows that a feature vector z of five dimensions yields the best performance. Beyond five dimensions, a performance deterioration is observed. This can be explained as

the additional dimensions usually include undesired artifacts that reduce the correlation between the input modalities.

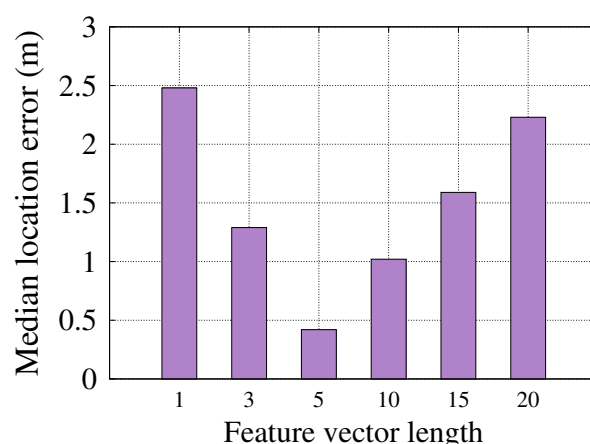


Figure 10. Effect of changing the feature vector length on *RRLoc* accuracy.

6.2.5. Impact of Each Modality

Figure 11 shows the performance of the *RRLoc* system when an individual modality is used as well as the hybrid RSSI-RTT version. The figure shows the favorable performance of RTT compared to the RSSI. This can be justified by noting that the RSSI readings are more noisy compared to RTT due to its higher sensitivity to multipath effect. The combination of RSSI and RTT yields an improvement in the median accuracy by 289% and 129% compared to RSSI-only and RTT-only, respectively. This is due to leveraging the advantages of both modalities leading to a remarkable enhancement of the *RRLoc* accuracy in all percentiles. Specifically, the RSSI measurements boost the *RRLoc* performance compared to using the RTT counterpart alone in cases of the absence of a direct line-of-sight transmission, while RTT maintains the system robustness in case of noisy and fluctuating RSSI signals. These results validate the gain of such combination on the system performance.

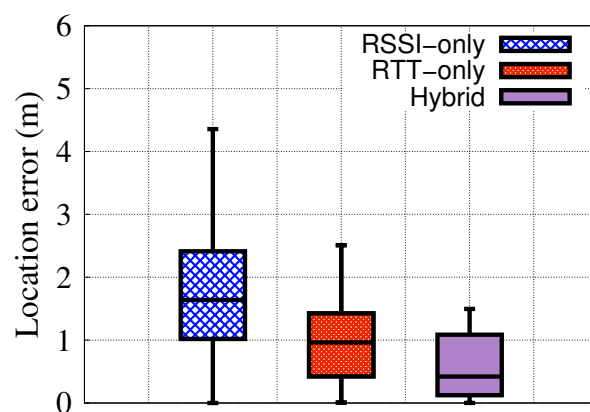


Figure 11. Effect of varying the considered modality on *RRLoc* performance.

6.2.6. Effect of Access Points Density

Figure 12 shows the impact of varying the number of access points in use on the median localization error and response time (i.e., average time per location estimate). This experiment is performed by removing APs from the feature vector randomly. The figure shows that increasing the APs installed in the area of interest yields a better localization accuracy. This can be explained by noting that increasing the number of APs leads to richer feature vectors to accommodate the blocking that might occur to any connection between the transmitter and any receiver at the run time.

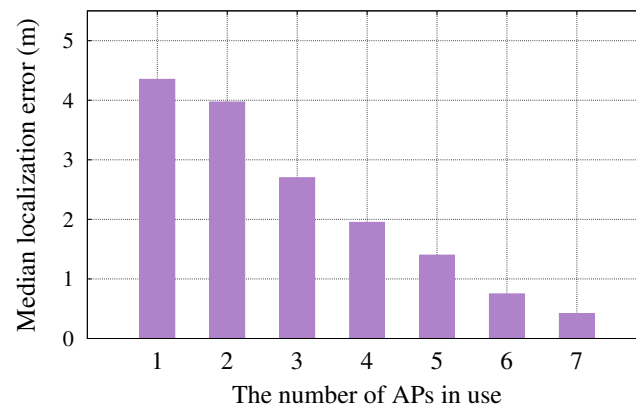


Figure 12. Effect of changing the number of access points on *RRLoc* performance.

6.2.7. Effect of Fingerprint Points Density

In this section, we study the effect of the density of fingerprint points in the area of interest on the *RRLoc* performance. Figure 13 shows that increasing the density (shorter spacing between points) of the fingerprint points leads to better localization accuracy. However, unlike traditional RSSI-based systems [29,48,56], increasing the spacing between fingerprint points leads to a slight decrease in the performance of *RRLoc*. In particular, the *RRLoc* localization accuracy loses just 32 cm by doubling the fingerprint spacing from 1 m to 2 m. This confirms the *RRLoc*'s ability to operate at lower fingerprint densities, leading to a drastic saving of time and effort associated with the data collection process.

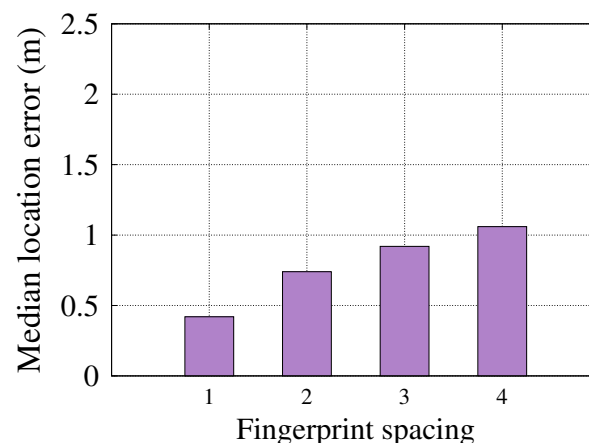


Figure 13. Effect of varying the spacing between reference points on *RRLoc* performance.

6.3. Comparative Evaluation

In this section, the performance of *RRLoc* is compared to three Wi-Fi-based localization systems: WiNar [14], WiDeep [29] and ranging-based system [17]. WiNar [14] leverages a deterministic approach that matches the captured RTT measurements to the prerecorded fingerprint map to estimate the user's location. WiNar leverages the RSSI to weigh the estimated locations. On the other hand, WiDeep [29] builds a RSSI-based localization system using the deep denoising autoencoder neural network. The ranging-based system in [17] uses the multilateration approach based on the RTT for enabling indoor localization while detecting NLOS. All techniques have been evaluated on the same data for a fair comparison.

6.3.1. Localization Accuracy

Figures 14 and 15 show the CDF of localization error of the different techniques in the two testbeds. Figure 14 shows that *RRLoc* gives an improvement in median error obtained

in the Office testbed by 129%, 267%, and 632% compared to the WiNar [14], WiDeep [29] and ranging-based [17] systems, respectively. On the other hand, the performance of the different systems in the Lab testbed (i.e. the larger testbed) is shown in Figure 15. The results depicted in the figure show that *RRLoc* outperforms WiNar [14], WiDeep [29] and ranging-based [17] systems by 45%, 337% and 469%, respectively. In summary, as shown in Tables 4 and 5, *RRLoc* improves all the percentiles upon the other system in both testbeds. This can be explained by noting that the concatenation-based approach of WiNar loses the correlation information between RSSI and RTT measurements. Moreover, the deterministic matching method adopted by WiNar cannot cope with the noisy measurements of both modalities. WiDeep leverages only RSSI measurements to train a powerful deep-learning model for localization purposes. Nevertheless, the accuracy of this approach depends on the quality of the captured signals, which are generally noisy in cluttered environments. The performance of a ranging-based multilateration system usually suffers from NLOS problems leading to coarse-grained accuracy. Different from these systems, *RRLoc* leverages the flexibility of the proposed DCCA-based approach to maximize the benefits of RSSI-RTT fusion. Additionally, *RRLoc* considers the spatial dependency between the two modalities through a powerful deep neural network that learns robust location-discriminative features given the inherent noise in each modality. This highlights the promise of *RRLoc* as the next generation of robust Wi-Fi-based positioning system.

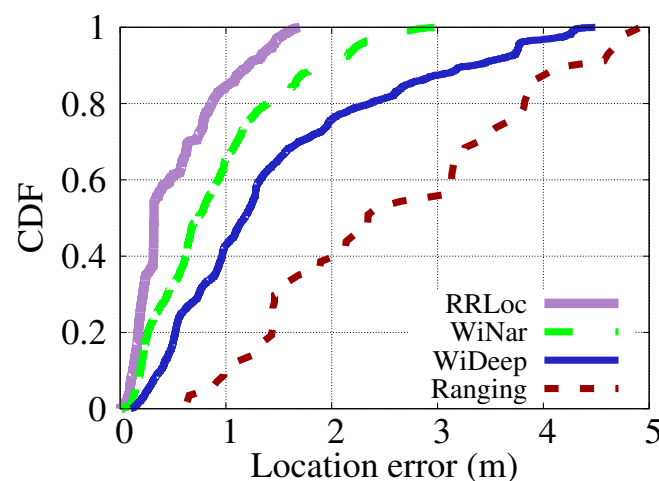


Figure 14. Comparison of CDFs of different systems in the Office testbed.

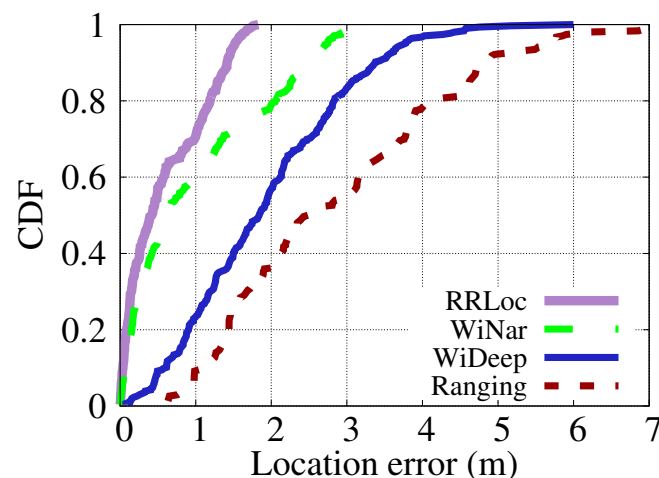


Figure 15. Comparison of CDFs of different systems in the Lab testbed.

Table 4. The localization error percentiles in Office testbed.

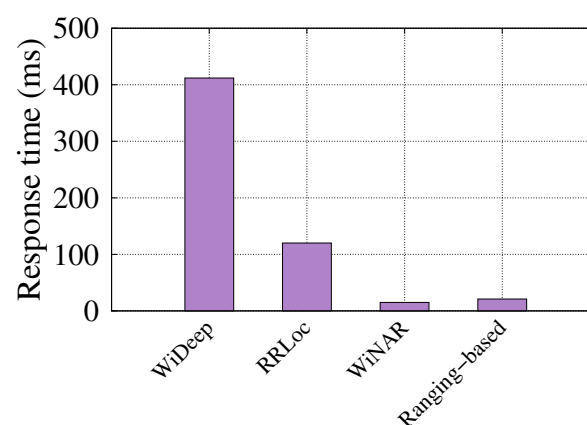
Technique	Average	25th Percentile	50th Percentile	75th Percentile	Maximum
<i>RRLoc</i>	0.51 m	0.19 m	0.32 m	0.79 m	1.70 m
<i>WiNar</i> [14]	0.89 m (−72%)	0.34 m (−82%)	0.73 m (−129%)	1.20 m (−53%)	2.99 m (−76%)
<i>WiDeep</i> [29]	1.46 m (−183%)	0.58 m (−208%)	1.17 m (−267%)	1.97 m (−151%)	4.49 m (−164%)
<i>Ranging</i> [17]	2.59 m (−401%)	1.44 m (−664%)	2.34 m (−632%)	3.68 m (−368%)	4.92 m (−189%)

Table 5. The localization error percentiles in the Lab testbed.

Technique	Average	25th Percentile	50th Percentile	75th Percentile	Maximum
<i>RRLoc</i>	0.59 m	0.12 m	0.42 m	1.08 m	1.83 m
<i>WiNar</i> [14]	0.99 m (−69%)	0.19 m (−51%)	0.61 m (−45%)	1.77 m (−63%)	3.0 m (−64%)
<i>WiDeep</i> [29]	1.92 m (−226%)	1.06 m (−753%)	1.84 m (−337%)	2.69 m (−149%)	6.00 m (−228%)
<i>Ranging</i> [17]	2.86 m (−384%)	1.46 m (−1077%)	2.51 m (−496%)	3.85 m (−255%)	7.38 m (−304%)

6.3.2. Time per Location Estimate

We used a a Lenovo Thinkpad X1 laptop running a 2.2 GHz Intel i7-8750H processor with 64 GB RAM, and a Nvidia GTX1050Ti 4GB GPU for evaluating the running time of the different systems. Figure 16 shows the results. The figure shows that as *RRLoc* and *WiDeep* [29] are deep-learning-based systems, they need to pass the input through all the layers of the network to calculate a location estimate. This takes more time than the traditional deterministic method used in *WiNar* [14] and ranging-based [17]. On the other hand, *RRLoc* has running time that is remarkably less than *WiDeep* [29]. This is due to the fewer number of layers and neurons and, by extension, less calculations compared to *WiDeep* [29]. Nonetheless, all systems allow real-time tracking of the user which can be further enhanced (if needed) by parallelization.

**Figure 16.** Comparison of run time of the different systems.

6.3.3. Device Heterogeneity

In this section, we investigate the system robustness to device heterogeneity, where one device is dedicated to capturing training data, and the other is used for testing. Figure 17 shows the system performance when varying the testing devices in the two testbeds, i.e., in the case of testing with Pixel XL, the training device is Pixel 2XL and vice versa. The figure shows that *RRLoc* achieves a consistent performance in all cases for the two devices

which is slightly better in the case of Pixel 2XL. This can be justified as different phones, in general, vary in hardware factors, e.g., form factors, chips, antenna locations, leading to a variation of the measured RSSI. The combined effect of these factors can be considered to be an offset that affects only the RSSI depending on the phone as addressed in [57–59]. However, hardware diversity has a negligible effect on RTT measurements [34]. The fusion of RSSI and RTT has shown to be effective in mitigating the effect of hardware diversity, leading to a more robust localization performance. It is worth mentioning that currently few devices support RTT scanning. However, the number of models of supported devices is increasing [49].

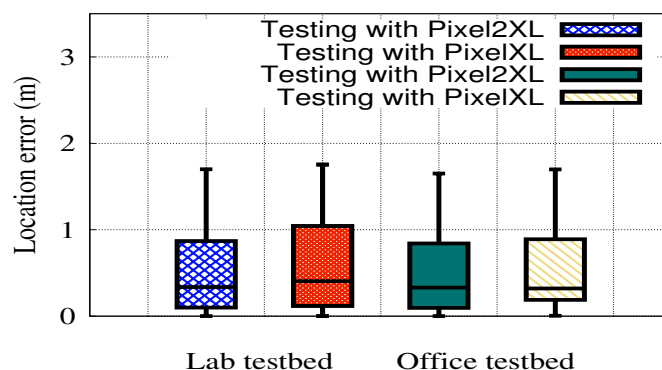


Figure 17. Effect of varying the testing device on *RRLoc* performance in the two testbeds.

7. Conclusions

We presented *RRLoc*, a hybrid deep-learning-based indoor localization system which fuses fingerprinting and time-based techniques to combine the merits of both techniques. A novel approach is adopted for fusing RSSI and RTT measurements and extracting high-level features using deep canonical correlation analysis. The proposed *RRLoc* showed great capabilities in overcoming the challenges of these techniques in indoor environments even with the use of heterogeneous devices. Different modules are incorporated to improve the deep model's generalization against overtraining and noise. The proposed system is evaluated in two different environments (office and lab). The *RRLoc* system achieved a submeter localization accuracy for both indoor environments with a median localization error of 0.42m and 0.32m respectively. *RRLoc* is able to improve upon all the percentiles of the other systems in both environments when using the coordinated representation of RTT and RSSI data. In the future, we plan to deploy *RRLoc* at scale and automate the data collection process to lessen the fingerprinting burden.

Author Contributions: Conceptualization, H.R., A.E. and H.Y.; methodology, H.R., A.E. and H.Y.; software, H.R.; validation, H.R., A.E. and H.Y.; formal analysis, H.R.; investigation, H.R., A.E. and H.Y.; resources, H.R. and H.Y.; data curation, H.R.; writing—original draft preparation, H.R. and A.E.; writing—review and editing, H.R. and A.E.; visualization, H.R., A.E. and H.Y.; supervision, H.Y.; project administration, H.Y.; funding acquisition, H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by JST, CREST Grant Number JPMJCR21M5, Japan.

Data Availability Statement: Not applicable.

Acknowledgments: The authors extend their appreciation to JST for supporting this work by CREST Grant Number JPMJCR21M5, and JSPS, KAKENHI Grant number 22569883, Japan. The authors would like to thank WRC group headed by Moustafa Youssef for their assistance in the data collection process.

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

References

1. Büyükcöçak, S.; Erbaş, T.; Kurt, G.K.; Yongaçoglu, A. Indoor localization applications. In Proceedings of the 2014 22nd Signal Processing and Communications Applications Conference (SIU), Trabzon, Turkey, 23–25 April 2014; pp. 1239–1242. [\[CrossRef\]](#)
2. Emam, A.; Mtibaa, A.; Harras, K.A.; Mohamed, A. Adaptive forwarding of mHealth data in challenged networks. In Proceedings of the 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian, China, 12–15 October 2017; pp. 1–7. [\[CrossRef\]](#)
3. Bahl, P.; Padmanabhan, V. RADAR: An in-building RF-based user location and tracking system. In Proceedings of the IEEE INFOCOM 2000, Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), Tel Aviv, Israel, 26–30 March 2000; Volume 2, pp. 775–784. [\[CrossRef\]](#)
4. Rizk, H.; Torki, M.; Youssef, M. CellinDeep: Robust and Accurate Cellular-based Indoor Localization via Deep Learning. *IEEE Sens. J.* **2018**, *19*, 2305–2312. [\[CrossRef\]](#)
5. Aly, H.; Basalamah, A.; Youssef, M. Accurate and Energy-Efficient GPS-Less Outdoor Localization. *ACM Trans. Spat. Algorithms Syst.* **2017**, *3*, 1–31. [\[CrossRef\]](#)
6. Jung, S.Y.; Hann, S.; Park, C.S. TDOA-based optical wireless indoor localization using LED ceiling lamps. *IEEE Trans. Consum. Electron.* **2011**, *57*, 1592–1597. [\[CrossRef\]](#)
7. Wang, T.Q.; Sekercioglu, Y.A.; Neild, A.; Armstrong, J. Position Accuracy of Time-of-Arrival Based Ranging Using Visible Light With Application in Indoor Localization Systems. *J. Light. Technol.* **2013**, *31*, 3302–3308. [\[CrossRef\]](#)
8. Lanzisera, S.; Lin, D.T.; Pister, K.S.J. RF Time of Flight Ranging for Wireless Sensor Network Localization. In Proceedings of the 2006 International Workshop on Intelligent Solutions in Embedded Systems, Vienna, Austria, 30 June 2006; pp. 1–12. [\[CrossRef\]](#)
9. Bahillo, A.; Mazuelas, S.; Prieto, J.; Fernández, P.; Lorenzo, R.; Abril, E. Hybrid RSS-RTT localization scheme for wireless networks. In Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation, Zurich, Switzerland, 15–17 September 2010; pp. 1–7. [\[CrossRef\]](#)
10. Zafari, F.; Gkelias, A.; Leung, K.K. A Survey of Indoor Localization Systems and Technologies. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 2568–2599. [\[CrossRef\]](#)
11. Al Nuaimi, K.; Kamel, H. A survey of indoor positioning systems and algorithms. In Proceedings of the 2011 International Conference on Innovations in Information Technology, Abu Dhabi, United Arab Emirates, 25–27 April 2011; pp. 185–190. [\[CrossRef\]](#)
12. Mautz, R. Overview of Current Indoor Positioning Systems. *Geod. Cartogr.* **2009**, *35*, 18–22. [\[CrossRef\]](#)
13. Liu, Z.; Zhang, L.; Liu, Q.; Yin, Y.; Cheng, L.; Zimmermann, R. Fusion of Magnetic and Visual Sensors for Indoor Localization: Infrastructure-Free and More Effective. *IEEE Trans. Multimed.* **2017**, *19*, 874–888. [\[CrossRef\]](#)
14. Hashem, O.; Harras, K.; Youssef, M. Accurate indoor positioning using IEEE 802.11mc round-trip time. *Pervasive Mob. Comput.* **2021**, *75*, 101416. [\[CrossRef\]](#)
15. Schauer, L.; Dorfmeister, F.; Maier, M. Potentials and limitations of Wi-Fi-positioning using Time-of-Flight. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation, Montbeliard, France, 28–31 October 2013; pp. 1–9. [\[CrossRef\]](#)
16. Mtibaa, A.; Harras, K.; Abdellatif, M. Exploiting social information for dynamic tuning in cluster based Wi-Fi localization. In Proceedings of the 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Abu Dhabi, United Arab Emirates, 19–21 October 2015; pp. 868–875. [\[CrossRef\]](#)
17. Ibrahim, M.; Liu, H.; Jawahar, M.; Nguyen, V.; Gruteser, M.; Howard, R.; Yu, B.; Bai, F. Verification: Accuracy evaluation of Wi-Fi fine time measurements on an open platform. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, New Delhi, India, 29 October–2 November 2018; pp. 417–427.
18. Hilsenbeck, S.; Bobkov, D.; Huitl, R.; Schroth, G.; Steinbach, E. Graph-based data fusion of pedometer and Wi-Fi measurements for mobile indoor positioning. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous (UbiComp), Seattle, WA, USA, 13–17 September 2014. [\[CrossRef\]](#)
19. Wu, K.; Xiao, J.; Yi, Y.; Chen, D.; Luo, X.; Ni, L.M. CSI-Based Indoor Localization. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *24*, 1300–1309. [\[CrossRef\]](#)
20. Jiang, Z.; Zhao, J.; Li, X.Y.; Xi, W.; Zhao, K.; Tang, S.; Han, J. Communicating Is Crowdsourcing: Wi-Fi Indoor Localization with CSI-Based Speed Estimation. *J. Comput. Sci. Technol.* **2013**, *29*. [\[CrossRef\]](#)
21. Shu, Y.; Bo, C.; Shen, G.; Zhao, C.; Li, L.; Zhao, F. Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic Wi-Fi Sensing. *IEEE J. Sel. Areas Commun.* **2015**, *33*, 1443–1457. [\[CrossRef\]](#)
22. Zhou, M.; Lin, Y.; Zhao, N.; Jiang, Q.; Yang, X.; Tian, Z. Indoor WLAN Intelligent Target Intrusion Sensing Using Ray-Aided Generative Adversarial Network. *IEEE Trans. Emerg. Top. Comput. Intell.* **2020**, *4*, 61–73. [\[CrossRef\]](#)
23. Mendoza-Silva, G.M.; Torres-Sospedra, J.; Huerta, J. A Meta-Review of Indoor Positioning Systems. *Sensors* **2019**, *19*, 4507. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Al-Ammar, M.A.; Alhadhrami, S.; Al-Salman, A.; Alarifi, A.; Al-Khalifa, H.S.; Alnafessah, A.; Alsaleh, M. Comparative Survey of Indoor Positioning Technologies, Techniques, and Algorithms. In Proceedings of the 2014 International Conference on Cyberworlds, Santander, Spain, 6–8 October 2014; pp. 245–252. [\[CrossRef\]](#)

25. Bisio, I.; Lavagetto, F.; Marchese, M.; Sciarrone, A. Energy Efficient Wi-Fi-based Fingerprinting for Indoor Positioning with Smartphones. In Proceedings of the 2013 IEEE Global Communications Conference (GLOBECOM), Atlanta, GA, USA, 9–13 December 2013.
26. Wang, X.; Gao, L.; Mao, S.; Pandey, S. CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach. *IEEE Trans. Veh. Technol.* **2017**, *66*, 763–776. [\[CrossRef\]](#)
27. Firdaus, F.; Ahmad, N.; Sahibuddin, S. A Review of Hybrid Indoor Positioning Systems Employing WLAN Fingerprinting and Image Processing. *Int. J. Electr. Comput. Eng. Syst.* **2020**, *10*, 59–72. [\[CrossRef\]](#)
28. Youssef, M.; Agrawala, A. The Horus WLAN location determination system. In Proceedings of the the 3rd International Conference on Mobile Systems, Applications, and Services, Seattle WA, USA, 6–8 June 2005; pp. 205–218.
29. Abbas, M.; Elhamshary, M.; Rizk, H.; Torki, M.; Youssef, M. WiDeep: Wi-Fi-based Accurate and Robust Indoor Localization System using Deep Learning. In Proceedings of the International Conference on Pervasive Computing and Communications (PerCom), Kyoto, Japan, 11–15 March 2019.
30. Bishop, C.M. Information science and statistics. In *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
31. Golden, S.A.; Bateman, S.S. Sensor Measurements for Wi-Fi Location with Emphasis on Time-of-Arrival Ranging. *IEEE Trans. Mob. Comput.* **2007**, *6*, 1185–1198. [\[CrossRef\]](#)
32. Chan, Y.T.; Tsui, W.Y.; So, H.C.; Ching, P.C. Time-of-arrival based localization under NLOS conditions. *IEEE Trans. Veh. Technol.* **2006**, *55*, 17–24. [\[CrossRef\]](#)
33. Ens, A.; Reindl, L.; Bordoy, J.; Wendeberg, J.; Schindelbauer, C. Unsynchronized ultrasound system for TDOA localization. In Proceedings of the 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Korea, 27–30 October 2014; pp. 601–610. [\[CrossRef\]](#)
34. Hashem, O.; Youssef, M.; Harras, K.A. WiNar: RTT-based Sub-meter Indoor Localization using Commercial Devices. In Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications (PerCom), Austin, TX, USA, 23–27 March 2020; pp. 1–10. [\[CrossRef\]](#)
35. Hotelling, H., Relations Between Two Sets of Variates. In *Breakthroughs in Statistics: Methodology and Distribution*; Kotz, S.; Johnson, N.L., Eds.; Springer: New York, NY, USA, 1992; pp. 162–190.
36. Anderson, W.T. *An Introduction to Multivariate Statistical Analysis*, 2nd ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1984.
37. Kumar, P.; Reddy, L.; Varma, S. Distance measurement and error estimation scheme for RSSI based localization in Wireless Sensor Networks. In Proceedings of the 2009 Fifth International Conference on Wireless Communication and Sensor Networks (WCSN), Allahabad, India, 15–19 December 2009; pp. 1–4. [\[CrossRef\]](#)
38. Wu, L.; Meng, M.Q.H.; Lin, Z.; He, W.; Chao, P.; Liang, H. A practical evaluation of radio signal strength for mobile robot localization. In Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guilin, China, 19–23 December 2009; pp. 516–522. [\[CrossRef\]](#)
39. Yu, Y.; Chen, R.; Chen, L.; Guo, G.; Ye, F.; Liu, Z. A Robust Dead Reckoning Algorithm Based on Wi-Fi FTM and Multiple Sensors. *Remote Sens.* **2019**, *11*, 504. [\[CrossRef\]](#)
40. Guo, G.; Chen, R.; Ye, F.; Peng, X.; Liu, Z.; Pan, Y. Indoor Smartphone Localization: A Hybrid Wi-Fi RTT-RSS Ranging Approach. *IEEE Access* **2019**, *7*, 176767–176781.
41. Sun, M.; Wang, Y.; Huang, L.; Jia, H.; Bi, J.; Joseph, W.; Plets, D. Geomagnetic Positioning-Aided Wi-Fi FTM Localization Algorithm for NLOS Environments. *IEEE Commun. Lett.* **2022**, *14*, 1–5. [\[CrossRef\]](#)
42. Cao, H.; Wang, Y.; Bi, J.; Xu, S.; Si, M.; Qi, H. Indoor Positioning Method Using Wi-Fi RTT Based on LOS Identification and Range Calibration. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 627. [\[CrossRef\]](#)
43. Han, K.; Yu, S.M.; Kim, S.L. Smartphone-based Indoor Localization Using Wi-Fi Fine Timing Measurement. In Proceedings of the 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Pisa, Italy, 30 September–3 October 2019; pp. 1–5.
44. Banin, L.; Schatzberg, U.; Amizur, Y. Wi-Fi FTM and map information fusion for accurate positioning. In Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Madrid, Spain, 4–7 October 2016.
45. Ciurana, M.; Barcelo-Arroyo, F.; Izquierdo, F. A ranging system with IEEE 802.11 data frames. In Proceedings of the 2007 IEEE Radio and Wireless Symposium, Long Beach, CA, USA, 7–12 January 2007; pp. 133–136.
46. Hardoon, D.R.; Szedmak, S.; Shawe-Taylor, J. Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.* **2004**, *16*, 2639–2664. [\[CrossRef\]](#) [\[PubMed\]](#)
47. Andrew, G.; Arora, R.; Bihmes, J.; Livescu, K. Deep Canonical Correlation Analysis. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; Dasgupta, S., McAllester, D., Eds.; PMLR: Atlanta, GA, USA, 2013; Volume 28, pp. 1247–1255.
48. Rizk, H.; Yamaguchi, H.; Youssef, M.; Higashino, T. Gain without pain: Enabling fingerprinting-based indoor localization using tracking scanners. In Proceedings of the 28th International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 3–6 November 2020; pp. 550–559.
49. Android API Level 28 (Wi-Fi Location: Ranging with RTT). Available online: <https://developer.android.com/guide/topics/connectivity/Wi-Fi-rtt> (accessed on 25 December 2021).

50. Kang, Y.; Lee, K.T.; Eun, J.; Park, S.E.; Choi, S. Stacked denoising autoencoders for face pose normalization. In Proceedings of the International Conference on Neural Information Processing, Daegu, Korea, 3–7 November 2013; pp. 241–248.
51. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and Composing Robust Features with Denoising Autoencoders. In Proceedings of the 25th International Conference on Machine Learning, ICML '08, Helsinki, Finland, 5–9 July 2008; Association for Computing Machinery: New York, NY, USA, 2008; pp. 1096–1103. [[CrossRef](#)]
52. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
53. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
54. Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent neural network regularization. *arXiv* **2014**, arXiv:1409.2329.
55. Bengio, Y. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 437–478.
56. Rizk, H.; Youssef, M. MonoDCell: A Ubiquitous and Low-Overhead Deep Learning-based Indoor Localization with Limited Cellular Information. In Proceedings of the 27th SIGSPATIAL International Conference on Advances in Geographic Information Systems, Chicago, IL, USA, 5–8 November 2019.
57. Rizk, H.; Abbas, M.; Youssef, M. Omnicells: Cross-device cellular-based indoor location tracking using deep neural networks. In Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications (PerCom), Austin, TX, USA, 23–27 March 2020; pp. 1–10.
58. Rizk, H. Device-Invariant Cellular-Based Indoor Localization System Using Deep Learning. In Proceedings of the ACM MobiSys 2019 on Rising Stars Forum, Seoul, Korea, 21 June 2019; pp. 19–23.
59. Rizk, H.; Abbas, M.; Youssef, M. Device-independent cellular-based indoor location tracking using deep learning. *Pervasive Mob. Comput.* **2021**, *75*, 101420. [[CrossRef](#)]