

Article

# RSSI and Device Pose Fusion for Fingerprinting-Based Indoor Smartphone Localization Systems

Imran Moez Khan <sup>1,\*</sup>, Andrew Thompson <sup>2</sup>, Akram Al-Hourani <sup>1</sup> , Kandeepan Sithamparanathan <sup>1</sup>   
and Wayne S. T. Rowe <sup>1</sup> 

<sup>1</sup> College of Science, Technology, Engineering and Mathematics, RMIT University, Melbourne, VIC 3000, Australia

<sup>2</sup> Robert Bosch Australia & New Zealand, Melbourne, VIC 3168, Australia

\* Correspondence: imran.khan@student.rmit.edu.au

**Abstract:** Complementing RSSI measurements at anchors with onboard smartphone accelerometer measurements is a popular research direction to improve the accuracy of indoor localization systems. This can be performed at different levels; for example, many studies have used pedestrian dead reckoning (PDR) and a filtering method at the algorithm level for sensor fusion. In this study, a novel conceptual framework was developed and applied at the data level that first utilizes accelerometer measurements to classify the smartphone’s device pose and then combines this with RSSI measurements. The framework was explored using neural networks with room-scale experimental data obtained from a Bluetooth low-energy (BLE) setup. Consistent accuracy improvement was obtained for the output localization classes (zones), with an average overall accuracy improvement of 10.7 percentage points for the RSSI-and-device-pose framework over that of RSSI-only localization.

**Keywords:** indoor localization systems; RF fingerprinting; device pose; RSSI



**Citation:** Khan, I.M.; Thompson, A.; Al-Hourani, A.; Sithamparanathan, K.; Rowe, W.S.T. RSSI and Device Pose Fusion for Fingerprinting-Based Indoor Smartphone Localization Systems. *Future Internet* **2023**, *15*, 220. <https://doi.org/10.3390/fi15060220>

Academic Editor: Paolo Bellavista

Received: 29 April 2023

Revised: 11 June 2023

Accepted: 12 June 2023

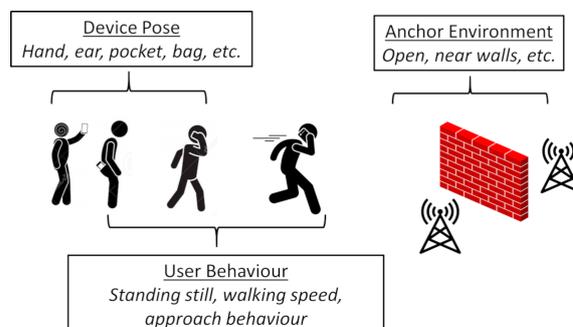
Published: 20 June 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

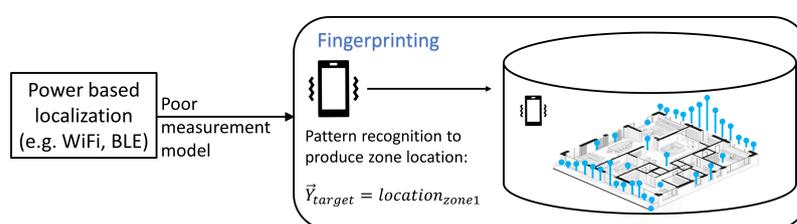
Modern smartphone-based indoor localization systems rely on the measurement of parameters, such as the received signal strength intensity (RSSI) of radio frequency (RF) signals, via spatially distributed anchors in the region of interest. These RF measurements in an indoor localization system can be influenced by at least three factors, namely, the phone environment (commonly known as “device pose” in the human activity recognition literature [1]), the user’s behavior, and the anchor environment, as shown in Figure 1. Each of these physical factors causes changes in the electromagnetic (EM) channel. For example, if the phone’s device pose is in a user’s pocket, the RF signals are likely to be far more attenuated due to shadowing in comparison to the device being located in a user’s hand.



**Figure 1.** Factors affecting RF measurements in localization systems.

Ultra wideband (UWB) signals that provide time of arrival/time difference of arrival (ToA/TDoA) measurements are less vulnerable to multipath and shadowing effects and,

therefore, have good measurement models (i.e., relationship between the measurement and the distance of the phone from the anchor). This makes UWB-based systems useful in calculating the point-estimate location of a phone. In contrast, WiFi and BLE signals that provide RSSI measurements are highly susceptible to these channel effects. Although some systems use indoor propagation models [2] or angle-of-arrival (AoA) measurements [3,4] to improve the resolution of distance from the RSSI, often the measurement models are too poor in real-world deployments to resolve to point estimates. A common approach in RSSI-based localization literature [5] is the use of a pattern-recognition style methodology known as RF-fingerprinting to perform coarse-grained localization (or zoning). RF fingerprinting involves conducting an initial measurement campaign in different zones within the area of interest to collect and store the RSSI data of the zones (“fingerprints”) in a database. Then, in an online phase, new RSSI measurements are compared with the fingerprints stored in the database, and the phone is localized to a zone, as depicted in Figure 2.



**Figure 2.** Power-based fingerprinting localization approach.

### 1.1. Research Contribution

Studying the fusion of smartphone sensor measurements with RF measurements is an important research direction to improve the accuracy of indoor localization. For systems that calculate the point location of a phone (such as UWB), well-known optimal sensor fusion algorithms such as Kalman Filtering can be easily used [6,7]. However, for fingerprinting-based localization, which is similar to pattern recognition, there is no apparent standard approach that can be employed.

Hence, the question arises: Is there an approach to use the onboard phone sensors to improve the accuracy of the underlying RF fingerprinting pattern recognition problem? To the best of the authors’ knowledge, no study has answered this question using the smartphone’s device pose. In this study, a novel conceptual framework was implemented using an end-to-end BLE setup that first uses accelerometer sensor measurements to identify the device pose of the smartphone and then combines the identified device pose with the RSSI measurements to locate the smartphone in the zone.

The original contributions of this study can be described as follows:

1. A conceptual data-level framework is proposed that combines a smartphone’s device pose obtained from accelerometer measurements with the smartphone’s BLE RSSI for indoor fingerprinting localization.
2. We employed cascaded neural networks to implement and evaluate the proposed framework’s model improvement (i.e., RSSI-only vs. RSSI-and-device-pose) via the neural network cross-entropy loss function improvement.
3. We applied the Kolmogorov–Smirnov statistical test to link the accuracy improvement seen in the neural network implementation to the data distribution.
4. We experimentally evaluated the localization accuracy improvement provided by the neural network implementation of the framework in an indoor environment.
5. We demonstrate the robustness of the conceptual framework by also implementing it using K-Nearest Neighbour and Naïve Bayes algorithms showing the localization accuracy improvement.

### 1.2. Scope and Outline

The novel framework proposed in this paper is a conceptual data-level framework that can be implemented using any pattern recognition algorithm. The implementation and

results of the framework using cascaded neural networks is described in detail, and results using K-Nearest-Neighbor and Naïve-Bayes are given to demonstrate that the framework has utility when implemented using other algorithms as well. Hence, this paper does not propose a new *algorithm*; we explore the novel combination of device pose data with RSSI data to produce an increase in localization accuracy over RSSI-only data, as this has not been explored in the literature. The selection or optimization of a pattern recognition algorithm to implement the framework was not within the scope of this study.

The remainder of this paper is structured as follows: Section 2 reviews related literature and details the rationale behind the proposed framework; Section 3 describes the proposed framework and its implementation using cascaded neural networks, and specifies the experimental setup. Section 4 presents detailed results and analysis of accuracy improvements; Section 5 discusses the advantages and disadvantages of the proposed framework and provides the conclusion to the paper.

## 2. Related Work and Motivation

Using inertial measurement unit (IMU) sensors on a smartphone or as a separate wearable for user-behavior detection has been copiously researched as a complementary data modality to improve indoor localization system accuracy [5], and most studies [8] have examined pedestrian dead reckoning (PDR) as the data fusion approach utilizing methods such as filtering [6,9,10]. However, smartphone-based PDR is also known to have major drawbacks, such as the accumulation of errors over time [11] and the impossibility of resolving the direction of travel without constraining the location or orientation of the smartphone (which is still an open research question [11,12]). The latter is an especially important consideration when looking at indoor localization systems because they are used in a diverse range of settings such as hospitals and shopping malls, in which users cannot be constrained to holding their phone in a particular manner. So, while smartphone-based PDR techniques to improve indoor localization accuracy may provide good results in studies, they may never be suitable for practical implementation.

The phone's device pose (i.e., how the phone is being carried/held by the user) changes the electromagnetic channel and affects the RSSI values at the anchors (even if the individual is not moving, as demonstrated in [13]) due to two easily identifiable reasons. Firstly, the phone may be placed in a bag or pocket or held at the ear, both of which cause varying magnitudes of greater attenuation due to shadowing compared with when the phone is held in an outstretched hand [14]. Secondly, being placed in a low carried handbag or being held at the ear can change the distance vector to an anchor and degrade performance. It should also be noted that these effects are compounded with distance. As described in a short experiment [15], a 10 dB attenuated RSSI at 10 cm away from an anchor produced about the same range estimate as unattenuated RSSI. However, at 1 m away from the anchor, a 10 dB attenuated RSSI could cause a difference in range estimate of 5–10 m compared with that of unattenuated RSSI. Thus, if during the fingerprinting measurement campaign, the range of RSSI values for different device poses at the zones are also collected, this could improve the stored fingerprint by accounting for the variations in the RSSI due to device pose in the zones.

There are only a handful of recent studies that have explored such data-level conceptual frameworks to use the smartphone sensors in order to improve the underlying RF fingerprinting pattern recognition problem. In [16], the researchers used the concept of different unique activities being carried out in different rooms of typical living quarters (e.g., lying on a bed in a bedroom, cooking in a kitchen, etc.) to improve the accuracy of a BLE RSSI fingerprinting localization system. Smartphone sensor data were collected from individuals using the different rooms for their activities, and this room-level localization was combined with RSSI fingerprinting localization using various classifiers. This resulted in slight accuracy improvements for some of the rooms and a decrease in accuracy in others. In ref. [13], a BLE beacon selection strategy was developed to remove nearby beacons because it was found that the RSSI from these is most profoundly affected by small

movements of the user’s hand; however, the authors did not explore utilizing the device pose to improve the localization accuracy.

Notably, in a recent study [17], the researchers used machine learning algorithms on accelerometer measurements to detect whether a user is moving (dynamic) or standing relatively still (static); thus, they proposed reliability metrics for the localization on RSSI values received while the user is static and dynamic. As such, the authors assessed the relationship between the user behavior component in Figure 1 and the RSSI localization in the experimental setup of five zones in a large room. Although the dataset in the study was collected for different device poses as well, the authors did not study the effect of these on the localization. Hence, there is a need to address this research gap to determine the localization improvement attained when taking device pose into account, which is the novel contribution of this study (as contextualized within the broader literature in Table 1).

**Table 1.** Overview of some frameworks in the literature for combining accelerometers and RSSI for localization.

Framework	Advantages	Disadvantages
Pedestrian dead reckoning (PDR) fusion with RSSI using standard data fusion algorithms (e.g., Kalman filter) [6,9,10]	<ul style="list-style-type: none"> <li>• Good accuracy improvement</li> <li>• Widely researched</li> </ul>	<ul style="list-style-type: none"> <li>• Accumulation of dead-reckoning error [11]</li> <li>• Direction of travel resolution impossible without constraints [11,12], making practical use difficult.</li> </ul>
Accelerometer-based activity recognition for: <ul style="list-style-type: none"> <li>• Improved room-level localization [16]</li> <li>• Anchor selection strategy [13] based on movement</li> <li>• Reliability metric on RSSI due to movement [17]</li> <li>• Improving RSSI localization with device pose information (this study)</li> </ul>	Practical due to absence of constraints on carrying smartphone	<ul style="list-style-type: none"> <li>• Low accuracy improvement, even reduction in accuracy (e.g., [16])</li> <li>• Requires additional data collection (e.g., static/dynamic [17]) for training.</li> </ul>

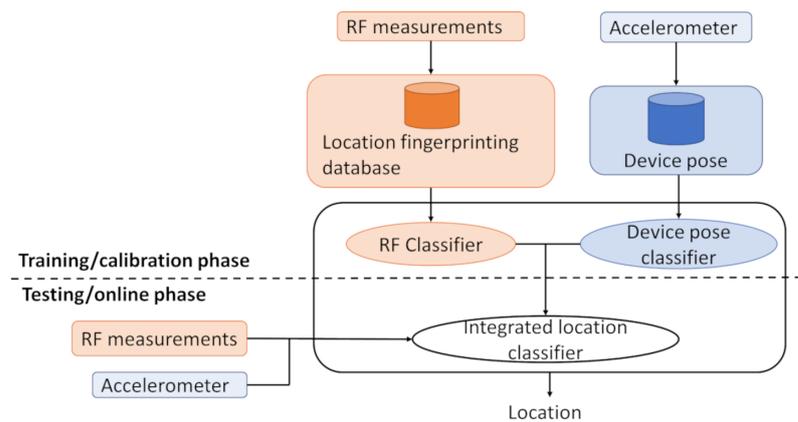
### 3. Methodology

This section describes the proposed data-level framework, its implementation using cascaded neural networks and provides details of the experimental setup, equipment, and data collection methodology.

#### 3.1. Proposed Framework

The proposed framework that was implemented in this study is an extension of typical RSSI fingerprinting localization, augmented with sensor data for phone device pose classification, as shown in Figure 3.

The accelerometer sensor was chosen to detect the smartphone device pose because it is the most common sensor used for the identification of human activity classes [11,12] and in sole device pose identification [18]. It is extremely accurate (>90%) for both applications; in refs. [19,20], the authors compared the use of different sensors individually and together for device-pose-only identification, and both studies concluded that using multiple sensors (together with, or aside from, the accelerometer) does not significantly increase accuracy compared with using the accelerometer alone.



**Figure 3.** Proposed framework overview.

During the training phase, RSSI measurements were taken at various locations in the region (such as zones on the floor of a building) and stored in a database. Additionally, during this training phase, phone accelerometer measurements were recorded for each use case under consideration. These data were then used to train classifiers in a supervised learning approach for the phone's device pose and the user's location. During the online/testing phase, the new RSSI measurements and phone accelerometer measurements were processed through an integrated trained classifier, and a location output class was determined.

### 3.2. Cascaded Neural Network Implementation of Proposed Framework

As mentioned earlier, the conceptual framework of this method, which uses device-pose identification with RSSI for localization, can be implemented using any pattern recognition algorithm. This subsection describes a cascaded neural network implementation of the proposed framework, which is one possible realization of the framework. The results of this cascaded neural network implementation are given in Sections 4.3 and 4.4. Results for cascaded K-Nearest Neighbor and Naïve Bayes implementations of the framework are also given in Section 4.5 to show the flexibility of the technique. It is stressed again that the proposed framework (i.e., classifying then combining the phone's device pose with the RSSI data) is general, and the optimization of the algorithmic technique to implement this framework (e.g., SVM, ensemble learning, CNN, etc.), which could possibly improve accuracy further, was not studied as it was outside the scope of this paper and would require an in-depth study in its own right.

Neural networks were chosen to implement and study the framework for two reasons:

- Because the RF-fingerprinting localization approach and phone's device pose identification are both fundamentally pattern recognition problems, machine learning classifiers and in particular neural networks were a prime choice for implementation due to their robustness and generalization capabilities with noisy and uncertain data. The trends in the literature also indicate their increased adoption for indoor localization [2].
- The universal approximation theorem [21] for neural networks implies that a single hidden layer neural network with enough neurons can approximate any hyperfunction to an arbitrarily high accuracy, thereby forming arbitrary decision boundaries in the classification input space. Because the proposed framework changes the input data vector for classification, this makes neural networks a good candidate to evaluate the effect.

The RSSI measurement from all  $N$  anchors at a particular time can be considered an  $N$ -dimensional vector  $\vec{r} = \langle r_1, r_2, \dots, r_N \rangle$ . Each RSSI measurement vector  $\vec{r}_i$  at the  $i$ th time corresponds to an element in the set of ground-truth zone labels  $Y = \{Y_1 = Zone1, \dots, Y_M = ZoneM\}$ , which relate to  $M$  zones of interest in the region's fingerprint map. Similarly, the 3-axis accelerometer measurement vector  $\vec{a}_i = \langle a_x, a_y, a_z \rangle$ , which is received at the  $i$ th time instance, corresponds to an element in the set of  $K$

ground-truth device pose labels  $X = \{X_1 = DevicePose1, \dots, X_K = DevicePoseK, \dots\}$ . The output vector  $\vec{O}_i = \langle P(X_1), P(X_2), \dots, P(X_K) \rangle$  of the device pose neural network denotes the probability of  $\vec{a}_i$  belonging to the  $k$ th device pose class, and the output vector  $\vec{L}_i = \langle P(Y_1), P(Y_2), \dots, P(Y_M) \rangle$  of the localization neural network denotes the probability of  $\vec{r}_i$  belonging to the  $m$ th location.

The difference between how an RSSI-only and an RSSI-and-device-pose localization neural network would perform zoning classification can be explained as follows:

- RSSI-and-device-pose: In order to combine the device pose and RSSI data, a neural network was trained using the accelerometer data to produce a device pose classification (neural network on the left in Figure 4), and we added this binary output classification vector as additional dimensions to the RSSI vector (thus,  $\vec{r}_i = \langle r_1, r_2, \dots, r_N, softargmax(\vec{O}_i) \rangle$ ). This projects the RSSI measurements from the different device poses onto orthogonal dimensions in the input space for the localization neural network. A second neural network (the neural network on the right in Figure 4) was then used on the combined RSSI-and-device-pose input vectors to produce the location output.
- RSSI-only: In contrast, an RSSI-only neural network was trained on  $\vec{r}_i$  to produce the location labels  $Y$  (only the neural on the right in Figure 4, with  $X_k$  removed in the input vector).

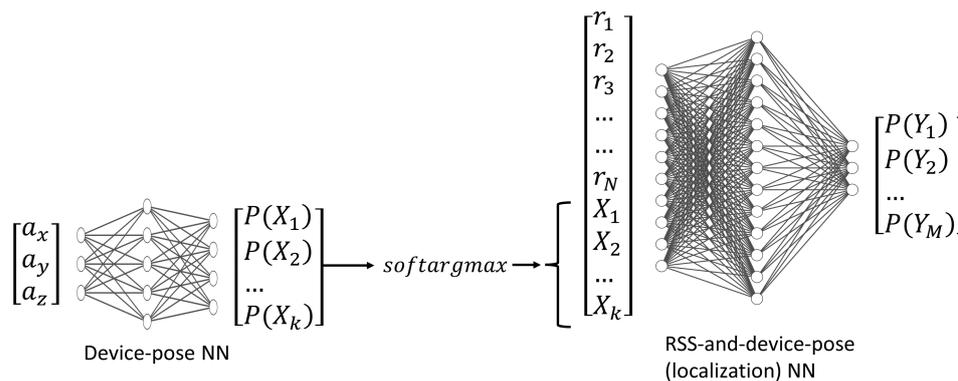


Figure 4. Cascaded neural network implementation of proposed framework.

### 3.3. Experimental Setup

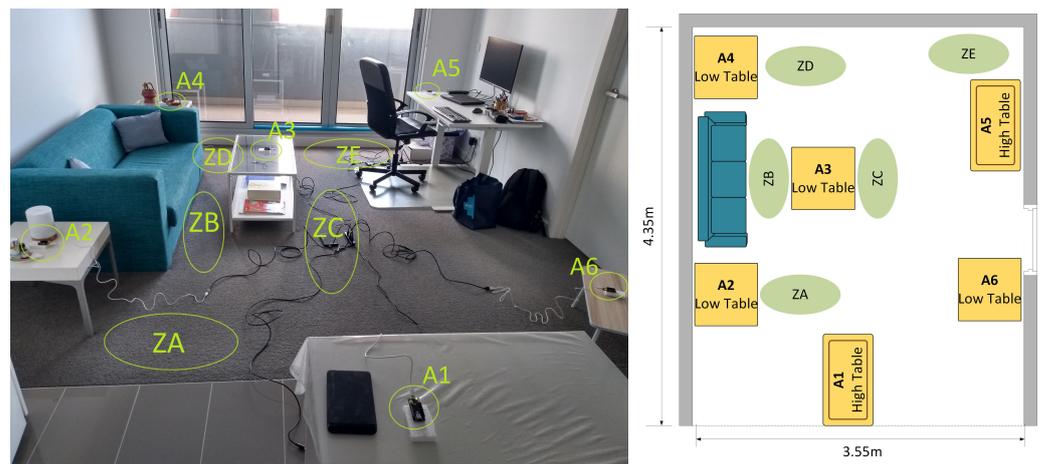
We used an experimental setup and protocol similar to that of [17], as shown in Table 2. Data were collected in a room-scale setup that nearly exactly replicated the experimental approach in [17], which allowed the accurate exploration of the research gap to determine the localization improvement attained when taking device pose into account.

Table 2. Details of the experimental setup.

	Setup in [17]	Setup in This Study
Number of zones	5	5
Number of anchors	5	6
Number of HAR device poses	4 (Hand, trouser pockets, jacket pocket, bag)	4 (AtEar, InHand, InPocket, InHandbag)
Movement type	Static and dynamic	Static and dynamic

RSSI and accelerometer measurements were collected at the five locations shown in Figure 5 (in the remainder of the paper, these are referred to either as ZoneA, ZoneB, ZoneC, ZoneD, and ZoneE, or ZA, ZB, ZC, ZD, and ZE, respectively) in an indoor environment with six spatially distributed anchors for four different device-pose use cases (shown in Figure 6, labelled AtEar, InHand, InHandbag, and InPocket, or AE, IH, IB, and IP, respectively). From Figure 5, it can be seen that each zone had a nearby anchor, and these could be written in

pairs as [ZA, A2], [ZB, A3], [ZC, A3], [ZD, A4], [ZE, A5]. For each location and each device pose, three data runs were recorded: one in which the device pose was kept completely static and two in which natural variations due to body movement were added to the device pose; these are referred to as dynamic1 and dynamic2. Larger and more frequent variations in the device-pose use cases were included while recording the dynamic2 data compared with those in the dynamic1 data (Figure 6 right subpanels). Approximately one minute of data was recorded for each static/dynamic1/dynamic2 file, resulting in a total of 60 min of data.



**Figure 5.** Picture of room and layout of anchors for data collection.



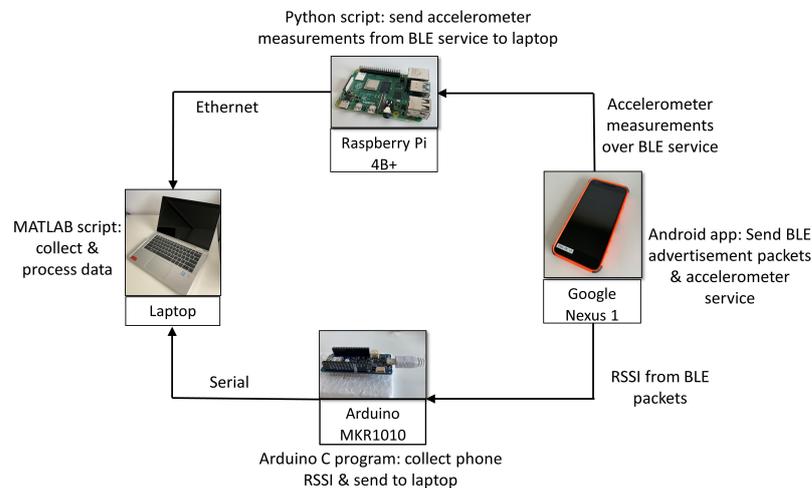
**Figure 6.** Device-pose use cases considered for experimental data collection.

Because the proposed framework depends on collecting a range of accelerometer and RSSI measurements while the phone is in a particular device pose, it was necessary for the accelerometer and RSSI transmitter to be on the same device. An Android smartphone application was developed to send BLE advertisements at 10 Hz (in practice, the RSSI was obtained at between 5 and 7 Hz due to missed packets) as well as a BLE Generic Attribute (GATT) server [22] to sample and send accelerometer measurements as BLE-notify messages (at around 15 Hz) to a GATT client. The two data modalities of interest were recorded as follows:

1. RSSI data were obtained from the BLE advertisements of a Google Nexus 5X phone with Arduino MKR1010 microcontrollers employed as anchors. The standard ArduinoBLE library was used to obtain the phone's BLE advertisement packet RSSI, which was then sent over a USB serial connection to an HP EliteBook x360 (Intel i5 quadcore with 8GB RAM) laptop running MATLAB 2022Rb.
2. Accelerometer data were obtained from the BLE-notify messages of a Google Nexus 5X phone with a GATT client programmed on a RaspberryPi 4B+, which then sent the accelerometer measurements over UDP to the laptop. The RaspberryPi was used to offload the processing for receiving the higher sampling rate BLE-notify messages for

the accelerometer service. The RaspberryPi's clock was time-synced with the laptop with Network Time Protocol to ensure synchronized timestamps.

The experimental measurement setup is summarized in Figure 7 with the hardware and software components and their roles.



**Figure 7.** Experimental setup for data collection.

To prepare the data into vectors for the classification, the following preprocessing steps were applied:

1. A 1-element buffer was used to hold the RSSI vector. This buffer was filled before being considered the first complete RSSI vector. Therefore, every anchor returned at least one measurement from the beginning of the recording of the data file for the first valid RSSI vector to be recorded.
2. After the first complete RSSI vector was recorded, as described above, every new RSSI value from an anchor was added to the 1-element buffer and the contents of the buffer recorded as a new RSSI vector. This simulated the behavior of systems that attempt a localization step every time a new RSSI measurement is received from an anchor.
3. Because the accelerometer sampling rate was higher than the RSSI sampling rate, the accelerometer measurement closest in timestamp to the latest timestamp of the RSSI measurements in the buffer was recorded to be the accelerometer measurement corresponding to that RSSI vector.

A total of 57,489 vectors (approximately 11,000 per location after minor resampling) were obtained as a dataset in this manner.

#### 4. Results and Analysis

This section presents the distribution of the collected dataset. The Kolmogorov–Smirnov test statistic was used to quantitatively expand upon the hypothesis of the proposed framework and predict the likely zone classes from the dataset where the highest accuracy improvement would be expected. The later subsections provide results of the cascaded neural network cross-entropy for different hyperparameters and give the accuracy improvement results for the RSSI-and-device-pose framework.

##### 4.1. RSSI Data Distribution

The distributions of the RSSI measurements of the anchors for the different use cases at each location are given in Figure 8 for the static files (i.e., user stationary in the particular device pose use case). In order to quantitatively explore the effect of device pose on the RSSI, the two-sample two-sided Kolmogorov–Smirnov (KS) statistic was calculated for all

locations, device pose use cases, and static/dynamic files. The two-sample KS statistic is a nonparametric statistic defined by:

$$D_{AB} = \max_x (|F_A - F_B|) \quad (1)$$

where  $F_A$  and  $F_B$  are the empirical cumulative distribution functions (ECDFs) of two different sample sets. The KS test was used because it is a nonparametric test used to identify whether two sample sets come from the same distribution [23]. The KS test statistic's p-values for the RSSI of Anchor1 for all locations and all device-pose use cases for the static file are shown in Figure 9 (because  $D_{AB} = D_{BA}$  in Equation (1), only the lower triangular entries of the symmetric matrix are relevant).

The KS test null hypothesis ( $H_0$ ) is that the samples come from the same distribution. The instances (cells) where the KS test's  $H_0$  would fail to be rejected (using the typical  $p$ -value  $\geq 0.05$ ) are indicated with red font and outlined with a border, and represent KS test type II errors. The KS test type II errors represent a failure of true negative (TN) classification (i.e., true negative is the rejection of  $H_0$  when  $H_0$  is false). The KS test statistic p-values were calculated in this manner for all anchors for all locations and device poses and for both the static and dynamic files.

In Table 3, these instances of the KS test type II errors are grouped by location, and the total number of type II errors between the locations for the anchors are given. Several trends can be seen in these RSSI distributions as well as from the KS statistics:

1. Anchors closest to a location tended to have the highest RSSI: As mentioned in Section 3.3, each zone had a nearby anchor. The trend of an elevated median RSSI value at that zone for its nearest anchor is clearly depicted in Figure 8, where Anchor 2, Anchor 3, Anchor 4, and Anchor 5 have high median values for nearly every use case for their corresponding closest positions.
2. RSSI variation due to device pose that resulted in distributions from different locations seemed alike: This trend encapsulates the premise of the proposed framework. For example, referring to column ZA\_AE and row ZB\_IH in Figure 9, the RSSI distribution for Anchor 1 in the static dataset while standing in the ZA zone and holding device AtEar leads to the null hypothesis acceptance of the KS test, implying that it is from the same RSSI distribution as the ZB zone while the device is held InHand (i.e., a KS test type II error for the distribution conditioned on location). Thus, based on the RSSI of Anchor 1 alone, it would be difficult to distinguish the ZA location from the ZB location while the phone is in those device poses. This would reduce the localization accuracy of any pattern recognition approach wherever Anchor 1's distribution emerges as an important feature for classifying the ZA and ZB locations. Importantly, as Table 3 shows, the KS test has type II errors more often for RSSI distributions grouped by different locations (which would affect localization accuracy) than for distributions grouped in the same location (which would not affect localization accuracy).
3. Most KS test Type II errors involving ZB: From Table 3, looking at the zones with the three highest numbers of KS test type II errors (bold and underlined in the right-most column of the table), these are for ZB and ZD, and ZA and ZB. Therefore, given the hypothesis of this study, it could be predicted that adding device pose information likely results in the highest improved TN rates for zone ZB.

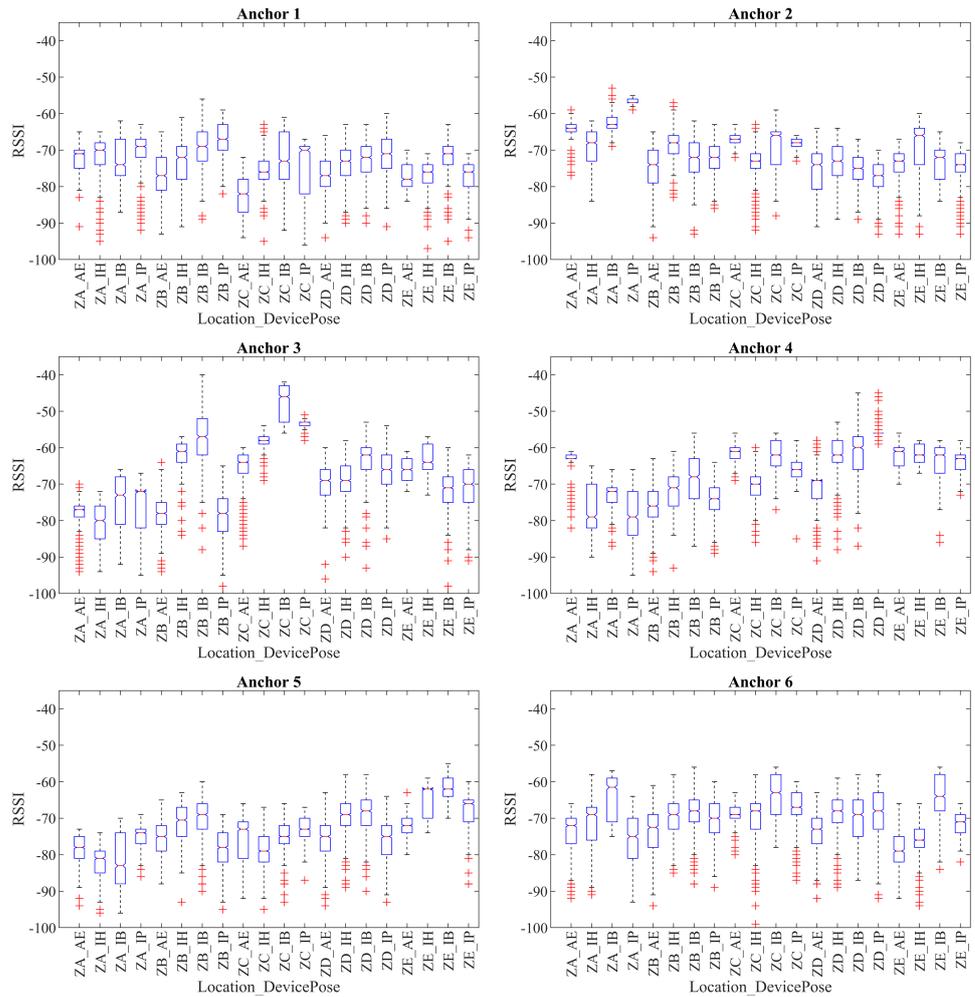


Figure 8. RSSI-boxplots of the 6 anchors for each location and each device pose (static files).

	ZA_AE	ZA_IH	ZA_IB	ZA_IP	ZB_AE	ZB_IH	ZB_IB	ZB_IP	ZC_AE	ZC_IH	ZC_IB	ZC_IP	ZD_AE	ZD_IH	ZD_IB	ZD_IP	ZE_AE	ZE_IH	ZE_IB	ZE_IP
ZA_AE	1.0E+00																			
ZA_IH	2.4E-01	1.0E+00																		
ZA_IB	2.8E-02	1.6E-05	1.0E+00																	
ZA_IP	1.3E-02	3.8E-06	2.6E-09	1.0E+00																
ZB_AE	3.1E-06	1.4E-15	1.7E-08	1.2E-22	1.0E+00															
ZB_IH	1.1E-01	6.5E-05	2.6E-03	2.8E-09	2.1E-05	1.0E+00														
ZB_IB	7.3E-04	1.3E-09	2.4E-07	2.7E-01	5.9E-17	4.1E-06	1.0E+00													
ZB_IP	2.2E-10	2.1E-23	9.1E-18	2.1E-09	9.2E-42	8.4E-22	1.7E-06	1.0E+00												
ZC_AE	4.5E-21	9.5E-48	3.1E-29	1.9E-52	1.2E-11	1.1E-29	3.7E-55	1.6E-81	1.0E+00											
ZC_IH	1.7E-07	2.3E-20	1.6E-08	5.9E-29	1.2E-05	4.7E-06	9.5E-23	1.1E-51	2.8E-33	1.0E+00										
ZC_IB	9.2E-04	1.2E-07	9.1E-05	1.1E-12	5.8E-08	6.3E-05	4.8E-08	4.7E-28	5.9E-43	4.2E-09	1.0E+00									
ZC_IP	5.3E-03	2.2E-03	1.4E-05	5.4E-09	8.9E-14	7.9E-04	4.7E-15	2.0E-41	5.3E-38	5.7E-17	6.4E-10	1.0E+00								
ZD_AE	1.8E-05	2.5E-13	8.0E-08	1.2E-19	8.0E-01	2.6E-03	1.8E-15	1.8E-37	1.2E-13	8.7E-04	9.9E-07	2.1E-12	1.0E+00							
ZD_IH	4.9E-01	1.5E-03	7.1E-04	5.2E-07	4.2E-06	9.7E-01	3.1E-06	3.4E-22	4.5E-32	2.6E-06	8.7E-06	2.6E-03	3.1E-04	1.0E+00						
ZD_IB	3.4E-01	7.1E-03	2.5E-02	6.7E-06	2.1E-06	9.5E-01	1.8E-04	8.0E-17	3.5E-30	1.8E-05	9.0E-05	2.0E-03	7.5E-04	6.8E-01	1.0E+00					
ZD_IP	3.6E-01	1.8E-03	3.8E-05	4.0E-02	7.8E-14	5.3E-04	9.9E-03	3.2E-11	2.0E-51	4.0E-18	8.1E-06	8.0E-09	8.1E-12	3.2E-03	1.9E-02	1.0E+00				
ZE_AE	9.1E-08	5.3E-17	6.6E-09	9.0E-23	1.0E-01	1.5E-05	8.9E-20	9.4E-41	2.5E-11	7.6E-02	2.9E-06	1.1E-16	3.0E-01	4.2E-06	1.6E-06	1.2E-14	1.0E+00			
ZE_IH	2.7E-13	2.3E-32	1.4E-13	6.3E-42	4.8E-04	3.1E-13	1.1E-34	4.3E-67	5.3E-30	1.1E-03	1.2E-19	1.0E-28	4.5E-03	1.9E-13	4.2E-12	1.1E-29	3.8E-02	1.0E+00		
ZE_IB	6.6E-01	7.5E-03	3.3E-05	1.7E-05	2.9E-14	2.4E-03	3.5E-07	1.7E-23	1.1E-54	9.3E-20	8.9E-07	9.4E-06	4.3E-11	3.0E-02	1.2E-02	1.5E-02	1.6E-11	1.3E-31	1.0E+00	
ZE_IP	3.0E-15	3.6E-35	1.4E-13	5.5E-44	5.8E-04	1.1E-15	2.8E-38	1.3E-66	1.6E-14	4.0E-06	4.8E-23	1.2E-30	2.5E-04	6.9E-16	1.7E-13	6.3E-33	6.5E-03	6.8E-03	3.4E-35	1.0E+00

Figure 9. Pairwise Kolmogorov–Smirnov statistic  $p$ -values for Anchor 1 (static file).

**Table 3.** KS test type II errors per anchor grouped by location.

Static/Dynamic	Zone	Anchor 1	Anchor 2	Anchor 3	Anchor 4	Anchor 5	Anchor 6	Total
Static	A and B	2	1	0	0	1	2	6
	A and C	0	0	0	0	1	0	1
	A and D	3	0	0	0	1	2	6
	A and E	1	0	0	0	0	1	2
	B and C	0	1	0	0	2	1	4
	B and D	3	3	1	1	3	9	<u>20</u>
	B and E	1	3	0	0	0	0	4
	C and D	0	0	0	1	2	2	5
	C and E	1	1	0	0	0	1	3
	D and E	1	4	1	0	0	0	6
Same location	2	3	3	2	1	4	15	
Total		14	16	5	4	11	22	
Dynamic1	A and B	9	2	3	9	6	4	<u>33</u>
	A and C	2	2	1	1	4	2	12
	A and D	6	0	1	0	6	2	15
	A and E	1	0	2	0	3	5	11
	B and C	3	4	2	4	2	4	19
	B and D	8	1	2	1	4	6	<u>22</u>
	B and E	1	2	0	3	4	2	12
	C and D	1	1	2	3	1	4	12
	C and E	4	2	2	5	1	1	15
	D and E	1	4	3	1	3	2	14
Same location	6	4	3	10	5	4	33	
Total		42	22	21	37	39	36	
Dynamic2	A and B	0	0	1	2	0	1	4
	A and C	4	3	0	0	4	3	14
	A and D	1	0	0	0	2	2	5
	A and E	1	0	3	0	2	1	7
	B and C	1	0	0	0	4	4	9
	B and D	0	2	0	0	0	1	3
	B and E	0	0	1	2	1	0	4
	C and D	0	0	0	0	0	1	1
	C and E	3	2	1	8	2	1	17
	D and E	1	1	0	0	0	2	4
Same location	2	4	3	9	3	6	27	
Total		13	12	9	21	17	22	

#### 4.2. Device Pose Neural Network

The details of the algorithmic study and the results for the antecedent device pose neural network of the RSSI-and-device-pose neural network framework (Figure 4) that utilizes the accelerometer measurements to detect device pose can be found in [24]. This device pose identification network was sized with just five hidden neurons and achieved good accuracy (>90%) for each device pose class. The network's high accuracy is not surprising, as datasets for much more complex activities and even for multiple individuals are also commonly classified at >90% accuracy rates in the human activity detection literature using accelerometers [18].

#### 4.3. RSSI-Only vs. RSSI-and-Device-Pose Neural Network Cross-Entropy Results

In this subsection, the results and analysis for a sweep of the hyperparameters of the RSSI-only and RSSI-and-device-pose neural networks are provided. Due to the low dimensionality of the data (6 dimensions of anchors in the case of the RSSI-only localization network and a maximum of 10 dimensions for the 6 anchors and 4 device poses in the case of the RSSI-and-device-pose localization network), a single hidden layer was used

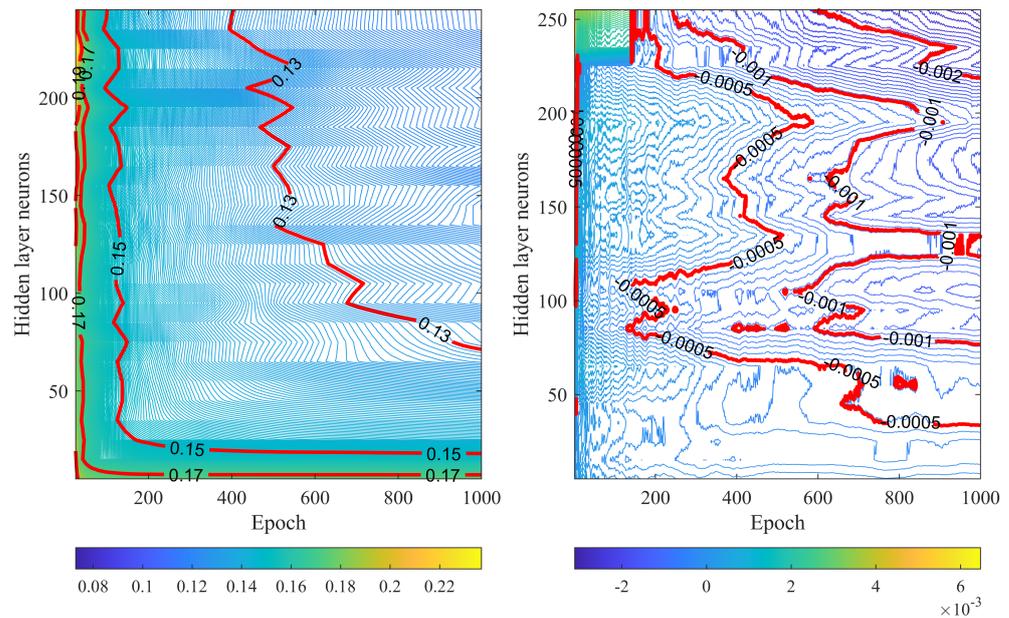
for both the RSSI-only and the RSSI-and-device-pose neural network. The entire collected dataset (57,489 vectors) was randomly split into 80% for training and 20% for validation. Using the training-only and validation-only data, a sweep of the number of hidden neurons (5 to 250 hidden neurons in 5 neuron increments) for both the RSSI-only and the RSSI-and-device-pose networks was conducted for 1000 training epochs. Both neural networks were presented with the same training and validation vector dataset and trained using the standard backpropagation algorithm. In general, during training, the measurement vectors from the training set  $V^{train} = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_w\}$ , whose ground truths were  $\gamma^{train}$ , were sequentially applied to the neural network whose weights were randomly initialized. The pre-softmax output vector of the neural network  $\vec{O}$  was then used to calculate the average cross-entropy ( $Loss_{CE\_TR}$ ) of the classification (the  $Loss_{CE\_VAL}$  can be calculated with the same equation using the validation dataset instead):

$$Loss_{CE\_TR} = \frac{\sum_{s=1}^w Loss_s}{|V^{train}|} = - \frac{\sum_{s=1}^w \bar{T}_s^{train} \cdot \ln(\vec{O}_s)}{|V^{train}|} = - \frac{\sum_{s=1}^w \sum_{j=1}^k T_j^{train} \ln(P(O_j))}{|V^{train}|} \quad (2)$$

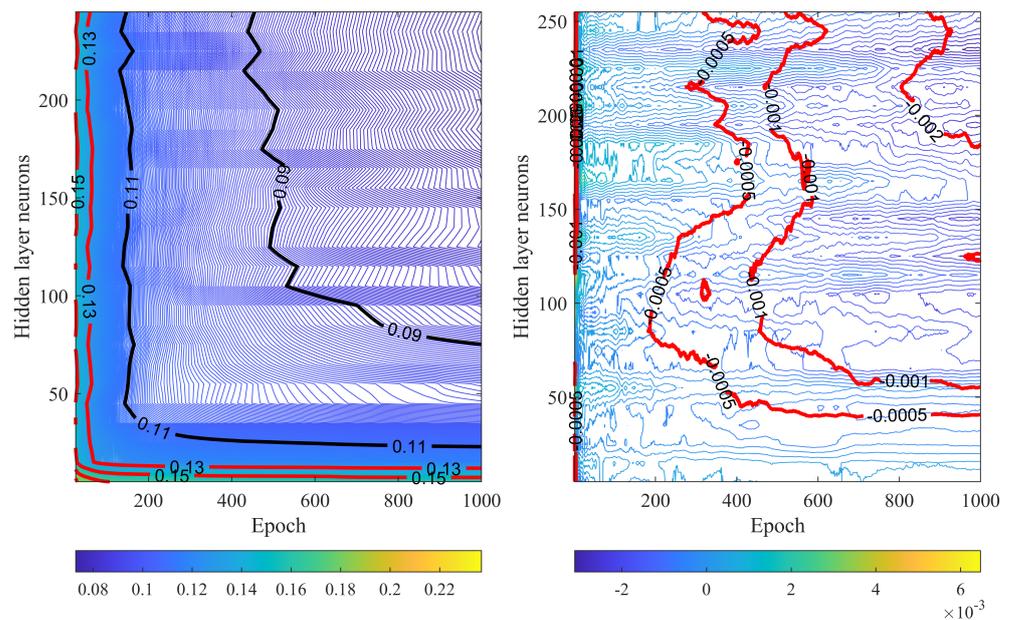
As the natural logarithm function has a range of  $(-\infty, 0]$  over  $[0, 1]$ , the higher the probability for a given class  $P(O_j)$ , the lower the cross-entropy value. Generally, the lower the average cross-entropy value, the better the model learned, though overfitting may occur. When overfitting happens, the training loss function  $Loss_{CE\_TR}$  continues to decrease; however, the validation loss function  $Loss_{CE\_VAL}$  starts to increase as the neural network overlearns the training vectors and loses generalization ability to correctly classify the validation vectors. One way to recognize overfitting is to evaluate the disparity between the training and validation performance ( $Loss_{CE\_TR} - Loss_{CE\_VAL} < 0$ ).

The  $Loss_{CE\_TR}$  and ( $Loss_{CE\_TR} - Loss_{CE\_VAL}$ ) for both the RSSI-only and the RSSI-and-device-pose neural networks are shown in Figures 10 and 11 (note, the  $Loss_{CE\_TR}$  is plotted for epochs >20 due to scale, and the  $Loss_{CE\_TR} - Loss_{CE\_VAL}$  is plotted as a five-sample moving average to smoothen the discontinuities for the hidden neuron sweep). From these training parametric runs, it was observed that the RSSI-and-device-pose localization network always achieved a lower  $Loss_{CE\_TR}$  at earlier epochs than the RSSI-only localization network and with fewer neurons. In fact, the RSSI-and-device-pose localization network achieved  $Loss_{CE\_TR}$  values of 0.11 and 0.09 (black contours in Figure 11), which the RSSI-only localization network was unable to reach within the range of hidden neurons and epochs. This was despite the higher dimensionality of input data, implying that a better localization model was learnt by the RSSI-and-device-pose localization network.

Additionally, as shown in Figures 10 and 11, the RSSI-and-device-pose network reached an overfitting point at earlier training epochs than the RSSI-only network (and with a lower  $Loss_{CE\_TR}$ ), implying that it reached an optimal training state at earlier epochs. This also points toward a better localization model being learnt more quickly (in epochs) by the RSSI-and-device-pose localization network, resulting from the addition of the device pose information.



**Figure 10.** RSSI-only neural network’s  $Loss_{CE\_TR}$  (left) and 5-sample moving average  $Loss_{CE\_TR} - Loss_{CE\_VAL}$  (right). Note:  $Loss_{CE\_TR}$  is shown for epochs > 20 due to scale.

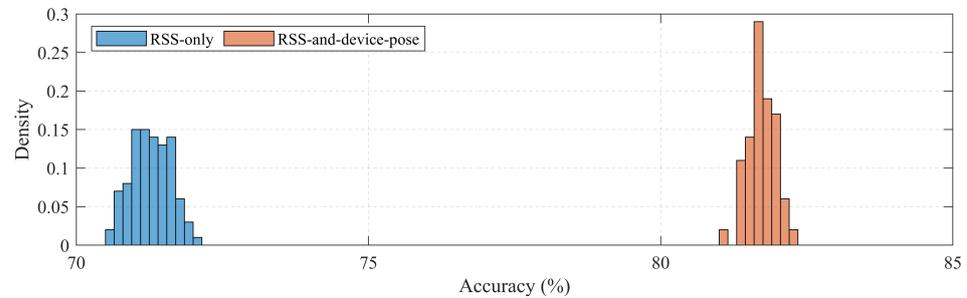


**Figure 11.** RSSI-and-device-pose neural network’s  $Loss_{CE\_TR}$  (left) and 5-sample moving average  $Loss_{CE\_TR} - Loss_{CE\_VAL}$  (right). Note:  $Loss_{CE\_TR}$  is shown for epochs > 20 due to scale.

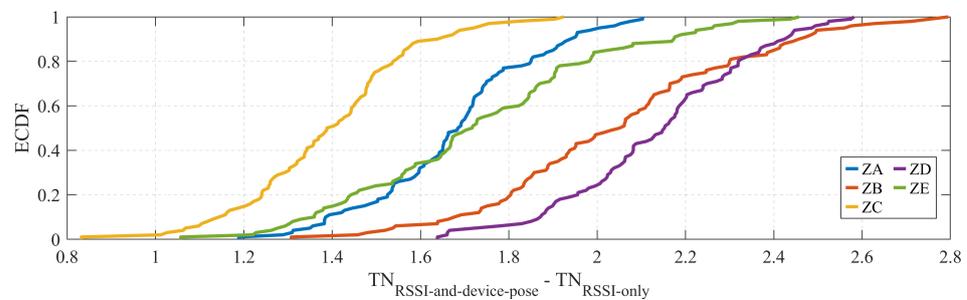
4.4. RSSI-Only vs. RSSI-and-Device-Pose Localization Results (NN)

Both the RSSI-only and RSSI-and-device-pose neural networks were equally sized (125 neurons), and both were trained for the number of epochs where the RSSI-only network achieved a cross-entropy loss goal of 0.15 (from Figure 10, this occurred at 136 epochs). Both neural networks were trained and tested 100 times, with the entire collected dataset (57,489 vectors) being randomly split into 70% training and 30% testing for each run. Both neural networks were presented with the same training and testing vector dataset (with device-pose information appended for the RSSI-and-device-pose neural network vectors from the device-pose identification neural network, refer to Figure 4), during each of the 100 runs. The total accuracy for both networks was taken over these 100 randomized runs and is given in Figure 12. In Figure 13, the percentage point improvement in true-negative

(TN) rates is depicted as an empirical cumulative density function (ECDF) over all the runs. For each zone class, the TN improvement was obtained by subtracting the rate of the RSSI-only neural network from the rate of the RSSI-and-device-pose neural network (i.e.,  $TN_{RSSI\text{-and-device-pose}} - TN_{RSSI\text{-Only}}$ ).



**Figure 12.** Normalized accuracy histogram for the RSSI-only and RSSI-and-device-pose neural networks.



**Figure 13.** Percentage-point improvement in TN rate between RSSI-and-device-pose neural network and RSSI-only neural networks for each zone.

From Figure 12 (also summarized in Table 4), the overall accuracy of the RSSI-only neural network was quite high (min–max range: 70.5–72.3%), but when device pose data was added, a consistent accuracy increase of between 8.8 and 11.8 percentage points was achieved (average improvement: 10.72 percentage points). The spread of the normalized histogram is also slightly narrower, implying better precision. Interestingly, the TN rate improvement (Figure 13) shows that the ZB zone had the largest percentage point improvements. This matches the expectations from the KS test results in Section 4.1 where the ZB zone had the highest number of type II errors.

**Table 4.** Accuracy and processing time for 3 different supervised learning algorithm implementations of the proposed framework.

Framework	Algorithm	Accuracy (%) Average [Min,Max]	Training Time (s) Average [Min,Max]	Testing Time (s) Average [Min,Max]
RSSI-Only	KNN	74.3 [73.8, 74.9]	0.043 [0.038, 0.260]	0.745 [0.657, 0.888]
	NB	66.2 [65.5, 66.7]	0.024 [0.019, 0.301]	0.014 [0.011, 0.046]
	NN	71.01 [70.5, 72.3]	520.75 [509.82, 532.25]	2.179 [2.052, 2.736]
RSSI-and-device-pose	KNN	76.6 [76.2, 77.1]	0.078 [0.071, 0.164]	1.360 [1.080, 1.702]
	NB	68.7 [67.9, 69.3]	0.126 [0.110, 0.395]	0.022 [0.021, 0.023]
	NN	81.73 [81.1, 82.3]	544.71 [539.56, 557.30]	2.945 [2.74, 4.69]

4.5. KNN, NB and NN Localization Results for Proposed Framework

To demonstrate the robustness of the framework, it was also implemented on the K-nearest neighbor (KNN) and Naïve Bayes (NB) supervised learning algorithms and trained and tested for 100 randomized runs. The hyperparameters of the algorithms were optimized using MATLAB’s hyperparameter optimizer for KNN and NB. The results of

the RSSI-only and RSSI-and-device-pose accuracy percentages are shown in Table 4, which also summarizes the neural network (NN) results given earlier. The results in Table 4 indicate that the RSSI-and-device-pose framework produced a consistent accuracy increase regardless of the algorithm implemented. The accuracy increase produced by the RSSI-and-device-pose framework was lower when implemented with KNN and NB. The lower rates of improvement for KNN and NB could be due to the following reasons:

- KNN relies on the Euclidean distance, so its performance typically degrades in higher-dimensional spaces (i.e., the curse of dimensionality). As depicted in Figure 4, when the device pose was added to the RSSI vector, the input vector increased in dimensionality. Thus, for KNN, the RSSI-and-device-pose performance could be expected to be lower than that of RSSI-only. However, we found that the performance was slightly higher, indicating that useful information for a better model existed in the combined RSSI-and-device-pose input vectors.
- The naïve assumption of the NB algorithm presumes that the components of the input vector are independent. However, this is almost certainly untrue because any movement (e.g., in the dynamic files) affects the correlation between the anchor RSSI readings; and, importantly, the device pose affects the RSSI. It is possible that because the naïve assumption does not hold for the additional features in the RSSI-and-device-pose vector, the accuracy improvement is low.

The processing time for the 100 randomized runs for both the training and testing stages of the algorithms was measured using MATLAB's internal timing functions and the results are given in Table 4. The results demonstrate an increased computational cost associated with the RSSI-and-device-pose framework but to various degrees for the different algorithms. The KNN implementation for the RSSI-and-device-pose framework required, on average, around double the computational time for training and testing compared with the RSSI-only framework; NB required, on average, five times longer to train but was nearly the same time during the testing phase. For both frameworks, the KNN testing time was longer than the training time, a theoretical peculiarity of KNN [25]. As expected, the neural network implementation required the longest to train due to the backpropagation algorithm, which was iterated over the training epochs, but the average increase for the RSSI-and-device-pose framework was only 4.6%. The average difference in inference time for the testing dataset between the NN framework implementations was less than 1 s.

## 5. Discussion

In smartphone-based localization systems, RSSI can vary due to the device pose of the smartphone. In this study, a novel data level framework was developed and explored using cascaded neural networks to detect the device pose via accelerometer measurements, which we then combined with the RSSI vector, resulting in consistent improvements in localization accuracy, with an average increase of 10.7%.

The experimental study, results, and analysis of this paper support the proposed framework in four ways:

1. The KS test statistics indicate that the device pose could contribute toward the RSSI distributions for the anchors having higher type II errors between zones (and hence a lower true negative rate for a class). This allowed for the prediction of zone ZB to be the zone that would result in the greatest TN rate improvement after the addition of device-pose information, which was the case in the neural network implementation of the framework.
2. The results of the hyperparameter sweep of the neural networks indicates that RSSI-and-device-pose achieved a lower  $Loss_{CE\_TR}$  with fewer neurons and with less epochs, indicating that a better model was learned when the RSSI vector was augmented with device-pose information
3. The results of the 100 randomized runs of the RSSI-only and RSSI-and-device-pose neural networks indicate that the RSSI-and-device-pose consistently had higher clas-

sification accuracy. It should be noted that both these neural networks were equally sized and trained for the same number of epochs with the same training and testing vector dataset (with device-pose information added for the RSSI-and-device-pose neural network).

4. With the framework implemented using other classification algorithms such as kNN and naïve Bayes, a consistent increase in accuracy was also seen but at lower rates.

*Crucially, in all the experimental runs in this study, the proposed framework not only had a higher average localization accuracy: it **always** had a higher localization accuracy than any other RSSI-only localization run* (as shown by the entire accuracy histograms shifting to the right in Figure 12 and the min–max range in Table 4). These results indicate that combining the RSSI with the device pose is likely to provide consistent increases in localization accuracy for fingerprinting systems.

The significance of this accuracy improvement should also be considered in the context of the entire system:

1. As mentioned in Section 4.2, the device-pose identification neural network is very small (five neurons). In the cascaded neural network runs on the collected dataset, the average difference in training time was 4.6% between the frameworks, and the average difference in inference time was less than 1 s. Thus, the accuracy improvement achieved came at a very low computational cost and memory overhead given the right selection of algorithm.
2. In fingerprinting localization systems, RSSI-based zone classification is not carried out on the smartphone, as it does not hold the RSSI fingerprint database, but rather on a central system that holds the fingerprint database. If a method such as pedestrian dead reckoning (PDR) is to be used for improving the zoning accuracy, this means that either accelerometer or PDR (heading and velocity) measurements must be communicated to the central system, or the central system must communicate the zone classification to the smartphone for it to combine with the PDR data. In either case, some sort of PDR location/zoning information must be conveyed over a wireless channel that may be intercepted by a malicious actor, posing a significant privacy risk. However, for the proposed framework, it is possible to conduct device-pose identification on the smartphone using a small neural network that has been calibrated for the individual and to send only a standardized device-pose class to the central system. In this case, no zoning or PDR location data are communicated, avoiding the risk of being intercepted.

In conclusion, the proposed framework provides consistent accuracy improvement across the board; it also comes at a low computational and memory cost and has low overall privacy risk. The accuracy improvement vs. timing performance trade-off illustrated by the results in Table 4 shows that studying the optimization of classification algorithms for the implementation of the framework (which, as mentioned in Section 1.2, was beyond scope of this study) is a good direction for further research. Developing the framework on a system with wireless communication protocols that enables more expansive deployment of anchors (e.g., in a multistorey building) is also an avenue for future work.

**Author Contributions:** I.M.K.: Conceptualization, Methodology, Software, Investigation, and Writing; A.T.: Conceptualization and Reviewing; A.A.-H.: Methodology and Reviewing; K.S.: Reviewing; W.S.T.R.: Conceptualization and Reviewing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study was supported by the Australian Government, Department of Industry, Innovation and Science grant AEGP000053 and the Australian Government Research Training Program scheme.

**Data Availability Statement:** Data are available upon reasonable request from I.M.K.

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

## References

1. Kasebzadeh, P.; Radnosrati, K.; Hendeby, G.; Gustafsson, F. Joint pedestrian motion state and device pose classification. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 5862–5874. [[CrossRef](#)]
2. Guo, X.; Ansari, N.; Hu, F.; Shao, Y.; Elikplim, N.R.; Li, L. A survey on fusion-based indoor positioning. *IEEE Commun. Surv. Tutor.* **2019**, *22*, 566–594. [[CrossRef](#)]
3. He, S.; Long, H.; Zhang, W. Multi-antenna array-based aoa estimation using bluetooth low energy for indoor positioning. In Proceedings of the 2021 7th International Conference on Computer and Communications (ICCC), Chengdu, China, 10–13 December 2021; pp. 2160–2164.
4. Fascista, A.; Coluccia, A.; Ricci, G. A Pseudo Maximum likelihood approach to position estimation in dynamic multipath environments. *Signal Process.* **2021**, *181*, 107907. [[CrossRef](#)]
5. Zafari, F.; Gkelias, A.; Leung, K.K. A survey of indoor localization systems and technologies. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2568–2599. [[CrossRef](#)]
6. Feng, D.; Wang, C.; He, C.; Zhuang, Y.; Xia, X.G. Kalman-filter-based integration of IMU and UWB for high-accuracy indoor positioning and navigation. *IEEE Internet Things J.* **2020**, *7*, 3133–3146. [[CrossRef](#)]
7. Ali, R.; Liu, R.; Nayyar, A.; Qureshi, B.; Cao, Z. Tightly Coupling Fusion of UWB Ranging and IMU Pedestrian Dead Reckoning for Indoor Localization. *IEEE Access* **2021**, *9*, 164206–164222. [[CrossRef](#)]
8. Kozłowski, M.; Santos-Rodríguez, R.; Piechocki, R. Sensor modalities and fusion for robust indoor localisation. *EAI Endorsed Trans. Ambient. Syst.* **2019**, *6*, e5. [[CrossRef](#)]
9. Zhou, B.; Elbadry, M.; Gao, R.; Ye, F. Towards scalable indoor map construction and refinement using acoustics on smartphones. *IEEE Trans. Mob. Comput.* **2019**, *19*, 217–230. [[CrossRef](#)]
10. Mahfouz, S.; Mourad-Chehade, F.; Honeine, P.; Farah, J.; Snoussi, H. Target tracking using machine learning and Kalman filter in wireless sensor networks. *IEEE Sens. J.* **2014**, *14*, 3715–3725. [[CrossRef](#)]
11. Ashraf, I.; Hur, S.; Park, Y. Smartphone Sensor Based Indoor Positioning: Current Status, Opportunities, and Future Challenges. *Electronics* **2020**, *9*, 891. [[CrossRef](#)]
12. Davidson, P.; Piché, R. A survey of selected indoor positioning methods for smartphones. *IEEE Commun. Surv. Tutor.* **2016**, *19*, 1347–1370. [[CrossRef](#)]
13. Ng, P.C.; Spachos, P.; She, J.; Plataniotis, K. A Kernel Method to Nonlinear Location Estimation with RSS-based Fingerprint. *IEEE Trans. Mob. Comput.* **2022**, *Early Access*. [[CrossRef](#)]
14. Mamun, M.A.A.; Anaya, D.V.; Wu, F.; Yuce, M.R. Landmark-Assisted Compensation of User’s Body Shadowing on RSSI for Improved Indoor Localisation with Chest-Mounted Wearable Device. *Sensors* **2021**, *21*, 5405. [[CrossRef](#)] [[PubMed](#)]
15. Faragher, R.; Harle, R. An analysis of the accuracy of bluetooth low energy for indoor positioning applications. In Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014), Tampa, FL, USA, 8–12 September 2014; pp. 201–210.
16. Tsanousa, A.; Xefferis, V.R.; Meditskos, G.; Vrochidis, S.; Kompatsiaris, I. Combining rssi and accelerometer features for room-level localization. *Sensors* **2021**, *21*, 2723. [[CrossRef](#)] [[PubMed](#)]
17. Filus, K.; Nowak, S.; Domańska, J.; Duda, J. Cost-effective filtering of unreliable proximity detection results based on BLE RSSI and IMU readings using smartphones. *Sci. Rep.* **2022**, *12*, 2440. [[CrossRef](#)] [[PubMed](#)]
18. Motani, K.; Wong, K.; Kamijo, S. Classifying Human Activity and Smartphone Holding Mode Using Accelerometer and Gyroscope. In Proceedings of the 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 15–18 October 2019; pp. 11–12.
19. Guiry, J.J.; Karr, C.J.; van de Ven, P.; Nelson, J.; Begale, M. A single vs. multi-sensor approach to enhanced detection of smartphone placement. In Proceedings of the 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Chicago, IL, USA, 26–30 August 2014; pp. 3691–3694.
20. Shoaib, M.; Bosch, S.; Incel, O.D.; Scholten, H.; Havinga, P.J. Fusion of smartphone motion sensors for physical activity recognition. *Sensors* **2014**, *14*, 10146–10176. [[CrossRef](#)] [[PubMed](#)]
21. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **1989**, *2*, 303–314. [[CrossRef](#)]
22. Bluetooth SIG. *Bluetooth Core Specification v5.1*; Bluetooth SIG: Kirkland, DC, USA, 2019.
23. Press, W.H.; Teukolsky, S.A. Kolmogorov-Smirnov Test for Two-Dimensional Data: How to tell whether a set of (x, y) data points are consistent with a particular probability distribution, or with another data set. *Comput. Phys.* **1988**, *2*, 74–77. [[CrossRef](#)]
24. Khan, I.M.; Sun, S.; Rowe, W.S.; Thompson, A.; Al-Hourani, A.; Sithamparanathan, K. Comparison of classifiers for use case detection using onboard smartphone sensors. In Proceedings of the 2022 32nd International Telecommunication Networks and Applications Conference (ITNAC), Wellington, New Zealand, 30 November–2 December 2022; pp. 261–266.
25. Manning, C.D. *An Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2009; p. 299.

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