

Robust Phase-based BLE Localization with a Single Multi-Antenna Receiver and Machine Learning

Georgios Andreadis*, Panos N. Alevizos# and Aggelos Bletsas*

*WINLAB, Dept. of Electrical and Computer Engineering, Rutgers University, The State Univ. of New Jersey

#Renesas Design, Athens 17674, Greece

{geoandrs, aggelos}@winlab.rutgers.edu, panos.alevizos.wm@renesas.com

Abstract—This study presents a neural network-based method for static Bluetooth Low Energy (BLE) localization, using phase measurements and carrier frequency per data packet, captured from a single (or multiple) multi-antenna receiver(s). Deterministic phase-based techniques fail to accurately estimate the position of a tag using a single multi-antenna receiver with closely spaced antennas. This is due to the random and unknown carrier phase offset (CPO) at each antenna element, introduced during BLE’s inherent frequency hopping between consecutive packets. Our long short-term memory (LSTM) model learns to handle the unknown distribution of CPO and suppress phase noise, enabling robust localization with just one multi-antenna receiver. Among the tested architectures, the LSTM model showed notable resilience to phase noise and achieved higher accuracy during high-rate tag transmissions compared to the feedforward (FF) and convolutional neural network (CNN) models. All models were trained on simulated or real data and tested on real data from various environments, where they outperformed deterministic techniques—even in cases where those techniques either excelled or failed to estimate the tag’s position.

I. INTRODUCTION

Phase-based indoor localization has gained significant interest within the wireless research community due to its importance in applications like asset tracking, navigation, and location-based services. These applications demand high accuracy and low power consumption, making Bluetooth low energy (BLE) a popular choice.

Phase measurements [1] inherently introduce ambiguity in distance estimation, as distances that differ by an integer amount of wavelength offer the same phase, for one-way communication. This ambiguity can be mitigated using multiple wavelengths or increased bandwidth [2].

Several studies have utilized channel state information (CSI) [3], [4] and direction of arrival (DoA) [5] methods, leveraging the 80 MHz bandwidth of BLE to enhance localization. Trilateration methods based on RSSI are also common, as seen in [6] and [7]; the latter employs least square estimation, three-border and centroid methods. Fingerprinting methods are another alternative (e.g. in [8]), while neural network-based approaches, including [9] and [10], have also been explored. In [9], LSTMs are applied with RSSI measurements from different beacons; [10] addresses multipath fading using machine learning with a multi-carrier phase-based (MCPD) approach introducing a model containing a convolutional neural network

*This work was supported by NSF project 2433991. The measurements can be found in <https://github.com/geoandrs/ble-dataset.git>

(VggNet) [11] and deep neural networks to classify line-of-sight (LOS) and non-line-of-sight (NLOS) environments.

This work proposes a LSTM neural network (NN) architecture, which offers BLE tag localization where deterministic techniques completely fail. The reason is that the specific NN architecture “learns” and mitigates the unknown and random CPO that appears at each antenna element, due to carrier frequency hopping between consecutive packets, inherent in BLE operation. LSTM outperforms FF and CNN-based NNs, while the NN-based approach is utilized with a single multi-antenna BLE locator, in contrast to prior art.

Sec. II offers the system model; Sec. III offers a deterministic DoA-based localization technique, for baseline comparison; Sec. IV describes datasets and design of the NN models; Sec. V provides the numerical results, and finally, work is concluded in Sec. VI.

II. SYSTEM MODEL

BLE tag location is denoted as $\mathbf{x}_T \triangleq [x_{tag} \ y_{tag} \ z_{tag}]^T$ and similarly, BLE locator location as \mathbf{x}_R . The BLE tag transmits a packet followed by a continuous wave, coined as Constant Tone Extension (CTE), across various carrier frequencies $f_c^{(i)}$, alongside an unknown, random initial phase ϕ_0 . The received signal at the locator experiences a time delay of $\tau = d_0/c = d_0/(\lambda^{(i)} f_c^{(i)})$, where $d_0 \triangleq \|\mathbf{x}_R - \mathbf{x}_T\|_2$ represents the Euclidean distance between the locator and the tag, c is the speed of light and $\lambda^{(i)}$ is the wavelength corresponding to each carrier. In the absence of multipath interference and with only line-of-sight (LoS) propagation, this corresponds to a signal received at the locator with phase equal to $\phi_0 - 2\pi f_c^{(i)} \tau = \phi_0 - 2\pi d_0/\lambda^{(i)}$.

Accounting for multipath and flat-fading channel propagation between the tag and the locator, the phase of the one-way propagation channel h (i.e., from tag to locator) is given by:

$$\phi_{\text{prop}} \equiv \angle h = -\frac{2\pi d_0}{\lambda^{(i)}} + \angle h_m, \quad (1)$$

where $\angle h_m$ represents the phase introduced by multipath effects. Additionally, there are fixed phase shifts due to cabling (ϕ_c) and phase noise $\tilde{\phi}_n$ at the locator’s receive chain:

$$\begin{aligned} \phi_R &= \phi_0 + \phi_{\text{prop}} + \phi_c + \tilde{\phi}_n, \\ &= -\frac{2\pi d_0}{\lambda^{(i)}} + \underbrace{\phi_0 + \phi_c}_{\theta} + \underbrace{\angle h_m + \tilde{\phi}_n}_{\phi_n}, \end{aligned}$$

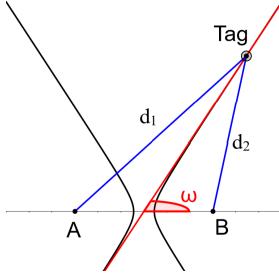


Fig. 1: Antennas and a tag result in a hyperbola and its approximation.

$$\Rightarrow \phi_R = -\frac{2\pi d_0}{\lambda^{(i)}} + \theta + \phi_n. \quad (2)$$

Receivers typically report a value in $[-\pi, \pi]$ or $[0, 2\pi)$. The locator in the conducted experiment utilize the latter, and thus,

$$\phi_{\text{meas}} = \phi_R \bmod 2\pi, \quad (3)$$

$$\begin{aligned} &= \left[-\frac{2\pi d_0}{\lambda^{(i)}} \bmod 2\pi + \underbrace{\theta \bmod 2\pi}_{\hat{\Theta}} + \underbrace{\phi_n \bmod 2\pi}_{\hat{\Phi}_n} \right] \bmod 2\pi, \\ &= \left[-\frac{2\pi d_0}{\lambda^{(i)}} \bmod 2\pi + \hat{\Theta} + \hat{\Phi}_n \right] \bmod 2\pi. \end{aligned} \quad (4)$$

III. DETERMINISTIC LOCALIZATION METHOD

For baseline reference and comparison to the neural network approach, a deterministic method was also implemented, provided below for completeness.

A. Multiple Locators

The phase of the signal received by antenna A and B is denoted as ϕ_A and ϕ_B , respectively (Fig. 1). Although phase measurements carry inherent ambiguity, recent findings show that when BLE antennas are positioned close enough (less than λ_{\min} , i.e., the smallest BLE carrier wavelength), the phase difference can uniquely determine a constant distance difference [12], [13], which in turn, defines a specific hyperbola:

$$d_1 - d_2 = \begin{cases} \frac{\phi_A - \phi_B}{2\pi} \lambda^{(i)}, & d_1 > d_2 \text{ and } \phi_A > \phi_B, \\ \frac{\phi_A - \phi_B}{2\pi} \lambda^{(i)} + \lambda^{(i)}, & d_1 > d_2 \text{ and } \phi_A < \phi_B, \\ \frac{\phi_A - \phi_B}{2\pi} \lambda^{(i)} - \lambda^{(i)}, & d_1 < d_2 \text{ and } \phi_A > \phi_B, \\ \frac{\phi_A - \phi_B}{2\pi} \lambda^{(i)}, & d_1 < d_2 \text{ and } \phi_A < \phi_B. \end{cases} \quad (5)$$

The asymptotic line of this hyperbola provides a good estimation of the signal's 2D direction-of-arrival (DoA). It can be shown that by applying Eq. (5) to the two axes in the

¹($\alpha + \beta$) mod γ = [$(\alpha \bmod \gamma) + (\beta \bmod \gamma)$] mod γ .

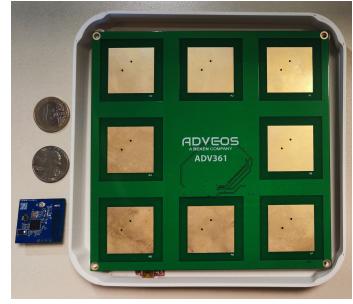


Fig. 2: BLE tag and 8-antenna locator/receiver.

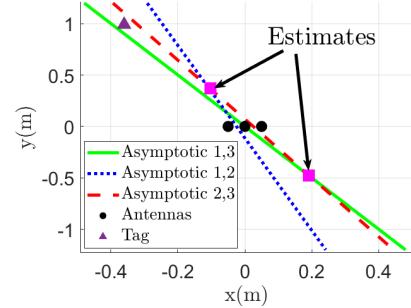


Fig. 3: Using the deterministic (hyperbolic) localization method with the antennas of a single locator, the position of the BLE tag cannot be estimated.

plane containing the locator, the azimuth (ϕ) and elevation (θ) can be estimated; thus, from each locator, a 3D DoA can be determined. With multiple locators, the tag's 3D position can be estimated using the following equation [12]:

$$\mathbf{p} = \left(\sum_i (\mathbb{I} - \mathbf{n}_i \mathbf{n}_i^T) \right)^{-1} \sum_i (\mathbb{I} - \mathbf{n}_i \mathbf{n}_i^T) \mathbf{s}_i, \quad (6)$$

where \mathbf{n}_i is the unit direction vector of the i -th 3D DoA line, which is calculated from the estimated azimuth and elevation based on the phase difference measurements from the i -th locator, \mathbf{s}_i is the origin of the i -th 3D DoA line, and \mathbb{I} is the identity matrix.

B. One Locator

Two approaches were tested. The first approach exploits the geometry of the locator (Fig. 2); the 2D position of the tag was estimated by intersecting the 2D DoA lines (based on the hyperbola method above) from each pair of antennas parallel to the xy -plane. The method is highly sensitive to noise, and the antenna pairs of the locator are positioned very close to each other, so the intersection points of the DoA lines cannot accurately estimate the tag's position, as shown in Fig. 3.

The second approach involves the previously estimated accurate 3D DoA, and then attempting to estimate the distance between the tag and the locator. Assuming the use of a single antenna at the locator and that two consecutive packets occur at frequencies i and $i + w$, the distance can be estimated as follows [14], [15]:

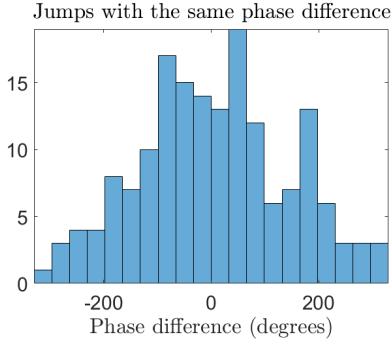


Fig. 4: Histogram of CPO when frequency jumps from $f_i = 2.446$ GHz to $f_j = 2.436$ GHz at the first antenna element (reference antenna).

$$d = \begin{cases} \frac{c \Delta\phi}{2\pi w f_h}, & w \leq w^* \text{ and } \Delta\phi > 0 \\ \frac{c \Delta\phi}{2\pi w f_h} - \frac{c}{w f_h}, & w \leq w^* \text{ and } \Delta\phi < 0 \\ \frac{c \Delta\phi}{2\pi w f_h} - k \frac{c}{w f_h}, & w > w^*, k \in \mathbb{Z}_*^- = \{-1, -2, \dots\} \end{cases} \quad (7)$$

where $f_h = 2$ MHz is the frequency spacing of two consecutive BLE channels, c is the speed of light, $\Delta\phi$ is the received phase difference from the same antenna, $w^* = \left\lfloor \frac{c}{f_h d_{max}} \right\rfloor$, d_{max} is the maximum allowed distance of the application, and k is an ambiguity parameter.²

The above estimation technique could not be applied to the current equipment because the BLE tag transmits each packet with an unknown random phase ϕ_0 that varies from packet to packet. A possible reason for this issue is the frequency hopping in the BLE tag, which results in an additive unknown phase for each packet. When measuring phase at the same antenna, with the locator and tag stationary at a predetermined distance, a constant phase difference was expected for a specific frequency hop. However, it was found that the measured phase from a particular hop, i.e., from f_i to f_j , is random (Fig. 4). Additionally, measurements across different hop pairs resulted in distributions that could not be approximated by any known models. Moreover, the distribution of the CPO cannot be reliably estimated, as it also depends on the environment and the number of measurements.

Other deterministic techniques could not be applied in this setup. The BLE locator uses the Bluetooth 5.1 protocol, which does not provide a timestamp indicating when the tag transmitted the packet. Additionally, since one-way communication is used, time-of-flight information is unavailable. Finally, the BLE locator did not provide reliable RSSI measurements—values remained nearly constant throughout the space—making techniques such as fingerprinting and RSSI-based distance estimation infeasible.

²When $w > w^*$, the measurement can be neglected due to the k ambiguity.

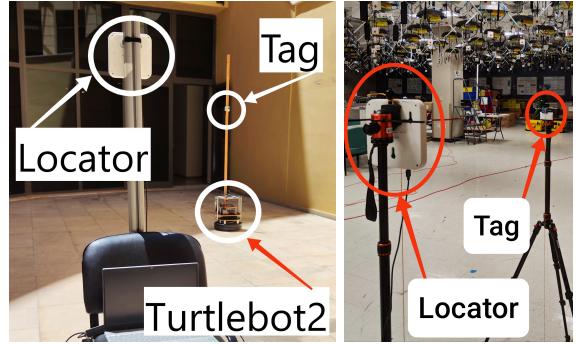


Fig. 5: Setup to collect measurements with 1 locator for the *One Locator Dataset* (left) and the *Orbit Dataset* (right).

IV. NEURAL NETWORK LOCALIZATION METHOD & ARCHITECTURE

Neural networks (NN) are employed to estimate the tag's position using only one locator, which is the case where the deterministic technique failed; NNs are tested with the hope to capture the unknown distribution of the CPO and filter the CPO noise that challenges any phase-based deterministic localization method, especially when the locator antennas are closely spaced. Three different NN architectures are tested, namely feedforward (FF), long short-term memory (LSTM) and convolutional neural networks (CNN), described below.

A. Data Collection and Pre-processing

One BLE tag was mounted on a Turtlebot2 platform, equipped with a Hokuyo UST-20LX light detection and ranging (LiDAR) sensor, able to perform simultaneous self-localization and mapping (SLAM) for precise collection of BLE measurements at specific grid positions, while one AD-VEOS ADV361 8-element BLE 5.1 locator (Fig. 2) was mounted at a stationary point (Fig. 5 (left)). The robot moved in 20 cm increments along the x - and y -axis, mapping a 5×5 m 2 grid with the tag at a fixed height of 1.47 m and collecting approximately 500 phase measurements at each of the 635 positions of the tag (625 in the grid points and 10 randomly placed inside the grid). As shown in Fig. 5, the data was collected in a large indoor area without a ceiling (i.e., it can be considered an outdoor area) to maximize grid space and minimize multipath interference (*One Locator Dataset*).

A second dataset was created using a single locator in a complex, waveguided indoor environment within the Orbit room at WINLAB (*Orbit Dataset*), as shown in Fig. 5 (right). For this dataset, the tag was placed at 81 different positions on a 4×4 m 2 grid, with a fixed height of 1.47 m, collecting approximately 500 phase measurements at each position. This dataset was used for additional testing of the NNs that had already been trained and tested on the *One Locator Dataset*.

Simulation was also conducted to collect measurements from random tag positions within a 5×5 m 2 grid with the tag at a fixed height of 1.47 m, in order to create a new dataset (*Simulation Dataset*). A Rician channel between the tag and the locator was simulated using Rician K-factor of $K_{dB} = 5$, with

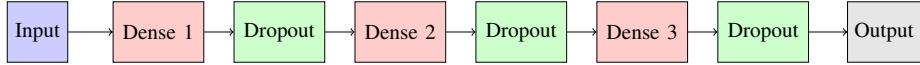


Fig. 6: FF Model Architecture.

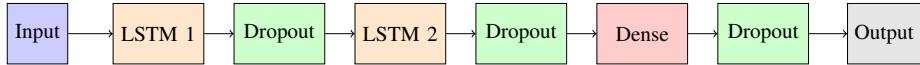


Fig. 7: LSTM Model Architecture.

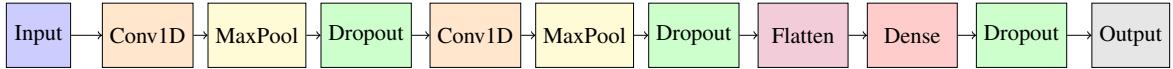


Fig. 8: CNN Model Architecture.

$\sigma_{\text{dB}} = 0$ to represent line-of-sight and multipath conditions. A total of 500 measurements were collected from 10^5 unique tag positions. The SNR in dB, denoted as SNR_{dB} , was randomly chosen for each position from a uniform distribution in the range [15, 30] dB; additive white Gaussian noise (AWGN) was also applied to each measurement according to the average received signal power and the desired SNR.

A third indoor dataset was created using four locators positioned at the corners of a room and rotated to converge on a central point (*Four Locators Dataset*). Measurements covered a set of 68 positions on a $9 \times 5 \text{ m}^2$ 2D grid with a tag at a fixed height of 1.3 m. The number of positions was lower but the number of measurements per position was notably higher, in the order of 6500 measurements per position.

As a pre-processing step, carrier frequency offset (CFO) correction is essential, as the CFO (multiplied by time) introduces a frequency shift that adds an additional phase offset unrelated to the actual position of the tag. A coarse correction is applied using the method in [16], while fine correction is omitted due to spacing constraints. By correcting the CFO, the phase data depend only on the tag's position and the unknown CPO, simplifying the NN task to focus on filtering the CPO and improving the localization accuracy.

B. FeedForward (FF) NN Architecture

The first architecture (Fig. 6) is a feedforward model designed to handle independent feature inputs, relying solely on dense layers for learning. Its input consists of the 9 features the locator provides (8 phase measurements from the locator's 8 antennas and the frequency of each packet). The model includes 3 dense layers with 128 units and a tanh activation function. Each dense layer is followed by a dropout layer with a rate of 0.5 to reduce overfitting by randomly dropping nodes. The model's output is the coordinates of the tag.

C. Long Short-Term Memory (LSTM) NN Architecture

The second architecture (Fig. 7) is an LSTM model specifically designed to capture sequential dependencies in time series data. The collected measurements can be viewed as sequential data since they were taken from the same position. The 500 measurements are structured into small time series, based on a parameter called `group_size`, which indicates the number of consecutive measurements in the sequence, using

non-overlapping windows (i.e., without sliding). Each measurement consists of the 9 features, as described above in the FF architecture. Thus, the model's input is shaped as a matrix of dimension `group_size` \times 9, allowing it to recognize patterns over time. The model features two LSTM layers, each with 128 units. The first LSTM layer outputs new features from each 9-feature measurement, allowing the model to capture detailed temporal information across the sequence. The second LSTM layer then condenses this information by outputting only a single value, summarizing the entire sequence into a single representation for the following dense layer. Each LSTM layer is followed by a dropout layer with a rate of 0.5 to mitigate overfitting. After processing through these layers, the model includes a dense layer with 128 units and a tanh activation function, followed by another dropout layer with a rate of 0.5 for regularization, before concluding with an output layer that predicts the coordinates of the tag.

D. Convolutional Neural Network (CNN) Architecture

The last architecture (Fig. 8) is a CNN model designed for sequence-based tasks. Its input has the same structure as the LSTM model. The architecture consists of two convolutional blocks, each featuring a 1D convolution layer with 128 filters and a kernel size of 3, followed by an 1D max pooling layer with a pool size of 2 to reduce the temporal dimension. To prevent overfitting, a dropout layer with a rate of 0.5 is applied after each pooling operation to mitigate overfitting. The extracted features are then flattened and passed through a fully connected dense layer with 128 units and a tanh activation function, followed by another dropout layer with a rate of 0.5 for regularization. The final output layer consists of three neurons, predicting the coordinates of the tag.

For all architectures a second, similar architecture was created by modifying their input layer to include 12 features (the original 9 features plus the 3D coordinates of the locators).

The tested `group_size` was set to 10, as this is the maximum number of measurements that can be captured in real time without the tag moving. Although results with larger values (e.g., 20) show improved performance, we omit a full ablation study due to space limits. A value of 10 offers the best trade-off between accuracy and real-time feasibility.

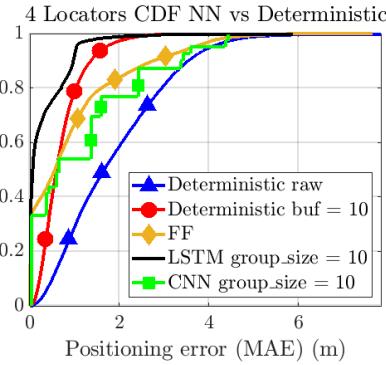
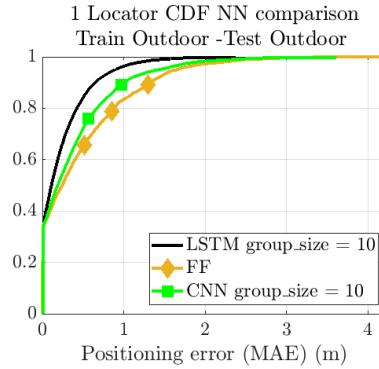
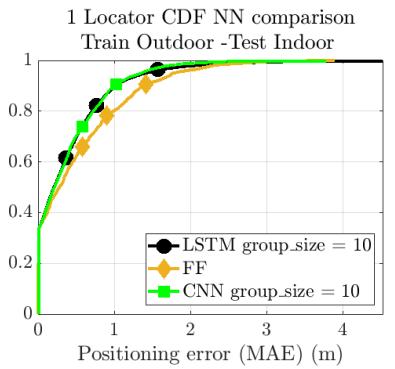


Fig. 9: Comparison of the three different architectures with their deterministic counterparts.



(a) NN trained and tested on different subsets of the same dataset.



(b) NN trained on one dataset and tested on another.

Fig. 10: NN performance in different testing environments.

The models were compiled using the Adam optimizer with a learning rate of 0.01, and the loss function was set to the mean absolute error, with mean squared error included as an additional evaluation metric. An early stopping callback was employed to mitigate overfitting and enhance generalization. This callback monitors the validation loss and halts training if no improvement is observed for 10 consecutive epochs, restoring the model weights corresponding to the best performance. Furthermore, a learning rate scheduler was used to reduce the learning rate by a factor of 0.2 if the validation loss plateaued for 5 epochs, with a minimum threshold of 0.0001. The models were trained for a maximum of 100 epochs using scaled (standardized) input features, a batch size of 64, and batch-level shuffling to improve convergence. Validation data was used throughout training to monitor the model's performance.

V. NUMERICAL RESULTS

Raw and buffered measurements were used to assess errors in the deterministic techniques. Buffering is a smoothing technique, typically used to reduce noise power and improve estimation accuracy. Each locator's DoA is first estimated and saved in a buffer; then, averaging across the buffer measurements yields the final DoA estimate.

The three NN architectures were tested on the datasets described above. The FF neural network was compared with the deterministic technique that uses raw measurements. In contrast, the LSTM and the CNN neural networks were compared with the deterministic technique that uses a buffer size equal to the group_size parameter. In the NN architectures, the datasets were split in 70%, 10%, 20% parts for training, validation and testing, respectively.

A. Four Locators Dataset

The models were initially tested on a dataset where the results of the deterministic techniques are known to be very accurate for the specific room.

For multiple locators, the second version of the networks is utilized to include the 3D coordinates of the locators to match the information used by the deterministic method. While

the FF and LSTM networks outperform their deterministic counterparts, the CNN struggles to learn the dependencies between measurements (Fig. 9), likely because they come from locators in different positions in the room, and the convolution operation cannot effectively capture this spatial variation.

B. One Locator Datasets

The challenge for the models is to estimate the tag's position using information from only one locator, as no closed-form deterministic technique currently exists to solve this problem. Successfully addressing this challenge would reduce the number of locators required for accurate tag positioning.

First, the networks are trained on a subset of the *One Locator Dataset* and tested on a different, unseen, subset of the same dataset. As shown in Fig. 9a, the LSTMs can learn the distribution of the CPO and filter out the noise of the measurements from the trained environment better than the FF and the CNN. Then, the models are tested on the *Orbit Dataset*, where the dependencies are completely different because the measurements were collected in a complex indoor environment. As shown in Fig. 9b, the LSTMs and the CNN achieve similar errors, both performing better than the FF. An interesting observation is that the CNN's error remains consistent when moving from one environment to another, making it more robust than the LSTM.

In the case of using a single locator, LSTM can be considered a better choice because it achieves better results in the environment it is trained on, and performs comparably to CNNs in unseen environments (Fig. 10). The training time for both models (LSTM and CNN) is similar when using real data (i.e., approximately 2-5 minutes) on a single Tesla T4 GPU. However, CNNs can be more easily parallelized, which can lead to faster training when using simulated data - since there is much more simulated data than real data. Due to the LSTM's superior performance in the trained environment and its comparable results in other environments, LSTM was chosen for further testing.

Training the LSTM model with simulated data and testing it with real data does not yield results as accurate as training

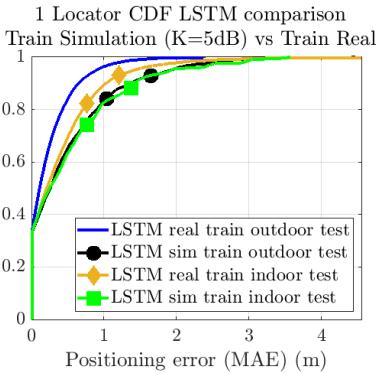


Fig. 11: Train the model on the *Simulation Dataset* or on a subset of *One Locator Dataset* and test on real data: the *Orbit Dataset* and the subset of the unseen measurements of the *One Locator Dataset*.

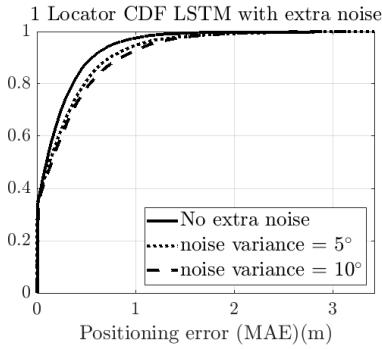


Fig. 12: By adding extra Gaussian noise with varying standard deviation from $\sqrt{5}^\circ$ to $\sqrt{10}^\circ$, the model can learn to filter out the noise from the data and estimate the tag's position nearly as accurately as when using data without added noise.

and testing with real data only, as shown in Fig. 11; that also suggests that the model not only filters noise but also captures dependencies not represented in simulations.

Gaussian noise with a mean of 0° and adjustable standard deviation from $\sqrt{5}^\circ$ to $\sqrt{10}^\circ$ was added to the training, real (experimental) phase measurements, testing the model's robustness to noise, with real data. As shown in Fig. 12, the model adapt to the additional noise and achieve comparable results. Various types of noise (uniform, deltas) and combinations of these were also applied, confirming that the primary function of these models is to filter out noise and the random phase introduced by the tag's frequency hopping.

VI. CONCLUSION

The use of neural networks for localizing BLE tags can yield remarkable results, surpassing those of deterministic techniques. These networks capture the unknown distribution of the CPO, filter associated noise, and estimate the tag's position using a single locator — an approach that challenges phase-based deterministic localization methods, especially when locator antennas are closely spaced, limiting aperture surface

and thus, antenna resolution. This capability enables neural networks to excel in scenarios where deterministic techniques fail. Additionally, the models demonstrate robustness against additive, hard-to-model, per antenna phase noise, observed experimentally during BLE frequency hopping. In cases of high-rate tag transmission with numerous measurements available, forming input data sequences, the preferred NN architecture is the LSTM model, which outperformed CNNs and FF NNs, achieving the best results across all scenarios.

REFERENCES

- [1] P. V. Nikitin, R. Martinez, S. Ramamurthy, H. Leland, G. Spiess, and K. V. S. Rao, "Phase based spatial identification of UHF RFID tags," in *Proc. IEEE Int. Conf. on RFID*, Orlando, USA, Apr. 2010, pp. 102–109.
- [2] D. Vasicht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Santa Clara, USA, Mar. 2016, pp. 165–178.
- [3] C. S. Mouhammad, A. Allam, M. Abdel-Raouf, E. Shenouda, and M. Elsabrouty, "BLE indoor localization based on improved RSSI and trilateration," in *7th Int. Japan-Africa Conf. on Electronics, Communications, and Computations (JAC-ECC)*, 2019, pp. 17–21.
- [4] R. Ayyalasomayajula, D. Vasith, and D. Bharadia, "BLoc: CSI-based accurate localization for BLE tags," in *Proc. of the 14th Int. Conf. on Emerging Networking EXperiments and Technologies*, Heraklion, Greece, Dec. 2018, pp. 126–138.
- [5] X. Qiu, B. Wang, J. Wang, and Y. Shen, "AOA-based BLE localization with carrier frequency offset mitigation," in *IEEE Int. Conf. on Communications Workshops (ICC Workshops)*, 2020, pp. 1–5.
- [6] B. S. Goh, A. K. Mahamad, S. Saon, K. Isa, H. A. Ameen, M. A. Ahmadon, and S. Yamaguchi, "IoT Based Indoor Locating System (ILS) using Bluetooth Low Energy (BLE)," in *IEEE Int. Conf. on Consumer Electronics (ICCE)*, Las Vegas, NV, Jan. 2020.
- [7] Y. Wang, X. Yang, Y. Zhao, Y. Liu, and L. Cuthbert, "Bluetooth positioning using RSSI and triangulation methods," in *IEEE 10th Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, Jan. 2013.
- [8] R. Faragher and R. Harle, "Location Fingerprinting With Bluetooth Low Energy Beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, May 2015.
- [9] K. Urano, K. Hiroi, T. Yonezawa, and N. Kawaguchi, "An End-to-End BLE Indoor Location Estimation Method Using LSTM," in *Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, Kathmandu, Nepal, Nov. 2019.
- [10] X. Xu, "Machine Learning for Improved Bluetooth Localization," Master's thesis, Eindhoven University of Technology, Oct. 2020, supervisor G. Dolmans.
- [11] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2015. [Online]. Available: <https://doi.org/10.48550/arXiv.1409.1556>
- [12] G. Andreadis, E. Giannelos, P. N. Alevizos, and A. Bletsas, "Hyperbolic DoA Estimation and Static Localization: Comparative Study of Bluetooth and RFID," in *Proc. IEEE Int. Conf. on RFID*, Jun. 2024.
- [13] T. Liu, L. Yang, Q. Lin, Y. Guo, and Y. Liu, "Anchor-free backscatter positioning for RFID tags with high accuracy," in *Proc. IEEE Int. Conf. on Computer Communications (Infocom)*, Toronto, Canada, Apr. 2014, pp. 379–387.
- [14] G. Andreadis, "Phase-based Localization of Low Power Bluetooth Tags with Multi-antenna Receivers and Comparison with RFID Technology," Diploma thesis available in: <https://doi.org/10.26233/heallink.tuc.100609>, School of ECE, Technical University of Crete, Chania, Greece, Aug. 2024, Supervisor A. Bletsas.
- [15] M. Pelka, C. Bollmeyer, and H. Hellbrück, "Accurate radio distance estimation by phase measurements with multiple frequencies," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2014.
- [16] Silicon Labs, "AN1297: Custom Direction-Finding Solutions using the Silicon Labs Bluetooth Stack." [Online]. Available: <https://www.silabs.com/documents/public/application-notes/an1297-custom-direction-finding-solutions-silicon-labs-bluetooth.pdf>