Energy-Efficient Indoor Positioning for Mobile Internet of Things Based on Artificial Intelligence

1st Alper Saylam
Dept. of Electrical and Electronics Engineering
Yaşar University
Izmir, Turkey
saylamalper7@gmail.com

2nd Rifat Orhan Cikmazel
Dept. of Electrical and Electronics Engineering
Yaşar University
Izmir, Turkey
rcikmazel@gmail.com

3rd Nur Kelesoglu
Dept. of Electrical and Electronics Engineering
Yaşar University
Izmir, Turkey
nuurkelesoglu@gmail.com

4th Mert Nakip
Institute of Theoretical and Applied Informatics
Polish Academy of Sciences (PAN)
Gliwice, Poland
mnakip@iitis.pl

5th Volkan Rodoplu
Dept. of Electrical and Electronics Engineering
Yaşar University
Izmir, Turkey
volkan.rodoplu@yasar.edu.tr

***Abstract*—We develop an energy-efficient indoor positioning system based on Artificial Intelligence (AI). In our system, first, at the positioning layer, a Multi-Layer Perceptron (MLP) estimates the current indoor position of an IoT device based on positioning indicators obtained from the anchors. Second, at the forecasting layer, a pair of MLPs estimate the future positions of the device based on the past position estimates obtained when the device woke up as well as the forecast positions of the device during the sleep periods. Third, the device is awakened to send a positioning beacon at intervals over which a significant displacement is predicted to occur by the forecasting layer. Our results demonstrate that our indoor positioning system saves significant energy via adaptive sleep cycles whose duration is determined by the prediction of a significant displacement. This work establishes a foundation for indoor positioning that utilizes AI-based positioning and trajectory forecasting.**

***Index Terms*—Indoor positioning, energy-efficient, machine learning, artificial intelligence, forecasting**

I. Introduction

Indoor positioning is one of the key challenges that face the development of Internet of Things (IoT) systems. While accurate position information can be obtained when IoT devices send positioning beacons to the network at frequent intervals, this leads to significant transmit energy consumption, which drains the limited battery supply of an IoT device [1], [2]. As a result, it is imperative that indoor positioning systems that balance the need for accurate positioning and low energy consumption be designed in next-generation networks that will serve a multitude of battery-limited IoT devices [3].

The algorithm that appears in this work was funded by TÜBİTAK (The Scientific and Technological Research Council of Turkey) under Project #1139B411901515 as part of the 2209-B program, where the industry sponsor was SADELABS (SADE Teknoloji, Inc.), Izmir, Turkey.

Our main goal in this paper is to develop a novel energy-efficient indoor positioning system based on Artificial Intelligence (AI). While the trade-off between indoor positioning accuracy and transmit energy consumption of IoT devices is well-known and systems that adapt the sleep cycle duration to conserve energy have been designed, to the best of the authors' knowledge, this is the first work that uses AI-based estimation for the current position in tandem with AI-based forecasting of the future trajectory of an IoT device in order to determine the sleep cycle duration.

In our system, first, at the positioning layer, a Multi-Layer Perceptron (MLP) estimates the current indoor position of an IoT device based on positioning indicators obtained from the anchors. Second, at the forecasting layer, a pair of MLPs estimate the future positions of the device based on the past position estimates obtained when the device woke up as well as the forecast positions of the device during the sleep periods. Third, the device is awakened to send a positioning beacon at intervals over which a significant displacement is predicted to occur by the forecasting layer.

Our results show that our indoor positioning system saves significant energy via adaptive sleep cycles whose duration is determined by the prediction of a significant displacement. Furthermore, even though our design uses the past forecast positions (which themselves contain potential forecasting errors) during the sleep cycles of the device when the position estimate is not available to the network, we demonstrate that the forecasting error converges. That is, even though there is error propagation that results from the use of past forecasts in forming future forecasts, our system is able to achieve a stable and reasonable value of the forecasting error during its dynamic operation.

The rest of this paper is organized as follows: In Section II, we describe the relationship of this work to the state of the art in this area. In Section III, we state the assumptions that underlie our work. In Section IV, we describe our system design. In Section V, we discuss our results. In Section VI, we state our conclusions.

II. Relationship to the State of the Art

In this section, we contrast our work against the state of the art, which we describe in three categories: (1) the past works that predict the trajectories of pedestrians and the mobile devices; (2) the past algorithms that reduce the average transmit power of a mobile device; (3) the past works that have examined the relationship between energy savings and positioning accuracy.

In the first category, the future trajectories of the mobile devices or the pedestrians have been predicted by applying pattern recognition algorithms: In [4], [5], the future trajectories of the pedestrians are forecast by machine learning models based on past positions of the pedestrians. The future trajectories of the pedestrians are adjusted in advance in order to avoid the collisions between them. In [6], by applying the Long Short Term Memory (LSTM) model, the pedestrian trajectories are forecast based on both the influence of social neighborhoods and scene layouts. In [7], the future trajectories of the different traffic agents, such as vehicles, bicycles, pedestrians, are forecast via an LSTM model. The forecast trajectories are utilized to help the navigation decisions of autonomous vehicles. In contrast with these articles, which do not use trajectory prediction for saving energy, the focus of our work is the determination of future trajectory of a device to save energy in indoor positioning.

In the second category, Reference [1] introduces a system called SmartDC, which predicts mobility patterns based on adaptive duty cycling; however, this reference aims to maximize the positioning accuracy under a given energy constraint. Reference [8] proposes SenseTrack, which is a location tracking service that adjusts the GPS sampling time of the mobile devices by utilizing the information from an accelerator and an orientation sensor in order to reduce the energy consumption of the mobile devices. In [9], an active sampling algorithm is introduced to reduce the energy consumption of mobile devices. This algorithm utilizes a sleep mechanism that has a random back-off time, which avoids invalid sampling. Reference [10] proposes a peer-to-peer navigation (ppNav) system that navigates the users to their destination points via tracking. It warns the users of potential deviation based on previous users' traces. In addition, the ppNav algorithm is only utilized during the navigation in order to increase the battery life of the navigation devices carried by users. In [3], a combination of the battery saving algorithm and WiFi fingerprinting is proposed. Since, the mobile device scans based on human activity, energy consumption of the mobile device is reduced. In contrast with these articles, which do not utilize AI-based forecasting, our system forecasts the future trajectory of the mobile device in order to reduce the energy consumption of the device.

In the third category, we compare our work against past works that address both energy savings and positioning accuracy. Reference [11] develops a positioning system that utilizes an accelerometer, which detects when the mobile device moves and uses this to reduce the energy consumption of the mobile device. Reference [2] proposes a system called by EnTracked, which schedules position updates in order to minimize the energy consumption and optimize the positioning accuracy. Reference [12] develops an energy-efficient positioning system called by EnLoc, which optimizes the positioning accuracy for a given energy budget. By identifying the trade-off between energy efficiency and accuracy, Enloc computes the probability map of the user's mobility patterns and predicts the mobility patterns of the mobile devices based on this probability map in order to achieve high positioning accuracy. In contrast with these articles, we do not utilize any sensor measurements but rather forecast the future trajectory of the mobile IoT device.

III. Assumptions

We assume that there exists a "deployment region", denoted by $\mathcal{R}$, over which the mobile device has been deployed. We assume that there is a set of M indoor positioning anchors, denoted by $\mathcal{M}$. Each anchor in $\mathcal{M}$ has a fixed position in $\mathcal{R}$.

In this paper, we focus on a single IoT device ("device" for short) whose position we aim to determine. We assume that this device transmits a signal, called a beacon, to all of the anchors in $\mathcal{M}$ in each slot.[1] We assume that whenever each anchor receives a beacon from the device, it decodes the beacon in order to recover a positioning indicator, such as the Angle of Arrival (AoA) information or the Received Signal Strength Indicator (RSSI).[2]

We assume that there is a gateway G to which all of the anchors are connected. We assume that G requires L successive beacons from the mobile device in order to compute the an estimate of the position of the device based on these beacons. We assume that there is a downlink channel from G to the device that is available whenever the device does not sleep. In addition, we define a "position measurement duration", denoted by T, for the device as the time interval during which the L beacons from the device are collected.

IV. System Design and the Positioning Interval Based on Displacement (PID) Algorithm

In this section, we describe our energy-efficient positioning system. Fig. 1 displays the block diagram of our system for a single mobile IoT device.[3] We have separated the block diagram into two main blocks (enclosed by dashed

[1] We assume that collision-free scheduling of the beacons across multiple frequency bands from all of the devices has been achieved at all of the anchors. We do not address multiple IoT devices in this work.

[2] Our system design works for any such positioning indicator.

[3] A replica of this system exists for each device in an actual system that has multiple IoT devices. Note that the indoor positioning system of each device is treated individually; there is no interaction between the positioning problems of multiple devices.

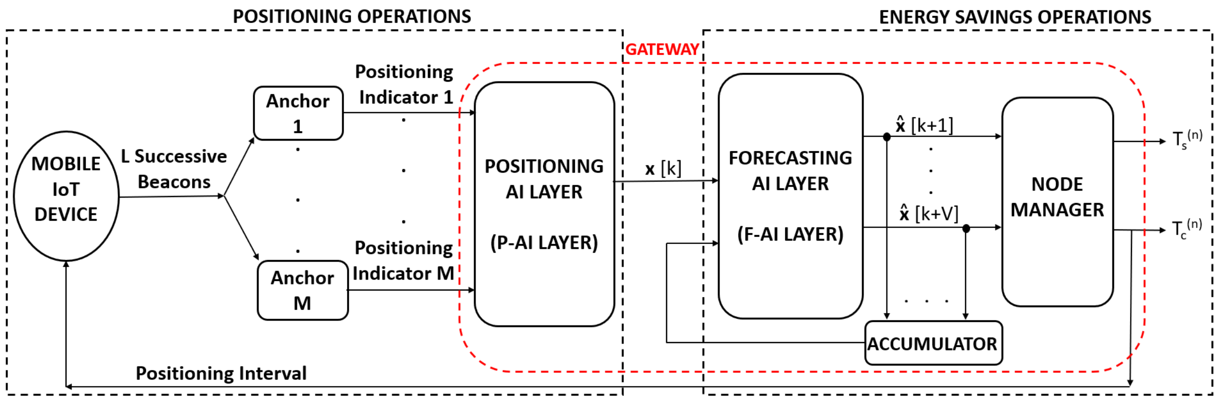

Fig. 1. System design

black lines), namely the positioning and the energy savings operations.

In the block for positioning operations, the device transmits its successive L beacons to all of the anchors in $\mathcal{M}$. When each anchor in $\mathcal{M}$ receives these successive beacons, it decodes each beacon and recovers the positioning indicator in that beacon. In Fig. 1, the following modules are located in the gateway G: Positioning Artificial Intelligence (P-AI) layer; Forecasting Artificial Intelligence (F-AI) Layer and the Node Manager. The positioning indicators are collected from each anchor in $\mathcal{M}$ by the gateway G. Then, G utilizes these indicators in the P-AI layer in Fig. 1 to compute the position estimate $\mathbf{x}[k]$ of the mobile device for each k, where k indexes the successive position measurement durations, each of which lasts T seconds.

In the block for energy savings operations in Fig. 1, the input to the F-AI layer is comprised of the current position estimate produced by the P-AI layer as well as the accumulated past forecasts that have been formed by the F-AI layer itself. Whenever a past position estimate is not available from the P-AI layer (when the device is asleep), the past positions are filled in by the past forecasts.

We assume that the F-AI layer takes a window of the past U such values in order to estimate a window of V future values of the position of the device, as shown in Fig. 1. Note that the output of the F-AI layer is a two-dimensional vector that consists of the forecast 2D positions of the device, denoted by $\hat{\mathbf{x}}[k+v]_{v\in\{1,\dots V\}}$.

We call the duration for which the device sleeps the "sleep period" (which is variable in our design). We denote the nth sleep period by by $T_s^{(n)}$, where we take $n = 0$ as the index of the current sleep period, $n < 0$ indexes the past and $n > 0$ indexes the future sleep periods, counting from the current one. Furthermore, we let $T_c^{(n)}$ denote "positioning interval" of the IoT device, which is the duration between the successive wake-ups of that device.[4] Thus, $T_c^{(n)} = T + T_s^{(n)}$; that is, the positioning interval is comprised of the ON period of the device during which the device sends beacons and the sleep period of that device.

We now state a novel algorithm, which we call the "Positioning Interval Based on Displacement" (PID) Algorithm. This algorithm is executed by gateway G. We divide the region $\mathcal{R}$ into sub-regions, each of which is called a "cell". In our design, each cell is a square; the cells tessellate the deployment region. We let $c[k+v]$ denote the cell in which the device falls at discrete time $k+v$, where k is the current discrete time and each discrete time slot is of duration T.

Whenever the F-AI layer forecasts $\hat{\mathbf{x}}[k+v]_{v\in\{1,\dots V\}}$, the Node Manager assigns the forecast positions of the device to the cells in which these positions fall. We let $\hat{\mathbf{c}}[k+v]_{v\in\{1,\dots V\}}$ denote the vector of cells in which the forecast positions fall, up to V, which denotes the maximum value of the step-ahead forecast. Then, the Node Manager examines the vector $\hat{\mathbf{c}}[k+v]_{v\in\{1,\dots V\}}$ and finds the discrete time v^* at which the device is predicted to cross over from its current cell to an adjacent cell. The node manager sets the sleep duration $T_s^{(n)}$ to Tv^*; that is, the device is put to sleep until the time that it is predicted to change cells.[5] Then, the Node Manager communicates to the device the value of $T_c^{(n)}$ (i.e. when the device needs to wake up next) based on this value of $T_s^{(n)}$.

Fig. 2 and Fig. 3 illustrate the PID algorithm. In Fig. 2, we show the sleep period $T_s^{(n)}$ and the positioning interval $T_c^{(n)}$ of the IoT device for 6 such past periods. In this figure, each vertical bar represents a transmitted beacon in a sequence of L beacons.[6] We see that the device sleeps for a potentially variable duration in each period. Fig. 3 shows that for the same results in Fig. 2, the future forecast positions are computed based on the past vector of a combination of estimated positions (for those times when the beacons arrived) as well as the past forecast positions that fill in the slots when no such estimates are available.[7]

[4]Recall that the device wakes up to send a series of beacons and then goes to sleep. In addition, note that the positioning interval is of variable length in our design.

[5]The value of V must be selected large enough such that the first cross-over occurs by the Vth step-ahead forecast.

[6]Only a subset of these beacons are shown in the figure. The beacons do not necessarily have irregular spacing.

[7]Recall that the duration of each slot is T.

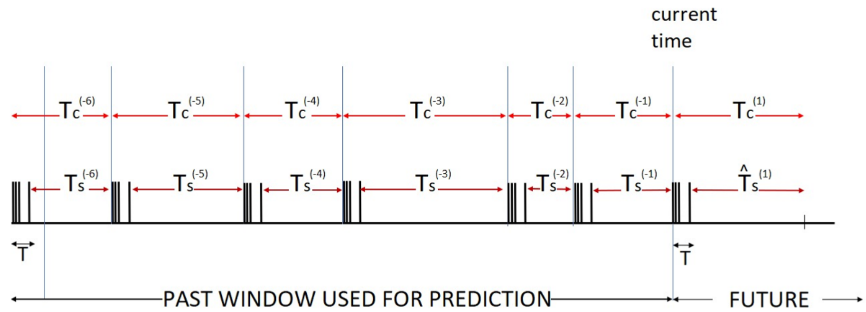

Fig. 2. Illustration of successive positioning intervals and sleep periods

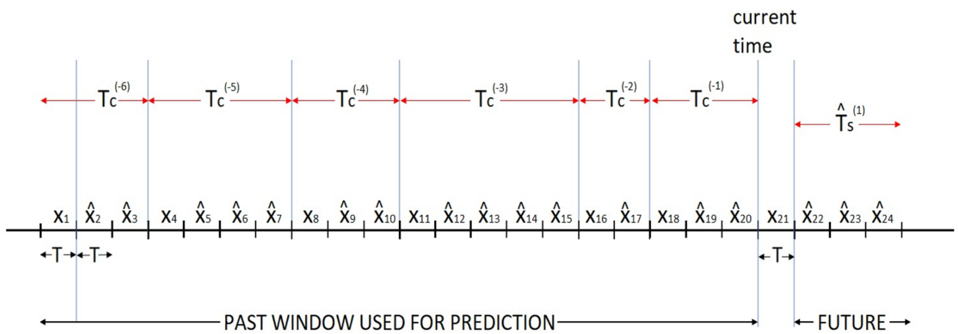

Fig. 3. Illustration of the formation of forecasts by the F-AI layer

V. Results

A. Experimental Methodology

1) Experimental Setup: We shall demonstrate our results under a Bluetooth Low Energy (BLE) positioning system; however, we emphasize the design that we have presented in this paper is general and can be applied to other underlying positioning technologies.

Fig. 4. Experimental setup

The setup of our positioning system is shown in Fig. 4. In this figure, the 4×4 m region enclosed by the red lines is the deployment region. Note that there are some obstacles in this area in addition to heavy reflections off the floor. Hence, this represents a challenging environment for AoA positioning.

Our setup consists of two anchors, a single mobile IoT device, and a computer that emulates the gateway. The anchors are located in two distinct corners of the deployment region. Each anchor is 1.95 meters above the ground. In Fig. 4, we assume that the origin of the deployment region is the corner closest to the viewer; the coordinates of the first anchor are (0, 4, 1.95) and those of the second anchor are (4, 0, 1.95), all in meters. In addition, we have designed the gateway modules in Python (Version 3.6) in the computer. In this setup, Texas Instruments AoA BoosterPack (BOOSTXL-AoA) and the CC2640R2F evaluation board were used. The evaluation board is utilized as the transmitter of the mobile IoT device, and the combination of the (BOOSTXL-AoA) and CC2640R2F serve as an anchor. In addition, the cell size is chosen to be 1 m by 1 m, since the average positioning error was found to be 1.09 meters.

First, the device connects to the two anchors and starts to broadcast successive beacons on the Bluetooth uplink channels every 100 ms. Then, each anchor computes its AoA and RSSI values and the BLE channels are utilized for communication between the anchors and the device.[8] They are then read via Python in the computer via serial port communication. The number L of successive beacons that are required by the gateway is set to 10. When the computer receives 10 beacons, the 2D positions of the device are computed by P-AI layer. Thus, the parameter T is 1 s, during which the gateway computes one position estimate of the device.

2) Data Collection Methodology: We collected the data on our deployment region in order to train the P-AI and the F-AI layers. Data were collected by moving the mobile device within the deployment region. While collecting data, the device was carried by a pedestrian, and it was kept approximately 1 meter above the ground as the pedestrian

[8]By default, the average transmit power consumption of the device is 48.75 μW.

moved in the deployment region.

A total of 76800 samples were collected. We have separated our data set into the two disjoint sets, which are used for the training and the test stages of the P-AI and F-AI layers. Each sample in our data set is comprised of the AoA and RSSI values, channel information for each anchor, dimensions of the deployment region, and the 3D coordinates of each anchor. The AoA values, RSSI values, channel information and dimensions of the deployment region are as follows: AoA1 and AoA2 denote the AoA that reaches the first and second anchors, respectively, from the mobile IoT device. Its units are degrees, and the range is between -100° and 100°. RSSI1 and RSSI2 denote the received signal strength indicator (RSSI) value measured by the first and second anchors, respectively. Its unit is decibel (dB). Channel1 and Channel2 denote the BLE channels used by the first and second anchors, respectively, in order to communicate with the mobile IoT device. In our setup, channels 37, 38 and 39 were used, which are the advertising channels of BLE.

3) Positioning Methodology Based on Machine Learning: We now describe a machine learning model that is utilized in the P-AI layer in our system. We use a Multi-Layer Perceptron (MLP) model in order to estimate the current position of the device based on the beacons received in a duration of T. In the MLP model, there are five layers: one input layer, three hidden layers and one output layer. The input layer has 8 neurons, the hidden layers have 32, 32 and 10 neurons, respectively, and the output layer has 2 neurons (one for each of the x and y coordinates).[9]

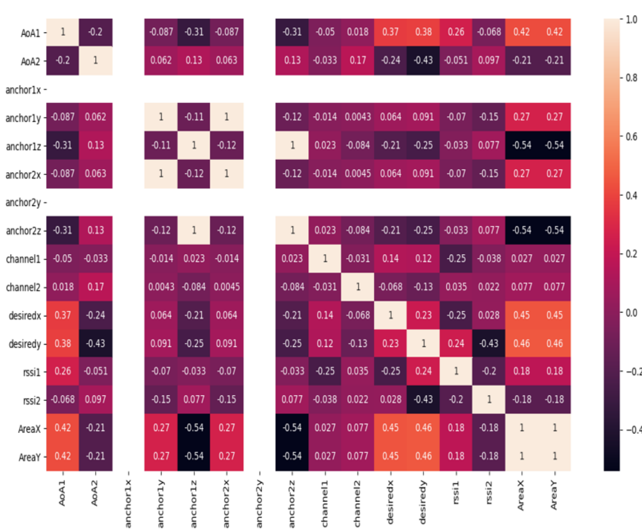

Fig. 5. Heatmap of the correlation coefficients of the features

In order to increase the performance of position estimation, we have applied feature selection on the input variables of the MLP model. We have examined the correlations of 16 distinct features and have chosen the features that have the highest magnitudes of correlation with the desired outputs for the position estimate. Fig. 5 shows the heatmap of the correlation coefficients of 16 distinct features as well as those of the desired outputs. The features that have the highest magnitudes of correlation with the desired outputs are as follows: The AoA values for each anchor, the RSSI values for each anchor, the communication channels for each anchor, and the dimensions of the deployment region.

The model parameters are selected as follows: The activation function of each neuron in the MLP is chosen as Rectified Linear Unit Function (ReLU). Adam (Adaptive Movement) is utilized as the optimizer. The number of epochs and the batch size parameters are chosen as 250 and 20 respectively. The loss function is chosen as Mean Squared Error Loss (MSE). In addition, 10-fold cross-validation is applied during the training of the data set.

4) Forecasting Methodology Based on Machine Learning: In this section, we describe our F-AI layer forecasting model. The x and y coordinates are separately forecast by two distinct MLP models. In each model, there are two layers, namely the input and the output layers, whose number of neurons are 20 and 10 respectively. The input and output of each MLP model are the past positions and the forecast future positions, respectively. In addition, the parameters of both of these MLP models are the same as those used in the P-AI layer above.

B. Performance Evaluation

In this section, we present the performance evaluation of our system and our PID algorithm.

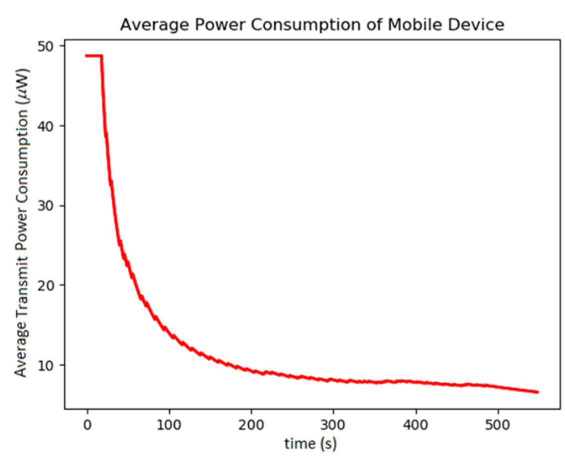

Fig. 6. The average transmit power consumption of the device during the demonstration

Fig. 6 shows the average transmit power consumption of the device during our demonstration. In this figure, we see that from $t = 0$ to $t = 20$, the average power consumption remains constant. On that interval, the device consumes an average transmit power of 48.75 μW while sending beacons to the anchors in order to collect a sequence of 20 positioning indicators that are required by the P-AI layer. Subsequently, the F-AI layer is able to start forecasting the future positions of the device at $t = 20$ s. From $t = 20$ to $t = 500$ s, the average transmit power consumption decreases to 6.62 μW due to the

[9] A randomly generated collection of MLP models were tested in order to determine the number of layers and the number of neurons in each layer. We found the local optimal MLP architecture for our data set.

increase in the positioning interval of the device and keeps decreasing after $t = 500$. This result shows that our system reduces the average transmit power consumption of the device as the system evolves from a cold start towards steady-state operation.

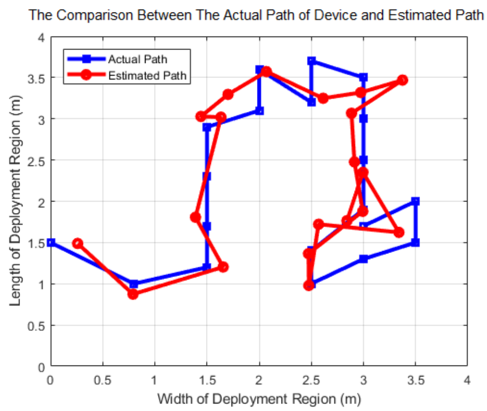

Fig. 7. The actual and estimated trajectories of the device

Fig. 7 demonstrates the position estimation performance at the output of the P-AI layer for consecutive positions in the deployment region. In this figure, the red line represents the path that consists of the consecutive estimated positions, and the blue line represents the path that consists of the consecutive actual positions. The results in this figure shows that the proposed system achieves relatively high accuracy for indoor positioning while significantly reducing the average transmit power consumption (as shown in Fig. 6).

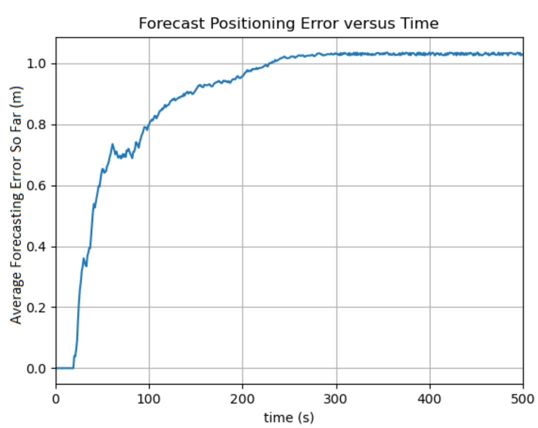

Fig. 8. The average forecasting error so far at the output of the F-AI layer during our demonstration

In Fig. 8, we show the forecasting performance of the F-AI layer based on average forecasting error. Since F-AI layer waits for the 20 positions of the device which are estimated by P-AI layer before it begins to forecast future positions, from $t = 20$ to $t = 300$ s, the average forecasting error increases. The reason is that the device sleeps between successive wake-ups. During these sleep periods, since the estimates of the current position are not available, the past positions are filled in by the past forecasts, which causes error propagation and thus increases the average forecasting error. However, we see that the average forecasting error converges to 1.09 meters in steady state. That is, despite this error propagation, the system settles down to a steady-state average forecasting error, whose magnitude is very reasonable for a BLE indoor positioning system based on AoA.

VI. Conclusion

We have designed a novel energy-efficient indoor positioning system and the "Positioning Interval based on Displacement" (PID) algorithm that adaptively selects the sleep duration of a mobile IoT device based on forecasts of the future trajectory of the device. Since our algorithm wakes up the device in order to receive positioning beacons only when a significant displacement is predicted to have occurred, the mobile IoT device saves significant energy via adaptive sleep cycles. Our experimental demonstration uses Multi-Layer Perceptron (MLP) models for both position estimation as well as position forecasting in two distinct layers in our system design. In our future work, we plan to demonstrate the effectiveness of our design under other underlying positioning technologies besides Bluetooth Low Energy (BLE).

References

[1] Y. Chon, E. Talipov, H. Shin, and H. Cha, "Mobility prediction-based smartphone energy optimization for everyday location monitoring," in *Proceedings of the 9th ACM conference on embedded networked sensor systems*, 2011, pp. 82–95.

[2] M. B. Kjærgaard, J. Langdal, T. Godsk, and T. Toftkjær, "Entracked: energy-efficient robust position tracking for mobile devices," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, 2009, pp. 221–234.

[3] J. L. S. González, L. M. S. Morillo, J. A. Álvarez-García, F. E. D. S. Ros, and A. R. J. Ruiz, "Energy-efficient indoor localization WiFi-Fingerprint system: An experimental study," *IEEE Access*, vol. 7, pp. 162 664–162 682, 2019.

[4] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.

[5] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 261–268.

[6] H. Xue, D. Q. Huynh, and M. Reynolds, "SS-LSTM: A hierarchical LSTM model for pedestrian trajectory prediction," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1186–1194.

[7] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6120–6127.

[8] L. Zhang, J. Liu, and H. Jiang, "Energy-efficient location tracking with smartphones for IoT," in *SENSORS, 2012 IEEE*. IEEE, 2012, pp. 1–4.

[9] X. Liu, Y. Zhan, and J. Cen, "An energy-efficient crowd-sourcing-based indoor automatic localization system," *IEEE Sensors Journal*, vol. 18, no. 14, pp. 6009–6022, 2018.

[10] Z. Yin, C. Wu, Z. Yang, and Y. Liu, "Peer-to-peer indoor navigation using smartphones," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1141–1153, 2017.

[11] I. Shafer and M. L. Chang, "Movement detection for power-efficient smartphone WLAN localization," in *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems*, 2010, pp. 81–90.

[12] I. Constandache, S. Gaonkar, M. Sayler, R. R. Choudhury, and L. Cox, "Enloc: Energy-efficient localization for mobile phones," in *IEEE INFOCOM 2009*. IEEE, 2009, pp. 2716–2720.