Indoor Positioning System using LSTMs over WLAN Network

Pranav Sankhe, Saqib Azim, Sachin Goyal, Tanya Choudhary,
Kumar Appaiah[*], Sukumar Srikant[**]
Indian Institute of Technology, Bombay, India
{pranav_sankhe, saqib_azim, sachin_goyal, tanya.choudhary}@iitb.ac.in
akumar@ee.iitb.ac.in srikant@sc.iitb.ac.in

Abstract*—The Global Navigation Satellite Systems (GNSS) like GPS suffer from accuracy degradation and are almost unavailable for indoor environments. Indoor positioning systems (IPS) based on WiFi signals have been gaining popularity. However, owing to the strong spatial and temporal variations of wireless communication channels in the indoor environment, the achieved accuracy is around several tens of centimeters. We present here, the detailed design and implementation of a self-adaptive WiFi based Indoor Positioning System using LSTMs. The system is novel in its method of achieving high accuracy by overcoming possible causes of channel variations and is self-adaptive to the changing environmental conditions. The proposed design has been developed and physically realized on a WiFi Network consisting of ESP8266 (NodeMCU) devices. The experiments were conducted in a real indoor environment while changing the surroundings to establish the adaptability of the system. Our model achieves an accuracy of 5.85 cm with a confidence interval of 93% outperforming the other approaches reported in the literature with a statistically significant margin.

***Index Terms* - Indoor Localization; WiFi; Received Signal Strength Indicator (RSSI); Long Short-Term Memory Network (LSTM).**

I. Introduction

Indoor positioning systems are aimed at solving the problem of localization of the objects and devices in closed rooms or in buildings where GPS signals cannot reach due to high attenuation. The exact location of objects relative to the environment is a crucial information for asset tracking, security, human-computer interface (HCI) applications as well as tasks like helping someone to find his or her way in and around an unknown building.

Currently, the Global Positioning System (GPS) is used for outdoor localization. GPS cannot be used indoors because the physics of radio propagation rules out the reception of weak GPS microwave signals indoors. Also, the reported accuracy of GPS is around 4 m and hence is insufficient for high accuracy demanding indoor positioning. Consequently, a lot of work has been done to develop similar kind of systems for closed indoor environments with sub-meter range accuracy.

The existing indoor positioning models are based on acquiring various signal parameters like Received Signal Strength Indicator (RSSI), Channel State Information (CSI), Angle of Arrival (AoA), Time Difference of Arrival (TDoA) in case of multichannel communication, etc. These models use techniques like trilateration and fingerprinting to localize the object. We discuss each of these approaches in detail in Section II.

Recently Deep Learning has been successfully applied to solve a wide spectrum of challenging problems. Deep learning models are able to extract complex features from the training data. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTMs) have performed excellently especially in the classification tasks. Hence we formulate the indoor localization problem as a classification task based on RSSI values acquired from the WiFi signals. We use LSTMs for classification as they have an inherent ability to learn spatial as well as temporal patterns making them suitable since RSSI values are recorded across time. We discuss more about the choice of LSTMs in Section III.

To the best of our knowledge, LSTMs have not been used previously on RSSI values to solve the problem of indoor localization. One of the important contributions of this paper is to incorporate the time dependence of RSSI values and show that LSTMs are well suited to develop systems based on RSSI. The relation between the distance and the RSSI value depends on the topology of the indoor environment as well. To model the environment topology, we propose a unique set-up consisting of 4 stationary WiFi nodes. We discuss the rationale behind the choice of four stationary nodes to model the multipath fading effects and one moving node consisting of the object being tracked to model the shadowing effects in greater detail in Section III.

experimentally
show that LSTMs
...

why unique ?

why 4
and not 5 ?

II. Literature Review

A. Indoor Positioning Methods

We give a brief description of various approaches for indoor positioning problem.

RSSI based approaches are most common and easily deployable for indoor localization ([1], [2], [3]) since RSSI values can be acquired easily from any WiFi based device like mobiles, laptops, etc. Here, the distances of the receiver from each of the transmitter is found using path loss models based on RSSI values acquired by the receiver. The average reported accuracies in these approaches ranges from 2 m - 4 m [1]. The calculations assume that all points lying on the boundary of a circle with transmitter at the center will have the same RSSI. This is not true in general, since the symmetry is dependent upon the environment, as explained

[*] Prof. Kumar Appaiah is a Faculty in Department of Electrical Engineering, IIT Bombay

[**] Prof. Sukumar Srikant is a Faculty in Systems and Control Department, IIT Bombay

affect

in Section III. This can adversely effect the accuracy of mapping RSSI to distance.

how are we exactly demonstrating that our system is feasible to realize and universal ?

Another approach would be to use the Angle of Arrival (AoA) of rays that can be inferred from the channel state information (CSI) of the channel between the sender and receiver [4]. Most such techniques are either not practically feasible to realize, or are not universal. ArrayTrack [5] requires 6 - 8 antennas, LTEye [6] requires rotatory antenna and Ubicarse [7] requires motion sensors on the tracking device and also need the user to rotate the device by at least 180°, thus limiting their utility and scalability.

basically a look up table

Fingerprinting based approaches ([8], [9]) involve scanning the area of interest and creating a database of recorded signal characteristics. During the test phase, the test signal is mapped to the nearest point in the recorded dataset using multivariate analysis, support vector machines [4] and K-nearest neighbours [10]. These approaches are inherently limited due to their inability to extract relevant and complex features from the data and use them in a similar way as is done in the deep learning based models.

Previous approaches for deploying fully connected neural networks [11], RNNs [12] and CNNs [13] for localizing objects used the acquired RSSI, CSI or AoA values to train the model. Lukito et al. [12] perform classification of the object location over 42 different public places using the signal RSSI values and report an accuracy of 83%. This does not exactly account to indoor positioning since the public places are quite far apart from each other. Therefore the acquired RSSI values inherently show a large difference and hence are easily classifiable compared to our case where the order of separation is in centimeters. Wang et al. [11] report an accuracy of 94 cm in a room of 4 m × 7 m dimension.

There have also been approaches based on ultrasonic sensors [14], visible light communication [15], RFIDs [16] and magnetic and light sensors data with LSTMs [17]. However, the scalability of these approaches is limited due to the hardware requirements and complexity compared to WiFi signal based methods.

B. Long Short-Term Memory (LSTM) Networks

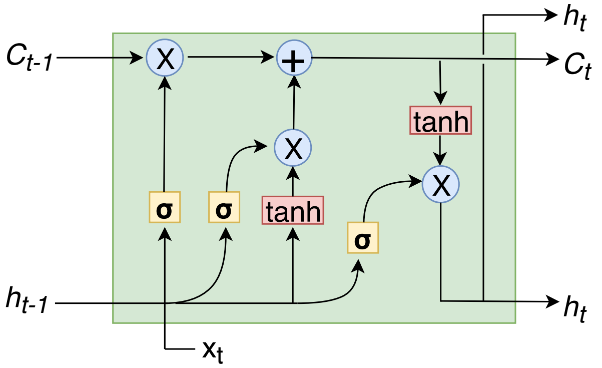

Figure 1: Representation of an LSTM Cell [18]. C_t represents the cell state at time step t, h_t represents the cell output at time t and σ represents sigmoid layer.

LSTMs are special kind of RNNs designed to remember and learn the long-term time dependency of data. Here we briefly describe LSTM architecture [18]. Key to their structure is the "cell state" to which information can be added by the cells through gates. Gates are sigmoid neural network layers which optionally let the information pass through and are of three types:

1) Forget Gates: A sigmoid layer is used to decide what information we are going to throw away from the current cell state. The output of this layer is multiplied with cell state to delete the information we do not want to proceed further.
2) Input Gate Layer: This consists of a tanh layer which creates a new vector to be added to the current state and a sigmoid layer to select which values to add in particular. The tanh and sigmoid layers are multiplied and then added to the current cell state.
3) Output Gate: The current cell state is run through the tanh layer and multiplied with a sigmoid layer to output only the desired parts.

III. System Design

A. Wireless Network

As shown in Fig. 2, our positioning system is a WiFi network consisting of a Wireless Access Point (WAP) and several WiFi clients (nodes). Wireless Access Point sets up the network to which the nodes can connect. We classify the nodes in two distinct categories based on their position and their motion. The fixed nodes are placed at the corners of the room or arena and have their position fixed throughout the experiment. The moving node keeps changing its position throughout and we track the distance between this node and the WAP. The strength of signal received at the nodes is dependent on their distance from the WAP. This relation between the signal strength and the distance is governed by various path loss models in wireless communication which we discuss here and also explain how they inspired our design. We also elaborate on how we relate the various path loss models and our machine learning (ML) architecture.

dont repeat multipl times

better to give name to two different typ of nodes

In the case of line of sight transmission, the received power can be modeled to decay as the inverse square of the distance. This model is called Free-Space Path Loss Model, wherein Friis free space equation relates the received and transmitted power as

$$P_r = \frac{P_t A_t A_r}{\lambda^2 d^2} \tag{1}$$

where P_r is received power, P_t is transmitted power, A_t and A_r are parameters of transmitter and receiver antennas respectively, λ is the carrier wavelength and d is the distance between the two antennas. Free-Space Path Loss model is applicable to situations where the transmission distance is much larger than the antenna size and λ as well as there are no obstacles between the transmitter and the receiver. This assumption holds good in long-range communication systems such as satellite communication. However in case

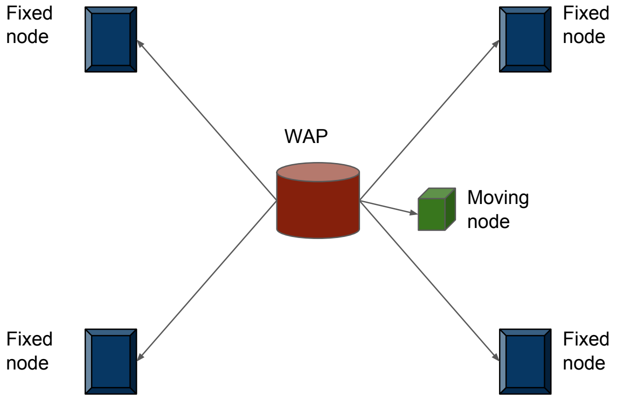

Figure 2: The WiFi Network model, which consists of a wireless access point (also acting as reference) and four fixed nodes at the corners of rectangular arena and a moving or target node which needs to be localized. The position of the fixed nodes with reference to WAP is known while the position of moving node needs to be calculated.

of the indoor localization problem, the distance ranges are such that the fading effects become substantial. In order to account for the attenuation caused due to the obstacles in the path between the transmitter and the receiver, the Free-Space model is modified to Simplified Path-Loss Model. The modified Friis equation for Simplified Path-Loss Model is

$$P_r = P_t K \left[\frac{d_0}{d}\right]^{\gamma} \tag{2}$$

where P_r is received power, P_t is transmitted power, d_0 is the reference distance for the antenna far field, K is the constant path-loss factor and γ is the path-loss exponent. However this simplified path loss model does not account for the fact that the same transmitter-receiver distance usually has different path loss according to the environment. This indeed is the case for localization in a cluttered room. The simplified path-loss model represents an average channel attenuation and hence is not applicable here.

From empirical measurements, it has been shown that the differences between the average and the actual path loss follows a log-normal distribution. Expressing the path loss in dB, we have

$$P_L(d)[\text{dB}] = P_L(d_0) + 10\gamma \log\left(\frac{d}{d_0}\right) + X_\sigma \tag{3}$$

where, $P_L(d)$ is the power received at distance d, $P_L(d_0)$ is the power received at the reference distance d_0, $X_\sigma \sim \mathcal{N}(0, \sigma^2)$ describes the random shadowing effects and γ is the path loss exponent. Appearance of log-normal distribution is due to the general attenuation model of a signal when it passes through an object of depth d. We can approximately model the attenuation as follows:

$$s(d) = \exp(-\alpha d) \tag{4}$$

where α is the attenuation factor. If we assume α to be approximately same for all blocking objects then,

$$s(d_t) = \exp\left(-\alpha \Sigma_i d_i\right) = \exp(-\alpha d_t) \tag{5}$$

Using the Central limit theorem, d_t can be approximated as gaussian distributed when the number of objects is large.

Shadowing is a large-scale effect, as it corresponds to substantial deviations of the RF signal from its mean due to large obstacles, which create shadow zones that cause deep fades if a receiver enters them. A log-normal shadowing model is more suitable for the indoor localization problem as it provides a number of parameters which can be configured according to different environments. The shadowing effect is spatial in nature. Therefore for the machine learning architecture to learn the parameters of the shadowing model, spatial variation in distance is necessary. This information about the change in distance will be indicated by the signal strength value measured at the moving node enabling the machine learning architecture to learn the shadowing model.

The shadowing model deals with spatial variation only. But in wireless communication, the channel changes with time as well. To capture the temporal changes, we consider multipath model which is both temporal and spatial model. Multipath effects are unwanted and can be mitigated. The unreliability of wireless networks can largely be attributed to multipath fading and hence it causes fairly large deviations from link quality predictions based on path loss models.

We commonly consider probabilistic models for multipath fading. In a network, the received signal measured at each receiver is the result of the superposition of all scattered paths that reach it. In the absence of a dominating component, the envelope of the received signal can be shown to be Rayleigh-distributed [19].

$$p_R(r) = \frac{r}{\sigma^2} \exp\left(\frac{-r^2}{2\sigma^2}\right), r > 0 \tag{6}$$

In the presence of a dominating, static component, which is usually the line of sight path, the envelope obeys a Rician distribution. Its probability distribution can be written as:

$$p_R(r) = \frac{r}{\sigma^2} \exp\left(-\frac{r^2 + A^2}{2\sigma^2}\right) I_0\left(\frac{Ar}{\sigma^2}\right), r > 0 \tag{7}$$

where the parameter, $A > 0$, gives an indication of the peak amplitude of the dominant path and I_0 is the modified bessel function of the first kind and order.

Multipath fading depends on the topology of the environment where the nodes are deployed [20]. In wireless sensor networks, we consider the multipath fading for static nodes only. Hence in order to model the multipath effects and thus the topology of the surroundings in the network, we place these static WiFi nodes at 4 corners of the room. Our machine learning architecture takes as input the signal strength values of all the WiFi nodes in the network where the signal strength information of the moving node helps to learn the shadowing model and the signal strength of the other 4 static nodes help to model the multipath fading model and hence the topology of the indoor environment. Since the 4 static nodes help to model the topology of the surroundings, our system is adaptable to any environment which we demonstrate in our experiments by testing our model on 4 different datasets, all recorded with varying conditions.

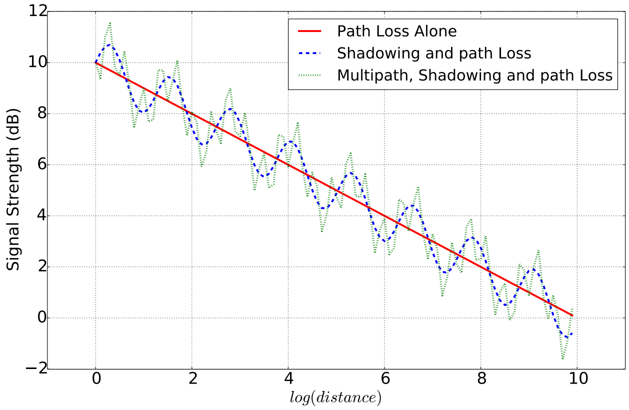

Figure 3: This figure compares the variation of signal strength and distance when we consider the various propagation models in wireless networks. The dotted plot represents the signal strength variation when we consider all the path loss models and is the most accurate representation of the actual scenario amongst the three models described.

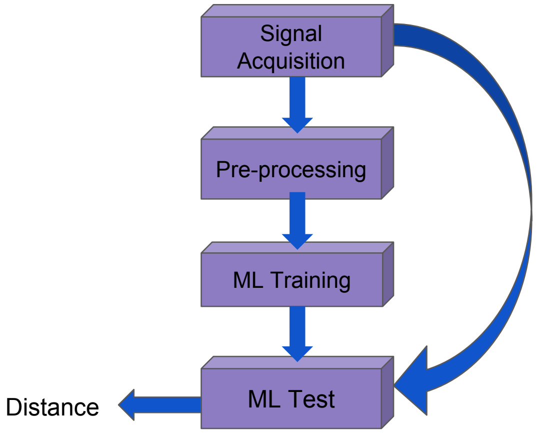

Figure 4: The overall system architecture. The signal acquisition represents the system of WiFi network which provides the received signal strength values. This signal information is pre-processed and fed to the ML architecture where the ML model is trained over this data. We can test the model either during training or after the model has been trained by accessing the signal strength values from the WiFi network.

B. Machine Learning Architecture

We assume that the objects to be localized do not move abruptly and have a smooth trajectory. Measured RSSI is the power of the received signal. Hence assuming the wireless channel to be an LTI system over short duration, received signal can be expressed as the squared convolution of the transmitted signal and the transfer function of the wireless channel. Since the channel characteristics change with time, the present RSSI value will also depend on past RSSI values as well as the channel characteristics over long duration. Therefore, the RSSI values can be modeled as a time series sequence.

We use LSTMs for time series modeling of RSSI values as LSTMs have performed well, especially with the classification of time series sequences and text sequences ([21], [22]). LSTMs are known to learn the long-term dependency as well as spatial and temporal features of the input data. Using LSTMs for time series modeling of RSSI values is one of the main reasons for high accuracy achieved by the proposed model.

We formulate our problem as a classification task, dividing the range of possible object distance values into 30 bins of equal size. Object distance is then classified over these bins using deep learning (LSTMs). As shown in Fig. 5, our deep learning architecture consists of 2 layers of LSTMs each with 64 and 128 cells. At the end of 2 LSTM layers, we get 128-dimensional features. Since the number of bins is 30, we go from the 128-dimensional space to a 30-dimensional logits space so as to apply the softmax layer later and extract the predicted class/bin. Instead of using a single matrix multiplication for this feature space reduction, we compress gradually using fully connected layers of dimension 64, 32 and 30 with a Rectified Linear Unit (ReLU) activation function between them. The activation function is used between the fully connected layers to introduce non-linearity in the model graph.

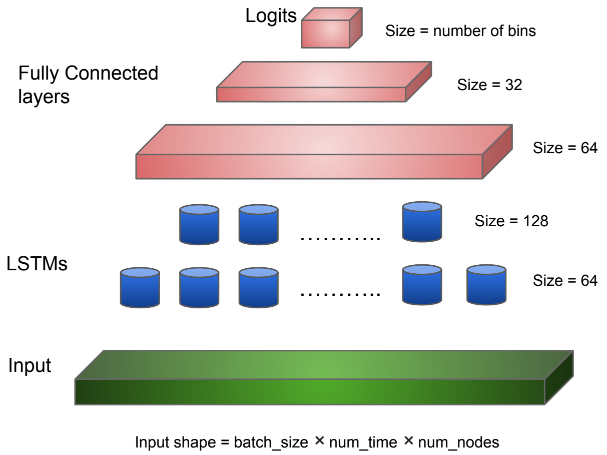

Figure 5: Neural network block diagram. 2 LTSM layers, followed by 3 fully connected layers, interconnected with ReLU. Softmax function is applied on the logits to get the predicted bin label. Num_nodes are the 5 nodes in system and num_time is equal to the time steps in LSTM (20 in our model).

IV. Experimental Evaluation

A. Experimental Setup

The entire experiment was carried out in a rectangular room of dimension 8.46 m × 6.98 m. The central rectangular arena of the room was used for the localization setup which consisted of four reference NodeMCUs at the four corners of the arena (dimension 4.14 m × 2.86 m) along with a TP-Link TLWR840N Router as a Wireless Access Point (WAP) placed as shown in Fig. 2. A three-wheeler programmable circular red patch of area X placed horizontally on top of robot

robot was developed with a circular-red patch on top of it. The robot had a NodeMCU chip mounted on it and was programmed to span the entire arena. In our experiments, this robot is the object which we try to localize or track. The four reference NodeMCU's, as well as the one attached with the robot (tracking object), are wirelessly connected to WAP. These 5 NodeMCU's send RSSI values of their connection with WAP to another coordinator NodeMCU (connected to Central Server via Serial Interface) every 50 ms using UDP. To quantify the performance of the system, a webcam, with frame rate of 60 frames per second and localization accuracy in the range 4 mm, was mounted on the ceiling of the room. An algorithm written in MATLAB detected the circular red patch consequently measuring the bot coordinates (origin at WAP). The position of bot with respect to the origin as detected by the webcam serves as the ground truth for the location estimated by our system.

built

need to mention for collection of dataset

can't we simply mention to create the dataset

what is localization acccuracy in a camera?

vertically opposite to the center of arena

The data for object location and corresponding RSSI values was collected on 4 different days at different day timings to ensure proper variation, if any, due to environmental changes. Each time, the lab setup was changed to create a sense of a different environment by varying the obstructions in and around the arena like chairs and table positions. Finally, after preprocessing of data we obtain 4 datasets on which we train and test our model separately. The data has been divided into 4 : 1 ratio for training and testing the LSTM respectively. Each instance of data consists of 5 RSSI values, the first 4 corresponding to the reference nodes and the last one corresponding to the NodeMCU mounted on the robot. Hence each training dataset dimension is [M × 5] where M is the total number of data samples collected.

can't we name this nodeMCU as target nodemcu instead of everytime telling the oe attached with bot

B. Training

The entire neural network architecture has been developed and trained using Tensorflow. We train and test our model separately on the 4 different datasets collected from our lab to ensure the correctness under varying and different conditions. The input to the LSTM is of dimension [Batch_size × LSTM Time Steps × Number_of_WifiNodes]. Hence we reshape our data into matrices of the above dimensions after normalizing it (mean and standard deviation normalization). The network weights and parameters are initialized using Xavier's initialization [23]. LSTM time steps were kept as 20. Cross-entropy loss function and Adam Optimizer [24] is used to train the model based on computed loss. Gradients were clipped to a maximum of 10 as it led to better and stable training. For completeness, we provide the list of all hyper-parameter values after final tuning in Table I. We train the model on NVIDIA GTX Titan GPU. It takes around 12 hours for 800 training iterations.

C. Results

As stated earlier, in our experiments, object distance varies from 1.51 cm to 3.41 m. We formulate the localization problem as a classification task over 30 equal bins each of length l_{bin} equal to 11.73 cm. Our model predicts the label (0 - 29) of the bin in which the object distance lies and reports the center value of the predicted bin as the predicted distance. We use 2 different accuracy metric to validate our proposed localization system:

Hyper-Parameters	value
Batch Size	1024
LSTM Time Steps	20
Maximum Gradient Norm	10
Learning Rate	10^{-4}
LSTM Sizes	2 layers - 64, 128 cells
Dropout	0.75

TABLE I: List of all network Hyper-Parameter values

1) 5.85 cm Confidence Interval: The center value of the predicted bin is reported as the predicted distance. If the LSTM classifies correctly and hence predicts the correct bin, the upper bound on the error in predicted distance will be half of the bin length (l_{bin}) which is 5.85 cm.

We obtained classification accuracies of 93.94%, 92.51%, 93.89% and 92.99% on the 4 datasets respectively. Thus our model assures a 5.85 cm upper bound on the error in distance with the same confidence intervals as the classification accuracy. Fig. 6 depicts the variation of classification accuracy as the model gets trained on dataset 1. Results obtained are also listed in Table II.

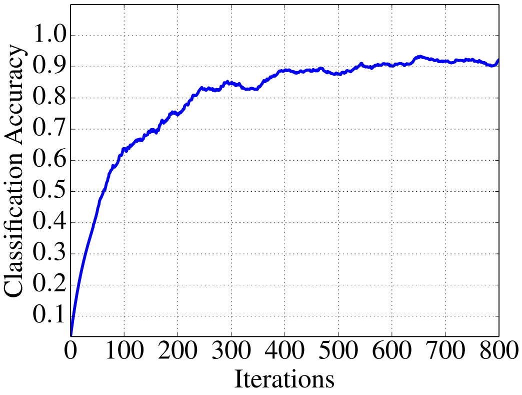

Figure 6: Classification accuracy variation as the neural network gets trained on dataset 1. We train the model for around 800 iterations.

2) Average Upper Bound on Error: We calculate the average upper bound on error in predicted distance over all the test cases including the ones where the model does not predict the correct bin label (around 7% cases as reported in the previous section). Let x be the correct bin label and y be the label predicted by the model. In worst case, upper bound on error in distance (E_{max}) for this observation will be given by

$$E_{\text{max}} = \|x - y\| \times l_{\text{bin}} + \frac{l_{\text{bin}}}{2}$$

We obtained an average error upper bound (E) of 7.67 cm, 7.36 cm, 8.12 cm and 7.55 cm on the 4 datasets respectively. Fig. 7 depicts the variation of E as the model gets trained on dataset 1.

$$E = \frac{\sum_{i=1}^{N} E_{\max}[i]}{N} \tag{8}$$

where N is the total number of test cases. We list all the obtained results in Table II.

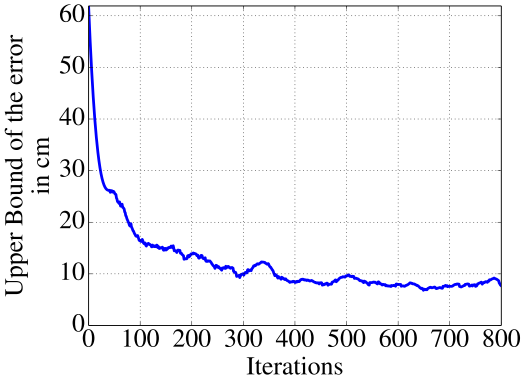

Figure 7: The average upper bound variation as the model training proceeds on dataset 1

Dataset	5.85 **cm Confidence Interval (in** %**)**	**Average Upper Bound on Error**
Day 1	93.94	7.67 cm
Day 2	92.51	7.36 cm
Day 3	93.89	8.12 cm
Day 4	92.99	7.55 cm

TABLE II: Obtained accuracies over the 4 datasets. Our model gives an upper bound on error in predicted distance of 5.85 cm with confidence intervals given in column 2. Column 3 lists the average of upper bounds over all test cases

V. Conclusion and Future Work

In this work, we have presented an indoor localization system that uses the RSSI values provided by the commercially available standard WiFi chips. Our model successfully uses LSTMs and achieves an accuracy of 5.85 cm with a confidence interval of around 93% and hence is a state of the art technology to the best of our knowledge with a statistically significant margin over other approaches. Our indoor localization system is self-adaptive as it can adjust its parameters in order to accurately estimate the position in different environments. We take into account the unwanted multipath fading effects without adding any extra hardware components dedicated to selectively filtering of the multipath effects. Also, since this system has been developed using the existing infrastructure of WLAN, the deployment cost is very low, making it commercially viable.

where have we proved that it is self-adaptive ?

CSI gives more information about the channel and can be used to derive angle of arrival (AoA) of the line of sight component between transmitter and receiver. AoA can be used along with the existing LSTM model for better precision in future.

References

[1] S. Sadowski and P. Spachos, "Rssi-based indoor localization with the internet of things," *IEEE Access*, vol. 6, pp. 30149 – 30161, 2018.

[2] B. Ferris, D. Fox, and N. Lawrence, "Wifi-SLAM using gaussian process latent variable models," IJCAI'07, pp. 2480–2485, 2007.

[3] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," *Proceedings of the sixteenth annual international conference on Mobile computing and networking - MobiCom 10*, 2010.

[4] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *SIGCOMM Comput. Commun. Rev.*, vol. 41, pp. 53–53, Jan. 2011.

[5] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, nsdi'13, (Berkeley, CA, USA), pp. 71–84, USENIX Association, 2013.

[6] S. Kumar, E. Hamed, D. Katabi, and L. Erran Li, "Lte radio analytics made easy and accessible," *SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 211–222, Aug. 2014.

[7] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, (New York, NY, USA), pp. 483–494, ACM, 2014.

[8] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: Mobile phone localization via ambiance fingerprinting," in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, MobiCom '09, (New York, NY, USA), pp. 261–272, ACM, 2009.

[9] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowd-sourcing for indoor localization," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12, pp. 293–304, 2012.

[10] J. P. Wilkinson, "Nonlinear resonant circuit devices," 1990.

[11] X. Wang, L. Gao, S. Mao, and S. Pandey, "Csi-based fingerprinting for indoor localization: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 66, pp. 763–776, Jan 2017.

[12] Y. Lukito and A. R. Chrismanto, "Recurrent neural networks model for wifi-based indoor positioning system," in *2017 International Conference on Smart Cities, Automation Intelligent Computing Systems (ICON-SONICS)*, pp. 121–125, Nov 2017.

[13] M. Ibrahim, M. Torki, and M. Elnainay, "CNN based indoor localization using rss time-series," 06 2018.

[14] X. Chen and Z. Gao, "Indoor ultrasonic positioning system of mobile robot based on tdoa ranging and improved trilateral algorithm," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, pp. 923–927, June 2017.

[15] H. Lv, L. Feng, A. Yang, P. Guo, H. Huang, and S. Chen, "High accuracy vlc indoor positioning system with differential detection," *IEEE Photonics Journal*, vol. 9, pp. 1–13, June 2017.

[16] J. Wang, F. Adib, R. Knepper, D. Katabi, and D. Rus, "Rf-compass: Robot object manipulation using RFIDs," in *Proceedings of the 19th Annual International Conference on Mobile Computing &#38; Networking*, MobiCom '13, (New York, NY, USA), pp. 3–14, ACM, 2013.

[17] X. Wang, Z. Yu, and S. Mao, "DeepML: Deep lstm for indoor localization with smartphone magnetic and light sensors," pp. 1–6, 05 2018.

[18] C. Olah, "Understanding LSTM networks," 2015.

[19] T. Rappaport, *Wireless Communications: Principles and Practice*. 1996.

[20] D. Puccinelli and M. Haenggi, "Multipath fading in wireless sensor networks," *Proceeding of the 2006 international conference on Communications and mobile computing - IWCMC 06*, 2006.

[21] Y. Zhang, Q. Liu, and L. Song, "Sentence-state LSTM for text representation," *CoRR*, vol. abs/1805.02474, 2018.

[22] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *CoRR*, vol. abs/1605.05101, 2016.

[23] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the* 13^{th} *International Conference on Artificial Intelligence and Statistics*, vol. 9, pp. 249–256, PMLR, 13–15 May 2010.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.