Review Article

Bluetooth Low Energy-based Indoor Localization using Artificial Intelligence

Moses Y. Ndebugre[1*], Tülay Yıldırım[2*]

[1] Yıldız Technical University, Orcid ID: 0000-0002-6941-6675, moses.ndebugre@std.yildiz.edu.tr
[2] Yıldız Technical University, Orcid ID: 0000-0001-9993-5583, tulay@yildiz.edu.tr
[*] Correspondence: yirimeah.official@gmail.com; Tel.: (+905350844879)

(First received March 15, 2022 and in final form September 22, 2022)

Reference: Ndebugre, Y. M., & Yıldırım, T. Bluetooth Low Energy-based Indoor Localization using Artificial Intelligence. The European Journal of Research and Development,2(3), 1–15.

Abstract

Bluetooth is one of the several technologies to cater to indoor localization. It has the lowest power consumption and good accuracy performance. In the world of IoT, data from sensors and software help in giving meaning to physical objects connected to the internet.

This paper uses data gathered using Bluetooth Low-Energy sensors in predicting an agent's location in an indoor environment.

We propose a Bluetooth-based model that is divided into two parts: a Convolutional Neural Network(CNN) that trains on data transformed into images and ideas from Game Theory that uses the Markov Decision Process(MDP) to determine the exact location of the agent. The data to image transformation uses the Image Generator for Tabular Data (IGTD) algorithm, which considers the Euclidean distances between the access points in creating the images.

The results show that the CNN trains well on transformed images and offers a solid approach to determining every beacon used for Bluetooth-based indoor localization. After a beacon is found, MDP finds the optimal policy to locate the access point under which the agent lies.

Keywords: Convolutional Neural Networks, Indoor Localization, Bluetooth, Markov Decision Process.

1. Introduction

For numerous applications in the world of Internet-of-things to operate, they rely on knowing the location of agents(agents could be humans, robots, or devices). For example, in logistics, the whereabouts of packages from the point of origin until they get to the final designation are critical in ensuring accountability and reliability. Digital property addressing systems assign unique codes to properties based on their location, and the Global Positioning System(GPS) gives the direction to these properties. Although GPS is overly-implemented for outdoor localization, a combination of crowdsourced Wi-Fi signals and in-built smartphone sensors gives a better performance in power consumption and positioning accuracy (Du, H., Zhang, C., et al., 2018).

In the case of indoor environments, GPS is entirely unused because the walls and other features of buildings heavily attenuate the downlink microwave signals approaching buildings. Hence, the options proposed for indoor localization are short and medium-range technologies such as Ultra-Wideband (Y. Zhang et al., 2022)., Zigbee (Z. Dong, Xu, Zhuang., 2019), Bluetooth (Roshan A. et al., 2018), and Wi-Fi(Liu et al., 2019).

In this paper, we used signals transmitted from Bluetooth Low Energy(BLE) beacons to design a model that determines the location of agents inside a building. Among all the short and medium-range technologies, Bluetooth offers the lowest power consumption (S. Sadowski and P. Spachos., 2018), albeit it does not have the best accuracy. It has a maximum range of 100m and a maximum throughput of 24Mbps (F. Zafari et al., 2019).

The receiver antennas of smartphones measure the signals transmitting from the BLE beacons as the Received Signal Strength(RSS). Similar to others, this data is recorded and then transformed into numerical features. Machine Learning(ML) allows systems to learn patterns in data and make inferences autonomously from the information gathered. Deep Learning is a subset of ML which allows the processing of large quantities of data, and its usefulness is evident in (C. S. Wickramasinghe et al., 2018) for smart-living.

2. Related works

The Bluetooth technology has upgraded with new versions released regularly, and many semiconductor manufacturers include Bluetooth in their products. (Mohammadi et al., 2018) presented a Semi-Supervised Deep Reinforcement Learning model that utilized unlabeled BLE Receiver Signal Strength Indicator(RSSI) data for training a Variational Autoencoder. The Variational Autoencoder learned the best policies that gave the highest rewards from latent variables created from features extraction of the unlabeled data. The proposed model outperformed a supervised learning model regarding average rewards and coverable distance.

In (N. Hernández et al., 2021), Wi-Fi Received Signal Strength (RSS) from various fixed access points in the building are arranged, and by adding an offset of +200 to the RSS, creates images. The authors proposed a Convolutional Neural Network(CNN) architecture consisting of thirteen (13) convolutional layers, thirteen (13) Batch Normalization layers, five (5) ReLU layers, and at the last layer, a Fully Connected + Softmax layer. The authors used Root Mean Square Error(RMSE) to evaluate their model, and the results show the proposed model having the lowest RMSE compared to other pre-trained models like the AlexNet, ResNet18, GoogLeNet, and SVM.

Meta-heuristic algorithms are such that they search and select an adequately good solution for optimization problems. With Particle Swarm Optimization being meta-heuristic, (A. K. Panja et al., 2022) proposed a Binary Particle Swarm Optimization (BPSO) based feature selection procedure that captures the best Access Point(AP) in the location. Different classifiers are designed based on different feature sets, and a neural network with two hidden layers predicts the outcome. The classification accuracy slightly increased after applying BPSO to some selected classifiers: K-Nearest Neighbors(KNN) and Naive Bayes, while the accuracy remained the same for Support Vector Machine(SVM) and Decision Tree(DT). Generally, the authors showed that the proposed model is robust and adaptive in a changing environment.

Environmental factors such as temperature, humidity, and noise affect the positioning accuracy of indoor localization (Byeon, Tae-Woo & Jang, Seong-Yong., 2016). (A. Guidara et al., 2021) came up with three ANN-based architectures using RSSI, temperature, humidity, and Link Quality Indication (LQI) values as input data. In the Centralized Architecture, each neuron in the input layer represents an input variable from the training samples and applied stochastic gradient descent as the optimization technique. In the Distributed Architecture, the network samples were split according to their sources and trained individually, albeit the training was carried out in a central unit to save computational cost. Lastly, they introduced a new input variable to the Centralized Architecture to create the Simplified Architecture that works better when the number of sensors increases. The authors demonstrated that a diverse input dataset outperforms a single dataset type in locations affected by environmental factors.

(Xu et al., 2021) identified adversaries that could harm the updating of fingerprint databases online and thus proposed a model to mitigate this. The authors proposed the BERT-ADLOC model. By assigning adversary samples as negative and reliable fingerprint samples as positive, the model can discriminate between them. The fingerprint samples are filtered out and used to update the previous model. After deploying many malicious beacons, a comparison of the performance between BERT-ADLOC and KNN and Deep Neural Network(DNN) models showed the proposed model having the lowest mean error rate of 1.17.

In this paper, the proposed model uses the Image Generator for Tabular Data (IGTD) algorithm to generate grayscale images from BLE RSSI values. An image characterizes every transmitter Bluetooth beacon, and a CNN trains these images. The algorithm goes through an iterative process to assign pixels and truly capture the dataset features. After identifying a beacon, the agent and the beacon undergo a two-player Game Theory strategy that determines the exact location of the agent in the vicinity of the beacon.

3. Problem Definition

We obtained the dataset from the UC Irvine Machine Learning Repository uploaded by Mehdi Mohammadi and Ala Al-Fuqaha from Western Michigan University (M. Mohammadi et al., 2018). The researchers recorded RSSI values from thirteen (13) BLE beacons with smartphones over a considerate period.

There are two datasets; a labeled set that contains the location, timestamp, and RSSI values, and an unlabeled set that holds just the RSSI values of the beacons. For our simulation, we used the labeled dataset for supervised learning They experimented on the first floor of Waldo Library, Western Michigan University, a space with an area of 200ft by 180ft. Fig. 1 below shows the layout of the building. They experimented on the first floor of Waldo Library, Western Michigan University, a space with an area of 200ft by 180ft (M. Mohammadi et al., 2018). The floor is divided into square blocks of 10ft by 10ft. We will refer to these square blocks as Access Points(APs) going forward. A combination of Alphabets(vertical) and numbers(horizontal) give names to the APs. For instance, AP S3 is in the vicinity of beacon four(4).

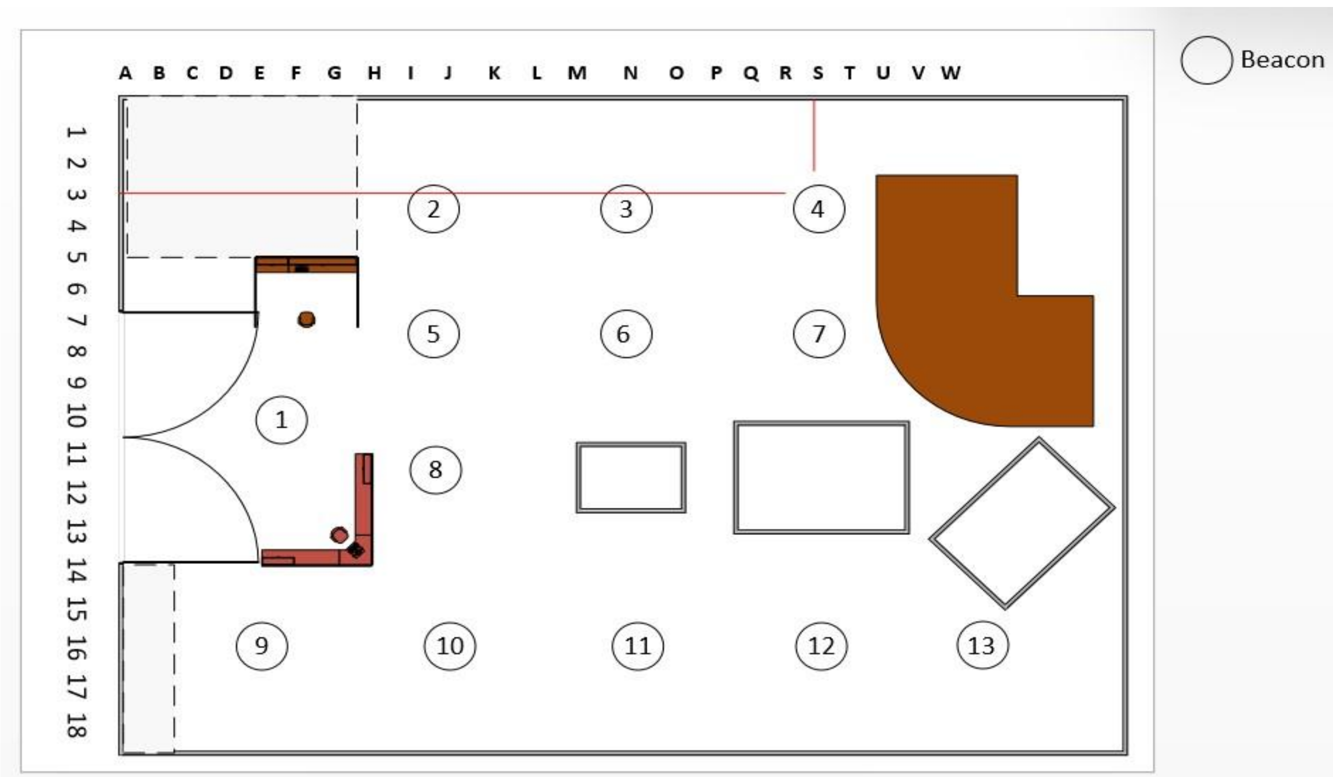
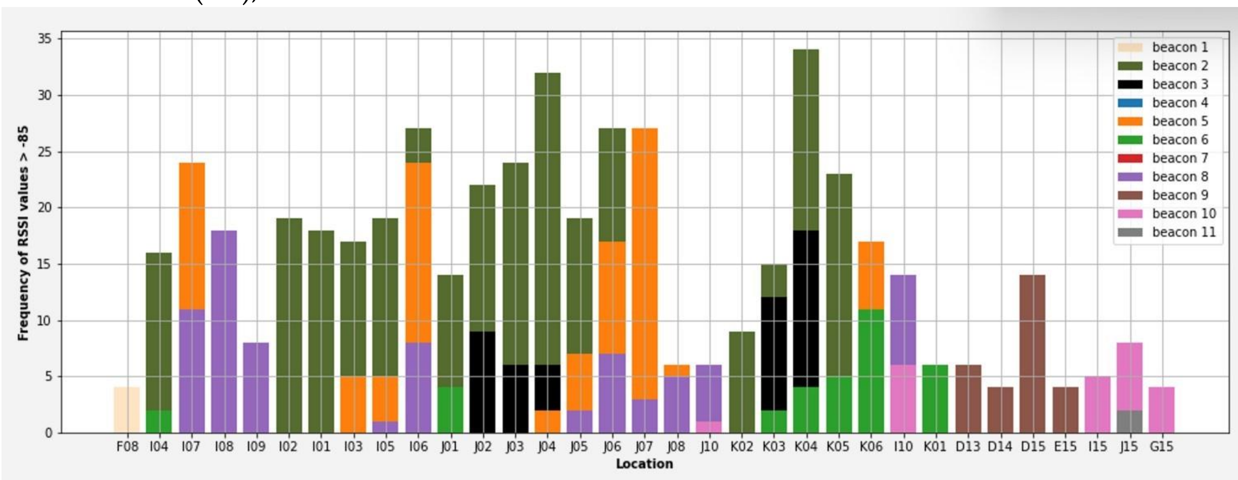


Fig. 1 layout of beacons

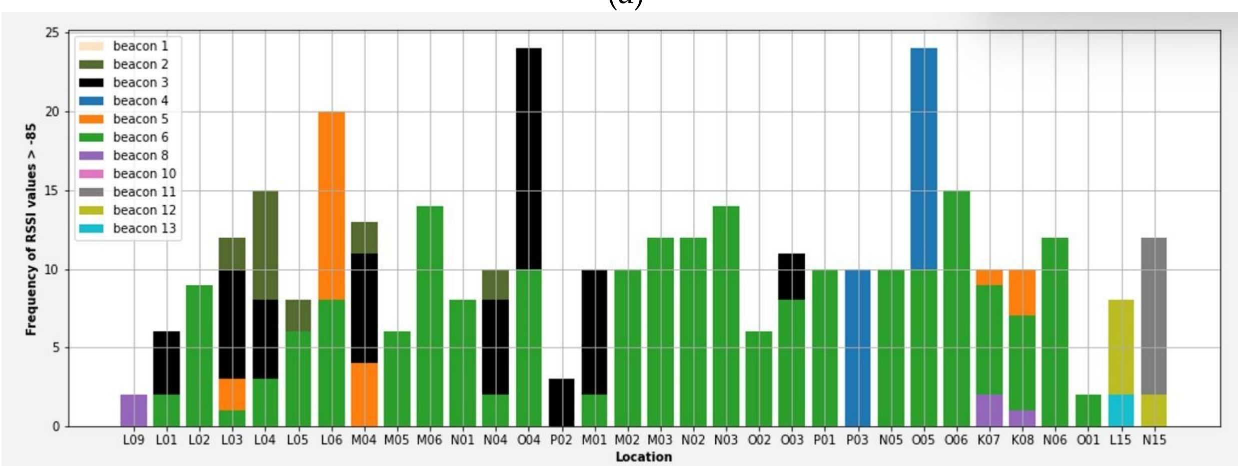
The RSSI values range from the best signal strength of -56dBm to the weakest signal strength of -200dBm. The figure below shows the distribution of good signals at different APs for all beacons. The RSSI is expressed mathematically as

$$RSSI = -10\log(d) + A. \quad (1)$$

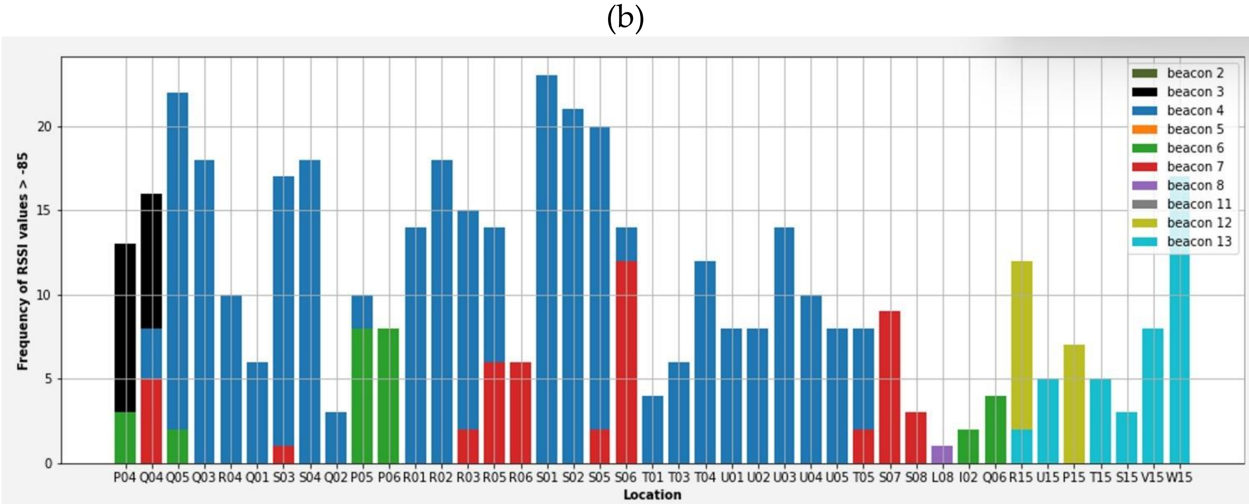
where n is the path loss exponent, d represents the distance between Transmitter (Tx) and Receiver (Rx), and A is the RSSI value at a reference distance from the receiver.



(a)



(b)



(c)
Fig. 2 Distribution of RSSI

4. System Design

The proposed design is divided into two parts. Part A introduces a brief description of the IGTD algorithm, and the Convolutional Neural Network supervised learning. Part B introduces the Normal Form of the static game representation and the Markov Decision Process.

A. Convolutional Neural Network

Convolutional Neural Networks are notably successful in image and video recognition. This approach utilizes the advantages of CNN and converts the BLE RSSI dataset into grayscale images using the Image Generator for Tabular Data (IGTD) algorithm instead of a conventional vector representation of the dataset (Zhu, Y., Brettin, T., Xia, F. et al., 2021). For each beacon, the algorithm takes the APs and assigns them features, ensuring that similar features are close to each other. The algorithm went through over 5000 iterations before converging at a runtime of 312 seconds.

The algorithm is summarized as follows; let X denote an M by N tabular data. xi,: is the ith feature, x(:,j) is the jth sample, and xi,j is the element in the ith row and jth column. Calculate the pairwise distance between features using Euclidean distance and transform each xi,: into Nr by Nc image where Nr x Nc = N. Rank the pairwise pixel distances ascendingly, forming an N by N matrix denoted by R, in which ri,j is the element at the ith row and jth column of R. Calculate the pairwise distance between pixels using Euclidean distance and transform each xi,: into Nr by Nc image where Nr x Nc = N. Rank the pairwise pixel distances ascendingly, forming an N by N matrix denoted by Q, in which qi,j is the element at the ith row and jth column of Q. An error function is defined as

$$err(R,Q) = \sum_{i=2}^{N}\sum_{j=1}^{i-1} diff\left(r_{i,j}, q_{i,j}\right). \qquad (2)$$

In (2), the function measures the difference between ri,j and qi,j, and if the error is high, it reorders the features so that similar features are close to each other until the error is minimized. The diagram in Fig. 3 shows the error decreasing exponentially after several iterations.

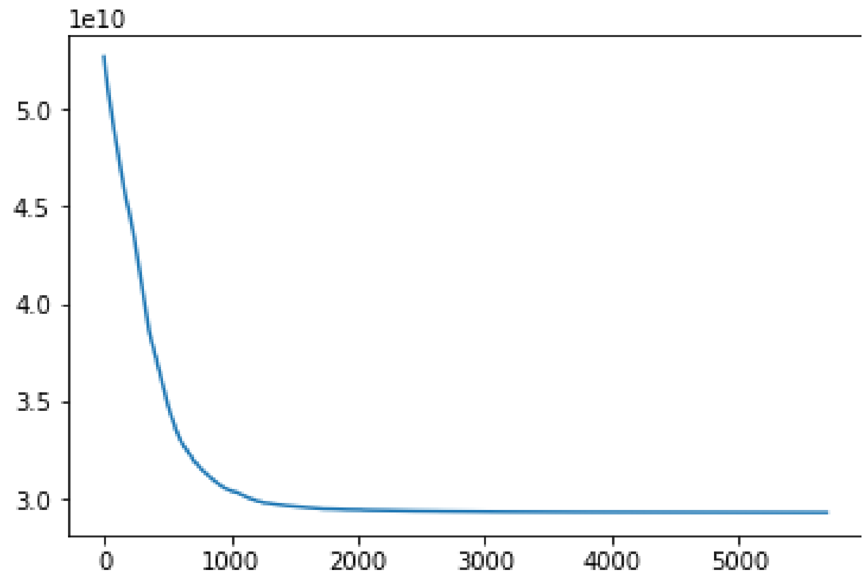
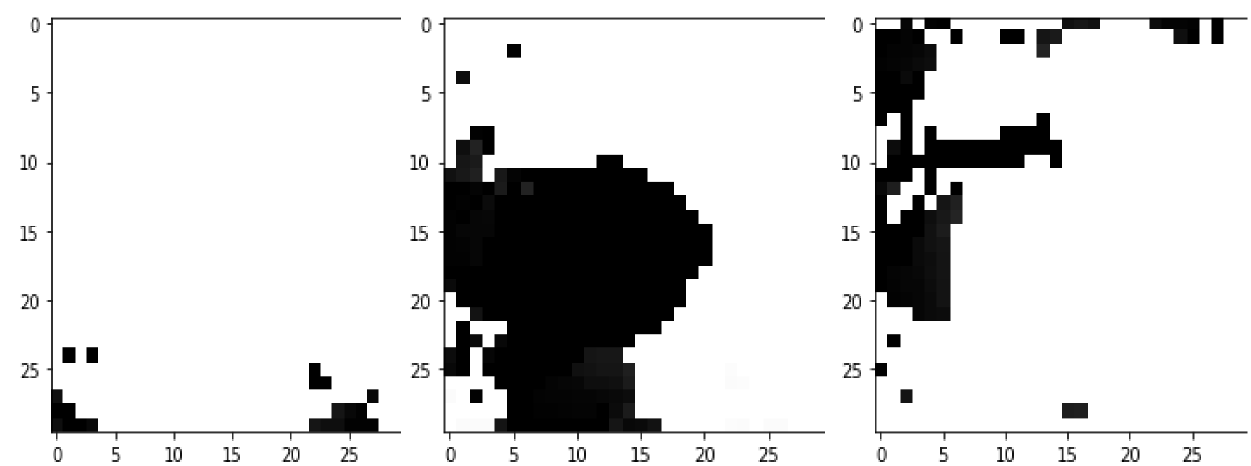
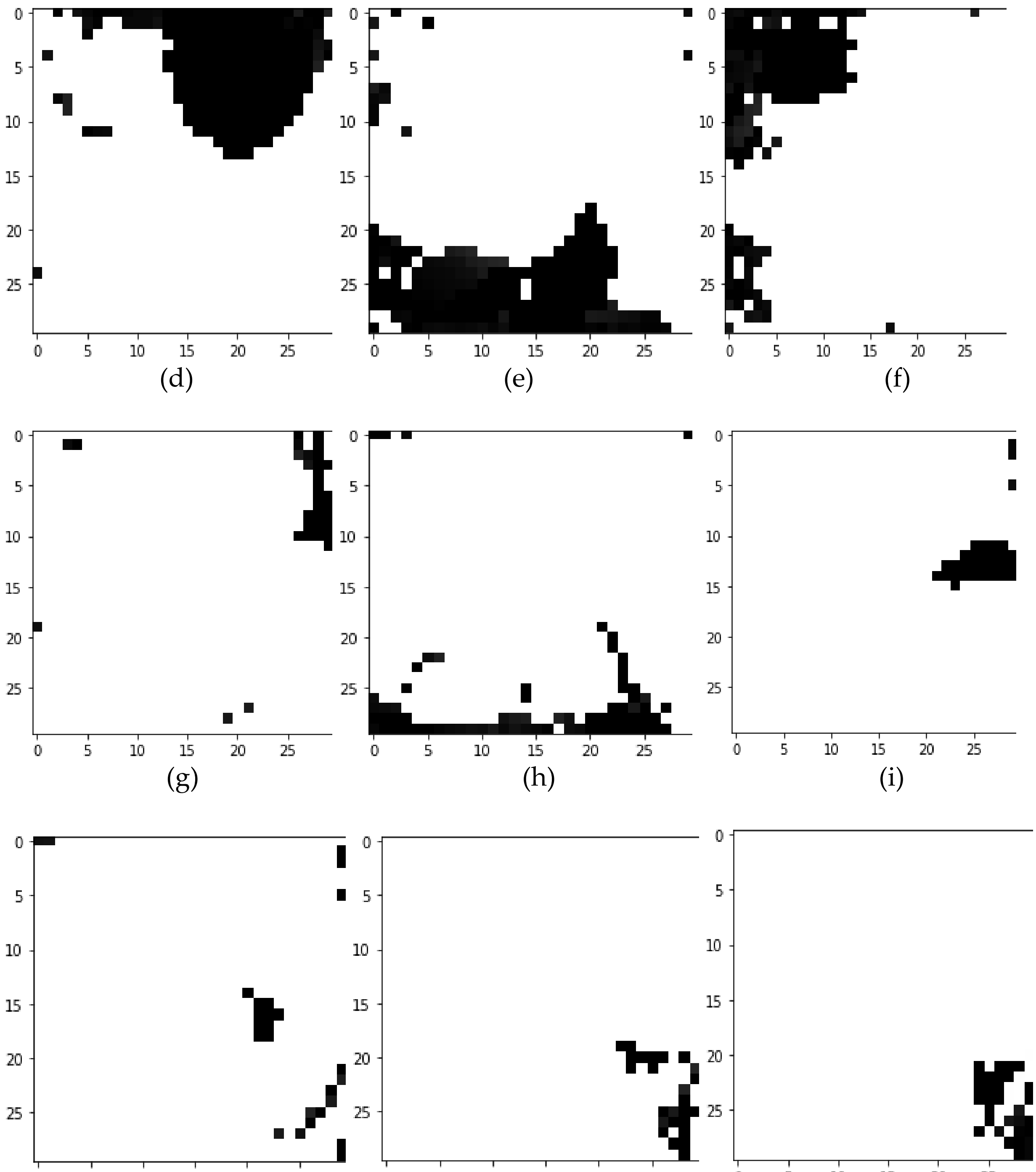


Fig. 3 Change in error during optimization

Most deep layer networks perform very well when the input image data has a 224x224 dimension, but in place of this, the dataset is transformed into a 30 by 30 dimensional matrix because of the IGTD algorithm's high GPU computational cost. The RSSI values are offset by -3,+3,-2,+2,-1, and +1 to increase the number of training and test examples. Fig. 4 below represents all the beacons in image form where (a) to (m) are images for beacon 1 to beacon 13.

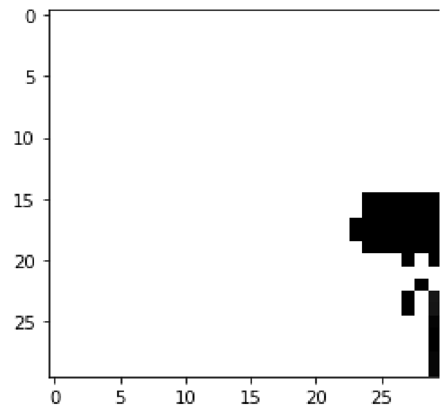

(a) (b) (c)



(d) (e) (f)

(g) (h) (i)

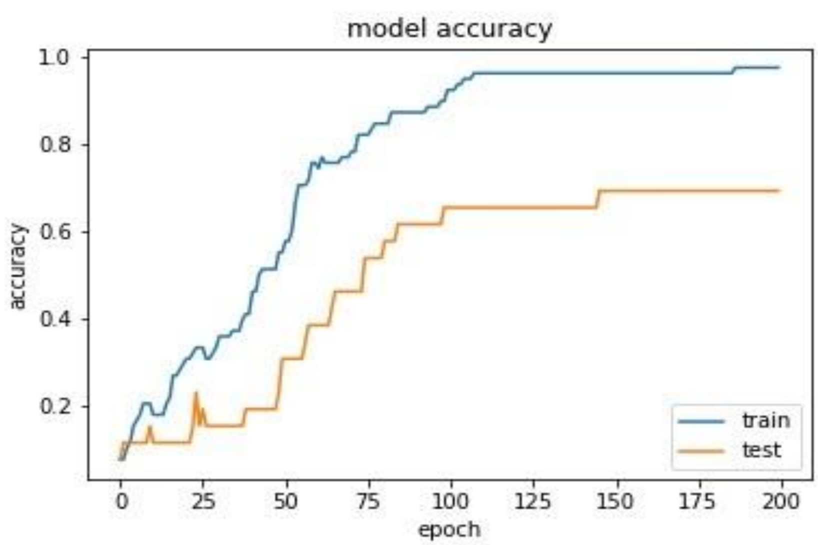
(j) (k) (l)



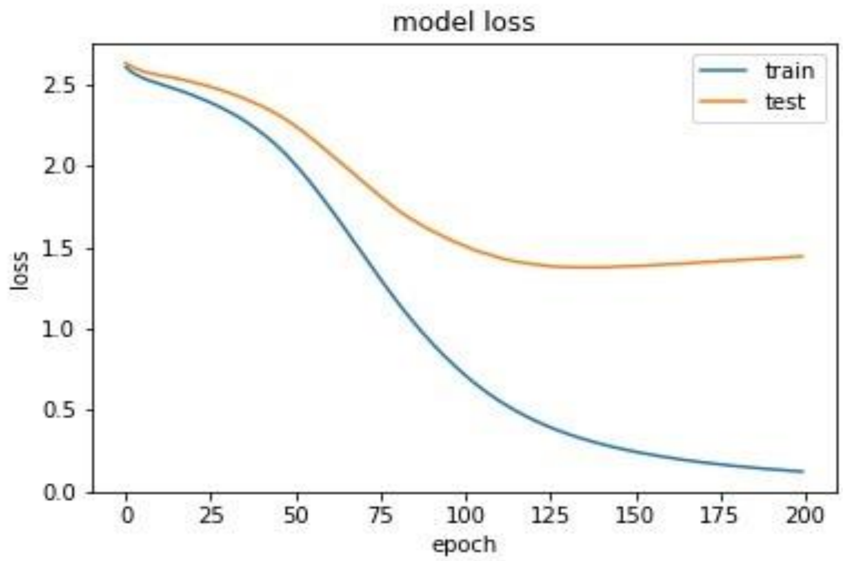
(m)

Fig. 4. Image form of the beacon data

The network has two(2) convolutional layers, with each layer followed by a Rectified Linear layer (ReLU) and a max-pooling layer. The first 2D Conv layer has eight(8) kernels of size four(4) and the second 2D Conv layer has sixteen(16) kernels of size four(4). Each Conv layer undergoes a stride of one(1). For max-pooling, the first layer has a stride of eight(8) with a pool size of eight(8), while the second has a stride of four(4) with a pool size of four(4). The network is represented as follows; Conv layer → ReLU → Max Pool → Conv layer → ReLU → Max Pool → Flatten → Dense. Since the architecture has thirteen(13) beacons, the dense layer has thirteen(13) neurons so that every neuron can learn about a corresponding beacon. The model is trained with the Adam optimization with a learning rate of 0.001. Also, for a multi-classification situation with thirteen(13) different classes, the model uses the categorical cross-entropy loss function to ensure that the input examples belong to just one class. Fig. 5 (a) and (b) show the model accuracy and loss respectively.



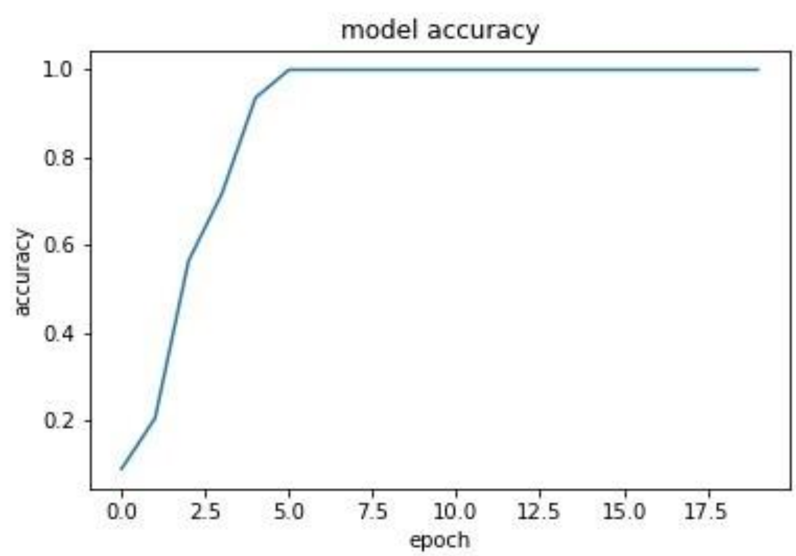
(a)



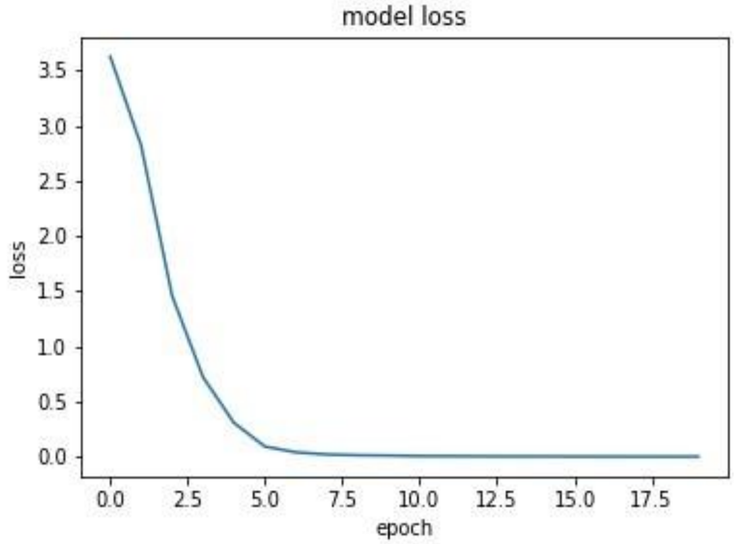
(b)

Fig. 5 ConvNet Model accuracy and loss plots.

Deeper networks require more parameters to be trained but give better results. Transfer Learning enables the reuse of pre-trained models, so we consider the ResNet50 to improve our model. The Residual Neural Network (ResNet) offers the opportunity to train deeper layers and is easier to optimize and gain better accuracy (K. He et al., 2016). The simple ConvNet model had just 885 parameters to train, whereas the ResNet50 had 6107413 parameters to train. Fig. 6 (a) and (b) below are the improved model accuracy and loss plot respectively after 20 epochs.



(a)



(b)

Fig. 6 ResNet50 Model accuracy and loss plots.

We have successfully trained the convolutional neural network model, which identifies the beacon of interest when its corresponding image is fed into the network. The next stage requires locating which of the access points within a beacon's coverage area the beacon is situated. We analyzed this by considering Nash's Equilibrium and finding an optimal policy that arrives at the access point of interest using Markov's Decision Process.

B. Reinforcement Learning

Game Theory is the field of study of strategic interactions between rational agents in an environment. It was developed in the 1950s and has seen its application across many fields such as politics, economics, and science (C. A. Holt and A. E. Roth., 2004). If we think about our environment as dynamic where beacons and agents play strategies, we can apply concepts from Game Theory. From Fig. 2., we see that beacon 1 covers six(6) possible APs; I08, F08, I07, I04, I09, and L09. The agent is hidden in one of these APs, and beacon 1 has to find it. Beacon 1 receives a payoff of 'x' should it find the agent and a utility of 0 if it does not. The agent receives a utility of 'y' if not found, and a utility of 0 if it does not, where x and y > 0. The Normal Form of this game is represented below, and the same concept applies to all of the beacons. The row player is the agent while the column player is beacon 1. The Nash equilibrium in this game exists when beacon 1 and the agent are at the same AP.

Table. 1 Beacon 1 Normal Form game .

Agent	Beacon 1					
	I08	I09	D13	D14	D15	F07
I08	y, x	0, 0	0, 0	0, 0	0, 0	0, 0
I09	0, 0	y, x	0, 0	0, 0	0, 0	0, 0
D13	0, 0	0, 0	y, x	0, 0	0, 0	0, 0
D14	0, 0	0, 0	0, 0	y, x	0, 0	0, 0
D15	0, 0	0, 0	0, 0	0, 0	y, x	0, 0
F07	0, 0	0, 0	0, 0	0, 0	0, 0	y, x

For the system to devise a policy that yields the highest utility, Reinforcement Learning is ideal. Reinforcement Learning is a subset of Machine Learning that allows machines to decide the best way to behave for performance enhancement. Markov Decision Process(MDP) is a Reinforcement Learning algorithm whereby agents decide the best policy based on its current state in a stochastic environment (M.L. Littman, 2001). The underlying property of MDP is that a "state" at day t+1 depends only on the "state" at day t. It contains a set of finite states X, transitional probabilities P, a finite set of actions, and a reward function.

I. Environment

Every beacon has its unique environment. Beacon 1 is a 2x4 matrix environment, and every cell in the matrix represents a state. Beacons 2 to 13 are 6x7, 6x6, 7x7, 6x6, 7x7, 3x4, 4x4, 2x4, 3x3, 2x2, 2x3, 3x3 matrixes respectively.

II. State

A state is a cell in the matrix from which an agent moves. Every state represents a particular Access Point, and an agent can move up, down, right, or left from a state depending on the optimal policy.

III. Value Iteration

The algorithm that computes the optimal policy for the MDP by calculating the utilities of each state is called Value Iteration. It is executed with the Bellman equation iteratively, which is expressed as:

$$U(s) = R(s) + \gamma \underbrace{max}_{a \in A(s)} \textstyle\sum_{s'} P(s'|s,a)U(s'). \quad (3)$$

IV. Policy

The policy is the solution for the MDP, and it states which action is optimal for every state. After several iterations, we obtain the optimal policy for every beacon.
In Fig. 7, let us assume the current location of the agent is at State I08 for beacon 1.There is no movement when the beacon's tracking collides with the wall. The current location has a reward of +1, and the state close to it is assigned -1 as a punishment measure.

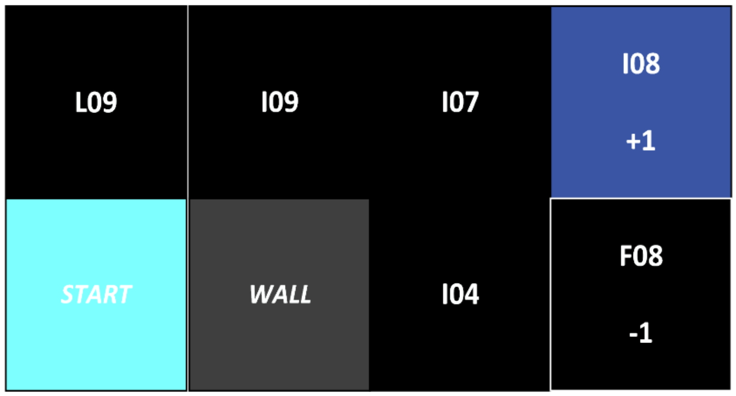


Fig. 7 Environment for beacon 1

the initial random policy to get to I08 is

| *Up* | *Left* | *Up* | *+1* |
| *Right* | *WALL* | *Up* | *-1* |.

The policy is optimized during policy iteration, and the optimal policy is found to be:

| *Right* | *Right* | *Right* | *+1* |
| *Up* | *WALL* | *Left* | *-1* |.

The same policy iteration is performed for all other beacons per their environment type. In this stochastic environment, every beacon learns the optimal policy on how to traverse to the point of interest. It means that the agent will always be located at its AP.

5. Conclusion

In this paper, we proposed a Bluetooth-based method for indoor localization that takes advantage of computer vision by using data transformed into images as examples for training a Convolutional Neural Network. Besides the CNN, this paper also looked at reinforcement learning in the form of the Markov Decision Process. After determining the Nash equilibria for a beacon's game with the agent, we used the MDP to find the optimal route to get to the agent moving from state to state. This paper shows that by creating a distributed system whereby we can determine the beacon of interest, the number of access points we have to deal with reduces, hence, finding an optimal policy becomes easier.

References

Du, H., Zhang, C., Ye, Q. et al. (2018). A hybrid outdoor localization scheme with high-position accuracy and low-power consumption. J Wireless Com Network.

Y. Zhang, Y. Chu, Y. Fu, Li, Y. Song. (2022). UWB Positioning Analysis and Algorithm Research. Procedia Computer Science, vol. 198 ,pp. 466-471.

Z. Dong, Xu, Zhuang.(2019). Research on ZigBee Indoor Technology Positioning Based on RSSI. Procedia Computer Science, vol. 154, pp. 424-429.

Roshan A., Deepak V., Dinesh B. (2018). BLoc: CSI-based accurate localization for BLE tags. CoNEXT '18: Proceedings of the 14th International Conference on emerging Networking Experiments and Technologies, pp. 126–138.

Liu, F., Liu, J., Yin, Y. et al. (2019). Survey on WiFi-based indoor positioning techniques. IET Communications, vol. 14, pp. 1372-1383.

S. Sadowski and P. Spachos. (2018). RSSI-Based Indoor Localization With the Internet of Things. IEEE Access, vol. 6, pp. 30149-30161.

F. Zafari, A. Gkelias and K. K. Leung. (2019). A Survey of Indoor Localization Systems and Technologies. IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2568-2599.

C. S. Wickramasinghe, D. L. Marino, K. Amarasinghe and M. Manic. (2018). Generalization of Deep Learning for Cyber-Physical System Security: A Survey. IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society. pp. 745-751.

M. Mohammadi, A. Al-Fuqaha, M. Guizani and J. -S. Oh. (2018). Semisupervised Deep Reinforcement Learning in Support of IoT and Smart City Services. IEEE Internet of Things Journal, vol. 5, no. 2, pp. 624-635.

Zhu, Y., Brettin, T., Xia, F. et al. (2021). Converting tabular data into images for deep learning with convolutional neural networks. Sci Rep 11, 11325.

M. Mohammadi, A. Al-Fuqaha, M. Guizani and J. -S. Oh. (2018). Semisupervised Deep Reinforcement Learning in Support of IoT and Smart City Services. IEEE Internet of Things Journal, vol. 5, no. 2, pp. 624-635.

N. Hernández, I. Parra, H. Corrales, R. Izquierdo, A. L. Ballardini, C. Salinas and I. García. (2021). WiFiNet: WiFi-based indoor localisation using CNNs. Expert Systems with Applications, vol. 177.

A. K. Panja, S. F. Karim, S. Neogy and C. Chowdhury. (2022). A novel feature based ensemble learning model for indoor localization of smartphone users. Engineering Applications of Artificial Intelligence, vol. 107.

Byeon, Tae-Woo & Jang, Seong-Yong. (2016). A Study on the Technological and Environmental Factors Affecting the Accuracy of Beacon Based Indoor Positioning System. Journal of the Korea Society for Simulation, vol. 25.

A. Guidara, G. Fersi, M. B. Jemaa, F. Derbel. (2021). A new deep learning-based distance and position estimation model for range-based indoor localization systems. Ad Hoc Networks, vol. 114.

Xu Sun, Haojun Ai, Jingjie Tao, Tan Hu, Yusong Cheng. (2021). BERT-ADLOC: A secure crowdsourced indoor localization system based on BLE fingerprints. Applied Soft Computing, vol 104.

K. He, X. Zhang, S. Ren and J. Sun. (2016). Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 770-778, doi: 10.1109/CVPR.2016.90.

C. A. Holt and A. E. Roth. (2004). The Nash equilibrium: A perspective. Proc. Nat. Acad. Sci. USA, vol. 101, no. 12, pp. 3999-4002.

M.L. Littman. (2001). Markov Decision Processes. International Encyclopedia of the Social & Behavioral Sciences, pp 9240-9242.