RSSI-Based Localization of IoT Devices Using Machine Learning Approaches

Benjamin He[1], Yan Han[1], Samer Hanna[1], Danijela Cabric[1]

[1] Electrical & Computer Engineering Department, University of California, Los Angeles

Introduction

The **Internet of Things (IoT)** is the idea of connecting all devices to the internet and each other. Interest in this topic has been rapidly growing in recent years due to the countless applications it provides including: **smart homes & architecture, intelligent traffic management, military surveillance, natural habitat monitoring, precision agriculture, industrial process control**. In many of these applications, it is important to locate the position of a particular device in the IoT network. This can be done using **wireless network localization**, which is the focus of this project.

- Range-free network localization often utilizes **received signal strength indicator (RSSI)**, which is a measure of the power of a device's received signal from another device.
  - Commonly used as the input for range-free localization in wireless networks.
- **Anchor nodes,** devices already aware of their position, are used as reference points for unknown devices to receive RSSI values from to localize that unknown device.
- There is often a lot of **noise** in RSSI measurements due to the environment.
  - **Support vector machines (SVM)** or **neural networks (NN)** are machine learning algorithms that may provide effective solutions for minimizing that noise for more accurate IoT localization.

Goal

To evaluate the feasibility of using machine learning methods--SVMs and NNs—for localization within an IoT wireless network for potential use in real-world applications.

Simulation Set-up

- Both machine learning localization algorithms are tested in MATLAB simulations (Figure 1):
  - Square **20 x 20 m²** area
  - **1 anchor node in each corner** of area
- Data points are randomly generated within the localization area (blue circles in Figure 1).
  - These data points consist of their **locations** and their **RSSI values** with **noise** and **shadowing**.
  - **80%** of points used for **training** the algorithms
  - **20%** used for **testing** the trained algorithms.

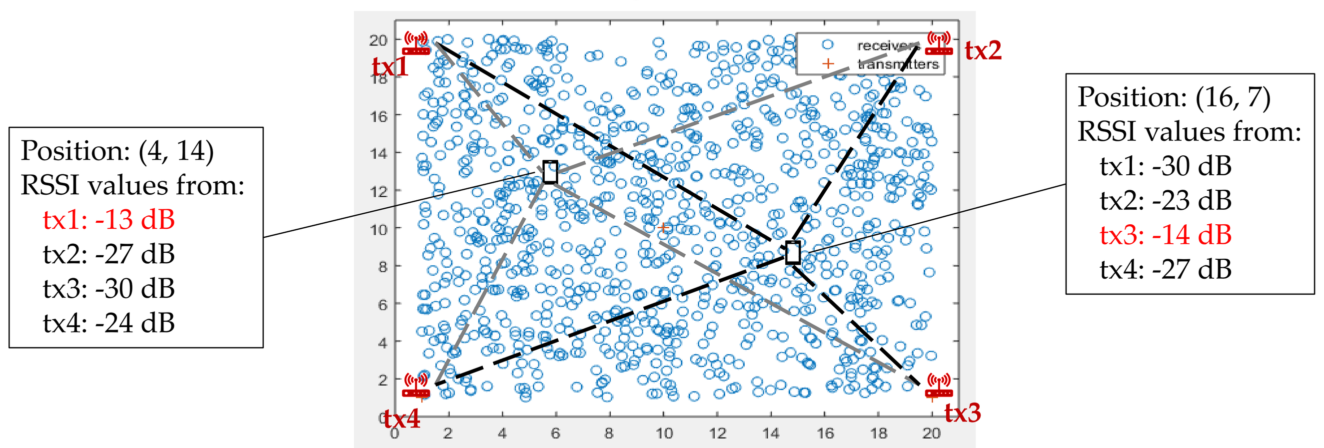

Figure 1: Visualization of Datapoint in Simulation Set-up

SVM-based Algorithm

- Using the training points, SVMs are recursively trained to **divide the localization area into halves** in x and y directions until the desired number of SVMs are obtained (Figure 2).

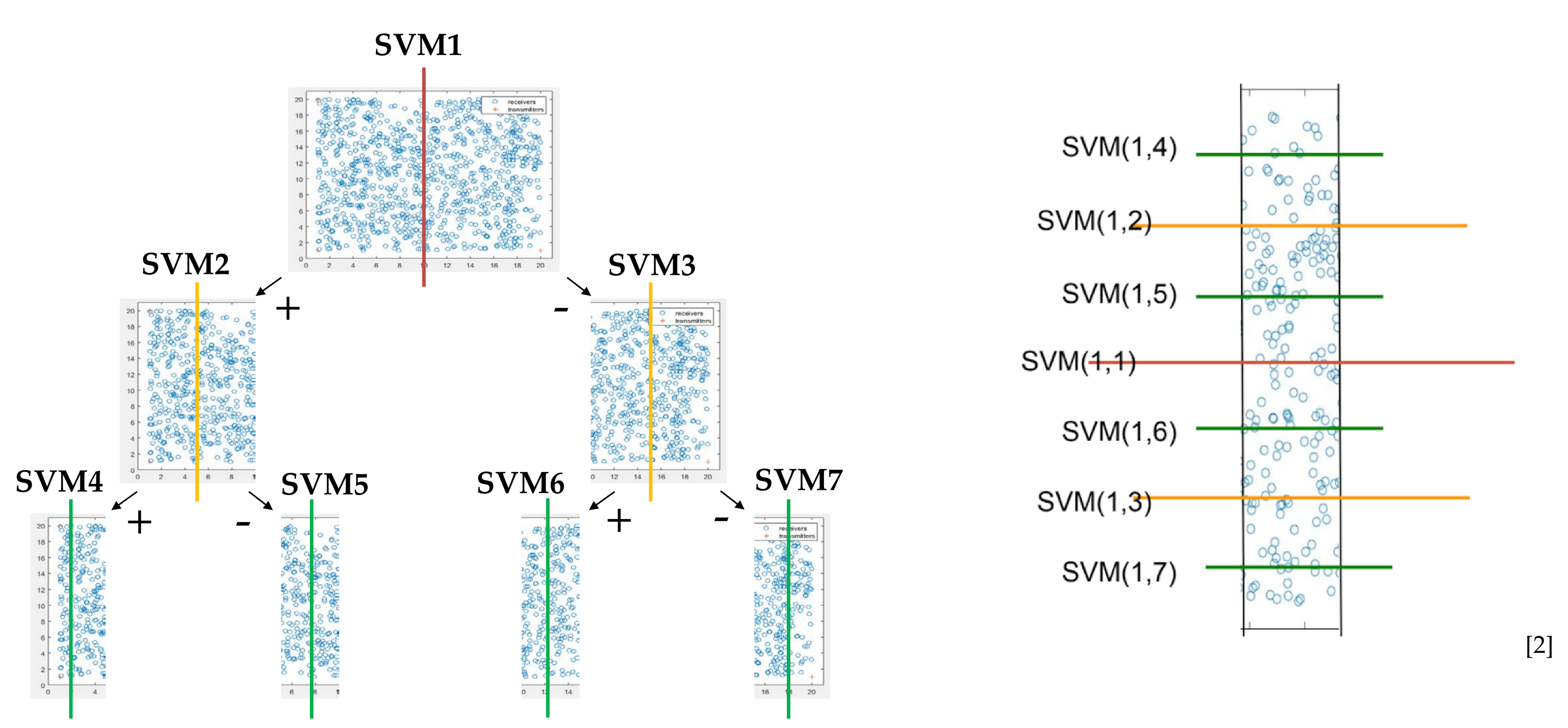

Figure 2: SVM Binary Search Tree

- Once all SVMs are trained, test points are localized by traversing a **binary search tree of the SVMs** (Figure 3) for both x and y directions until arrival at a leaf of the tree, which will indicate the respective estimated x and y coordinates of the test point.

NN-based Algorithm

- The training points are fed to a **multilayer perceptron (MLP)**—a type of neural network.
- Multiple multilayer perceptrons were tested with varying numbers of **hidden layers** and **layer sizes** (single hidden layer example shown below in Figure 3) to determine best architecture.
- The transfer function for each hidden layer(t1) is **tangent-sigmoid** (tansig) and the transfer function for the output layer (t2) is **linear** (purelin).

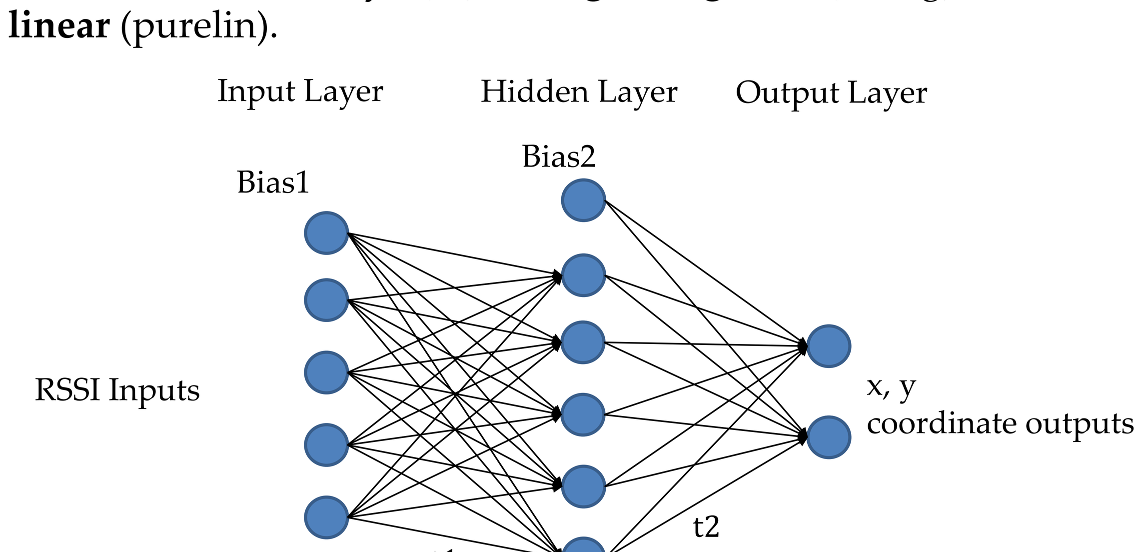

Figure 3: Example MLP Structure

$$t1(input1, input2, input3, input4) = \tanh(\theta_{1,1} * input1 + \theta_{2,1} * input2 + \theta_{3,1} * input3 + \theta_{4,1} * input4 + \theta_{5,1} * Bias1)$$

$$t2(hid1, hid2, hid3, hid4, hid5) = \theta_{1,2} * hid1 + \theta_{2,2} * hid2 + \theta_{3,2} * hid3 + \theta_{4,2} * hid4 + \theta_{5,2} * hid5 + \theta_{6\,2} * Bias2$$

- **Scaled conjugate gradient backpropagation** is used to adjust the network weights (θ).
- The cost function used was **mean-square error** and **regularization** was added to prevent overfitting.
- To localize test data, each test point's RSSI values are simply fed to the trained neural network, and the network calculates and outputs its estimated x and y coordinates.

Results & Discussion

SVM Algorithm

- The localization accuracy of the SVM-based algorithm was tested using different numbers and layouts of anchor nodes as well as varying training set sizes and amounts of trained SVMs.

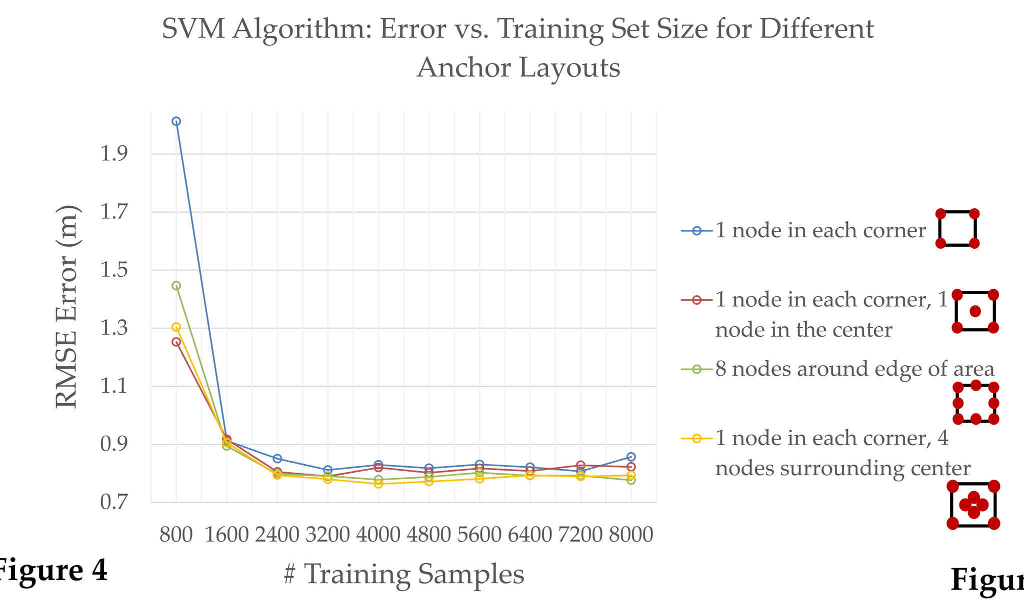

Figure 4

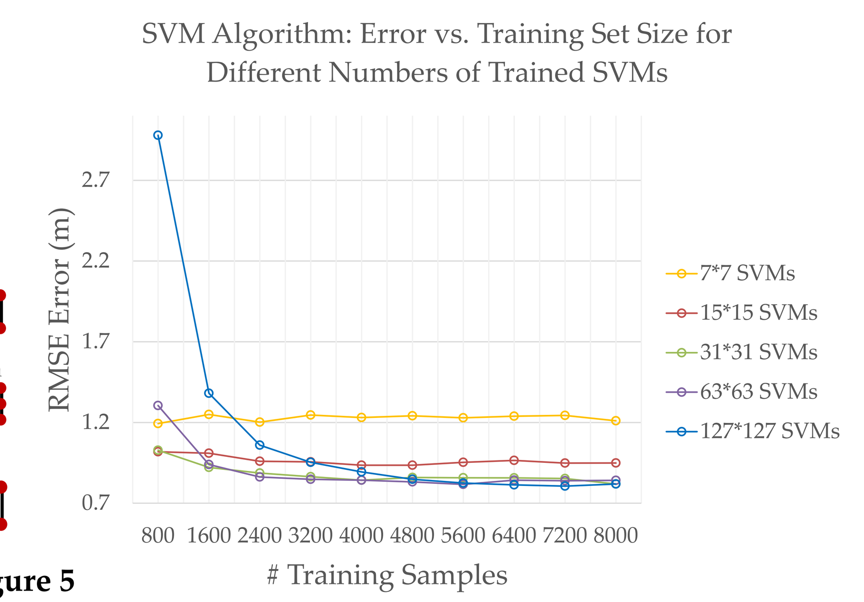

Figure 5

- The **number** and **layout of anchor nodes** does not seem to significantly affect the performance of the SVM-based localization algorithm.
- Overall, **larger training sets** and **more trained SVMs** produce **the best performance**.
- However, a **small training set** with a **large number of SVMs** suffer **significant error** due to the inability for all the SVMs to be trained properly.

NN Algorithm

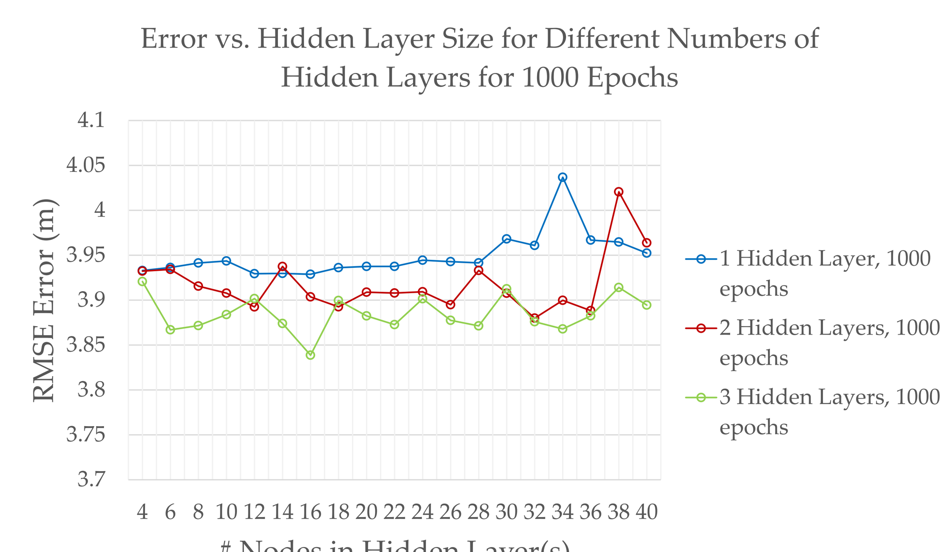

Figure 6a

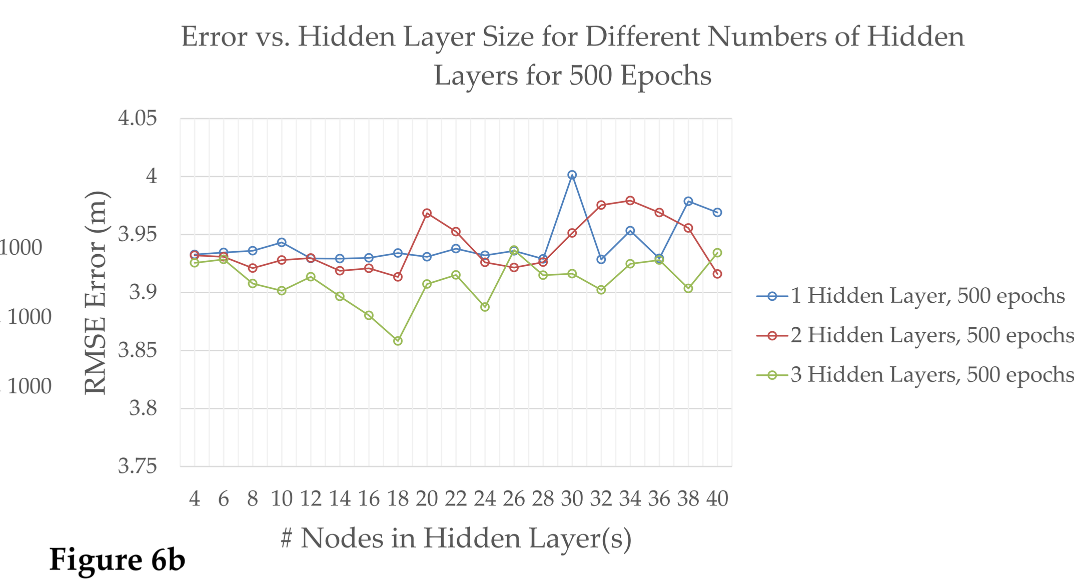

Figure 6b

- Figures 6a, 6b, and 7 show data collected to determine the optimal architecture of the MLP.
- The number of nodes in the hidden layers have negligible effect on the neural network's localization accuracy.
- Increasing the hidden layer count appears to improve localization accuracy, though not significantly.

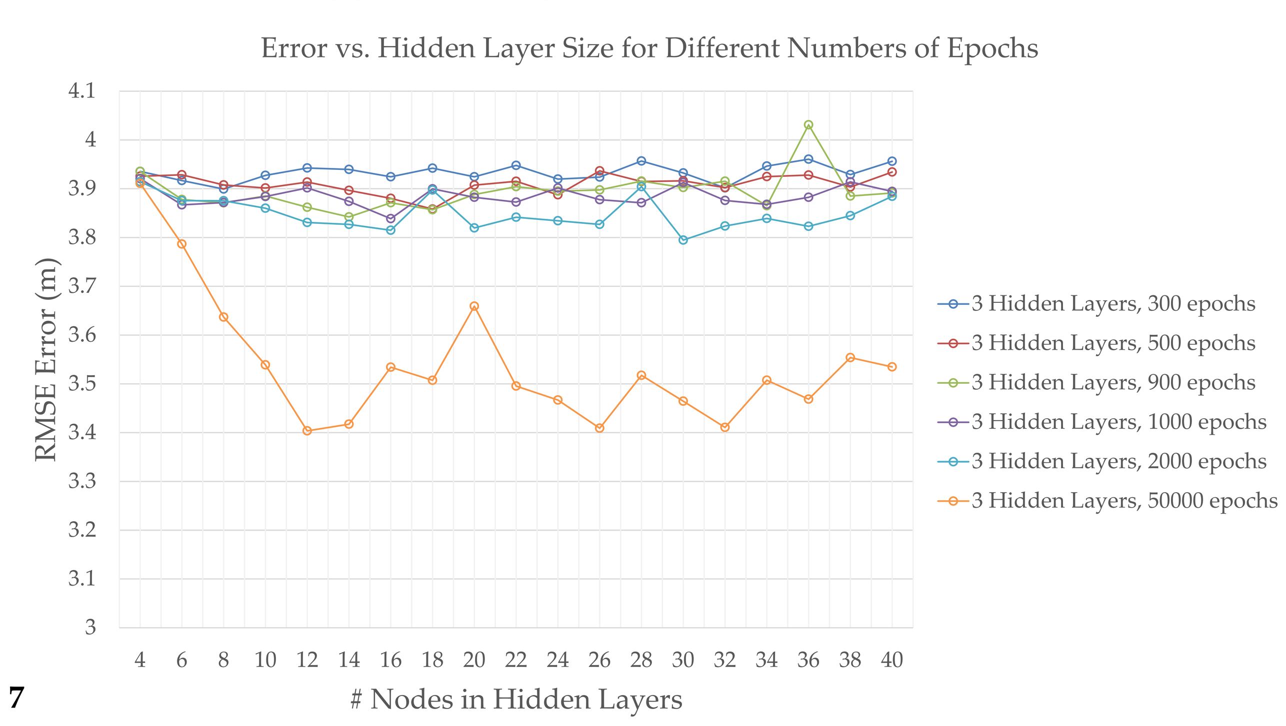

Figure 7

- Because scaled conjugate gradient optimizes the MLP iteratively, running the optimization with more epochs appears to produce a significant improvement on localization accuracy.
- However, it is expected that once the optimization is able to reach the global minimum of the cost function, increasing epochs will no longer improve accuracy.

Conclusions & Future Work

- The SVM-based algorithm reached an accuracy of approximately **0.8 m in a 20 x 20 m² area**.
  - Increasing the number of SVMs improves the localization accuracy of the algorithm, given a sufficiently large training set.
- The NN-based algorithm reached an accuracy of approximately **3.4 m in a 20 x 20 m² area**.
  - The NN-based algorithm is in the process of development--testing is still being done to determine the optimal architecture of the neural network.
  - Increasing the number of epochs and the number of hidden layers seems to improve localization accuracy, though more testing will be done to determine constraints on these parameters.
- Both SVM and NN algorithms will be further tested using training and test data points excluding noise to examine localization performance in ideal conditions.
- These two algorithms will also be compared to other model-based wireless network localization methods.

Acknowledgements

I would like to thank the CORES Lab, the 2018 Summer Undergraduate Scholars Program, the UCLA EE Fast Track Program, and the UCLA Electrical & Computer Engineering Department for providing this valuable summer research opportunity for me.

References

[1] Y. Han, S. Chaudhari, D. Cabric, "RSSI-Based Localization in Wireless Sensor Network Using Support Vector Machines," in 2018 ECE Annual Research Review. 2018.

[2] A. A. Abdallah, S. Saab, Z. M. Kassas, "A machine learning approach for localization in cellular environments," in 2018 IEEE/ION Position, Location, and Navigation Symposium. IEEE 2018. IEEE Conference on, Apr. 2018.