

Radio Tomographic Water Flow Detection

Jason Fong
Computer Science Department, California Polytechnic State University
San Luis Obispo, United States of America
jfong27@calpoly.edu

I. INTRODUCTION

Radio tomography is the imaging of objects by the analysis of radio waves that travel through a room. In the case of this study, the source of said radio waves originate from an ordinary Wi-Fi router in the form of Channel State Information (CSI). CSI describes the propagation of a signal from the router to listening devices, which is affected by scattering, fading, and reflection of the wireless signal as it travels and interacts with the environment. Its intended purpose is for correcting channel attenuation and providing more reliable wireless communication. However, we will take advantage of the information rich data to draw conclusions about the environment. Water has the ability to scatter radio waves and affect the CSI data received [1]. Whether or not the amount of channel disturbance caused by a running shower and sinks is enough to distinguish between an empty room or not will be explored.

II. BACKGROUND

Previous works have used CSI data to successfully train an artificial neural network to differentiate between six different people with 86% accuracy [2]. This study will add to those findings and reinforce the efficacy of the Wi-Fi router, cell phone, and neural network methodology. By carrying out this study, the upper limits of this technique's accuracy may be more precisely defined. Establishing this range of ability will also act as an indicator as to the privacy concerns and other ethics pertaining to the misuse of this technology.

On the subject of ethical concerns in technology, few areas of study have garnered more attention than [artificial] neural networks. A neural network is an amalgamation of various machine learning algorithms and techniques used to loosely simulate the neural network of a biological organism's brain. The neural network is trained on input data to "learn" how to make intelligent predictions on data, and can then be used to predict the classification of new data (e.g. can the radio waves travelling through this room indicate whether the room contains water or not?).

III. EXPERIMENTAL DESIGN

A. Data Collection

The goal of my experimentation was to collect CSI data that could potentially train a neural network to determine whether or not the Wi-Fi router and cell phone were in the presence of running water. The experiments were designed to have varying amounts of running water in order to determine the sensitivity of the system. Other experiments were included to possibly throw off the neural network, and if successful, give insight as to the features that the neural network is training on. For example, if the neural network found difficulty discerning between running water and standing water in the room (figure 2c), it could potentially mean that the neural network does not have the ability to determine if water is flowing from the CSI data alone. It could mean that it was instead simply training on the amount of water in the room as conveyed by the data. All six experiments were performed in two separate bathrooms to help control for other variables.

- 1) *Empty room: Control*
- 2) *Faucets running: Small amount of running water.*
- 3) *Shower running: Medium amount of running water*
- 4) *Shower and faucets running: Maximum amount of running water feasible.*
- 5) *Standing water in room: Distinction between still and running water*
- 6) *No water, human subject pacing in room: Distinction between water based organisms and pure water*

B. Motivation

The major motivation in my experiments is to contribute in discovering the full potential of consumer radio tomography. Specific motivations or use cases of detecting water flow would be home protection; detecting a leak in your home or detecting flow when there should not be could prove to be extremely useful.

C. Data Preprocessing

The CSI captured by the utilities is stored as a binary pcap file, a format not suitable for use in a neural network created with Python. Therefore, the binary file must be parsed into a csv file which can then be converted into a png image. Inspection of the csv reveals that 200 of the 256 total columns are constant throughout the experiment. These represent channels which are not used and do not contain any useful information. The remaining 56 columns will be used to create the image for model training, an example of such an image is seen in figure 1.



Fig. 1: Example image used for training/prediction

IV. IMPLEMENTATION

A. Data Collection

The necessary tools for data collection are a Wi-Fi router to emit CSI data and a device to capture the packets of data. Requirements for the router are the availability of a 5GHz band, the ability to set a fixed channel, and the option for a 20Mhz broadcast width. The channel and bandwidth options are necessary because the data capture utility filter is only able to capture the CSI data on a specific channel and bandwidth. If the channel were to automatically hop around as is the default option, the captured data would be inconsistent as the router hopped around channels. Of the many suitable consumer routers available, the TP-Link AC1200 was chosen.

The requirements for a data capture device are relatively stringent. Nexmon, the firmware patch used to extract CSI, only supports a few models of Broadcom/Cypress Wi-Fi chips [3]. The most affordable of the devices with compatible chips is the Nexus 5 with a BCM4339 chip. In order to flash the Nexmon firmware patch onto the Nexus 5, the phone had to be rooted in order to gain superuser privileges. With the Nexmon patch installed and router properly set up, the equipment are ready for data collection.

In order to maximize the amount of radio interference by test subjects, the router and phone should be placed in opposite corners of the room as seen in figures 2a and 2b. That allows the wireless signals to sufficiently interact with the environment and morph before being read by the phone. Once the shower/faucets are turned on, or whatever experiment is being run, the Nexmon utility filter can be given a payload to only collect CSI from a specific channel and MAC address. The tcpdump command is used to collect the data packets as they are received by the phone, 10 packets every second.



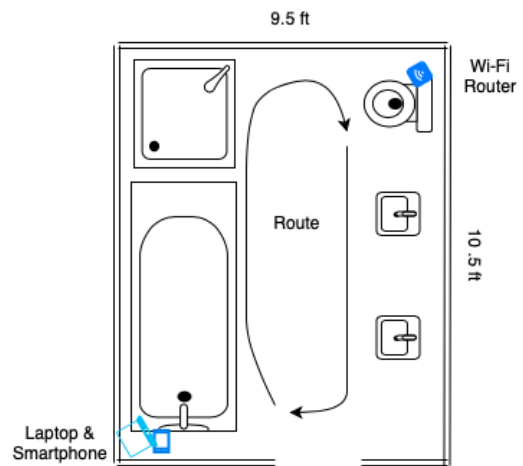
(a) Data capture device w/ laptop to control device



(b) Wi-Fi router in opposite corner of bathroom

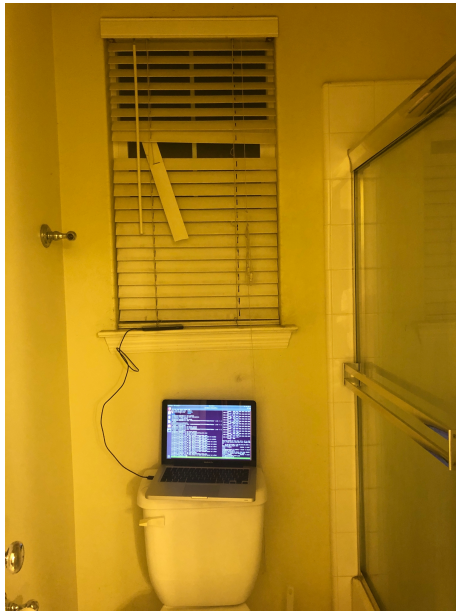


(c) Standing water in bowl and sink basins

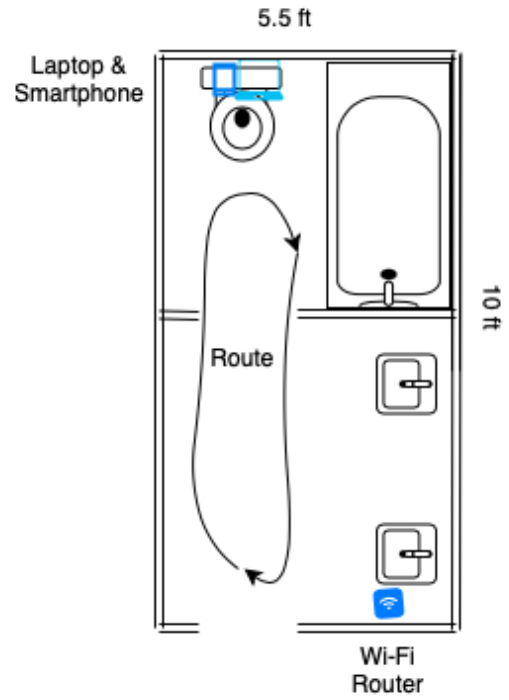


(d) Room Layout

Fig. 2: Bathroom used for data collection



(a) Data capture device w/ laptop to control device



(b) Room Layout



(c) Standing water in bowl and sink basins



(b) Wi-Fi router on opposite side of bathroom

Figure 3: Second bathroom used for additional data collection

B. Neural Network

The neural network is built using Keras, a high-level neural networks API, which abstracts away many of the minute details of neural networks and machine learning. The use of Keras allows for quick and easy experimentation and tuning of one's model. The final result is a convolutional neural network consisting of multiple convolutional layers, pooling layers, dropout layers, a bidirectional long short term memory hidden layer, and a dense layer.

1) *Pooling Layer: Pool size of 2*

Pooling layers reduce the spatial dimensions of the model; this improves performance in two main areas. Having less spatial information to train on means the computational runtime is decreased, allowing for quicker feedback and time for tuning of the model. Less spatial information also means fewer parameters are used, which decreases the chance of overfitting the training data.

2) *Dropout Layer: Dropout rate of 0.5*

Dropout layers randomly select units/neurons in the network to be “ignored” or “dropped out” with a given probability. Without dropout layers, neurons tend to develop co-dependency amongst each other, resulting in overfitting [4]. By having a sufficiently high dropout rate, the performance of individual neurons and the network as a whole is improved.

3) *Convolutional Layer: Tanh activation function*

The convolutional layers are the main building blocks of convolutional neural networks. They apply filters on the given images and create feature maps of textures and patterns. The model uses the maps and an activation function to activate neurons when specific features are detected in the input.

4) *Bidirectional LSTM Layer*

The LSTM layer allows for the model to have a long and short term memory of past patterns. This increases the chance of the neural network recognizing a previously seen pattern and making a correct classification based on the pattern.

5) *Dense Layer: Softmax activation*

The fully connected dense layer applies a linear operation, connecting inputs to outputs. This is the final layer that produces predictions.

V. Results

The final model was trained on around 42 minutes of accumulated data (6 experiments, 2 bathrooms, 5 minutes each experiment). The remaining ~18 minutes of data was used for validation. As seen in figure 3, the result was a 79.4% validation accuracy. When examining the confusion matrix in figure 4, it appears the two categories the model had trouble classifying were the running shower and running faucets. One guess for these mislabelings is that the combination of the two faucets’ flows (~1.25 gallons/minute) is roughly equal to the flow of the showerhead (~2.1 gallons/minute). Regardless, the high accuracy of the model is noteworthy. The model could be able to perform so well with relatively small data due to the static nature of the experiments. Compared to a more dynamic experiment, such as humans walking in a room, the neural network likely does not need as much data. Classifying a person would need larger CSI snippets to analyze their walking gaits, whereas running water is much simpler.

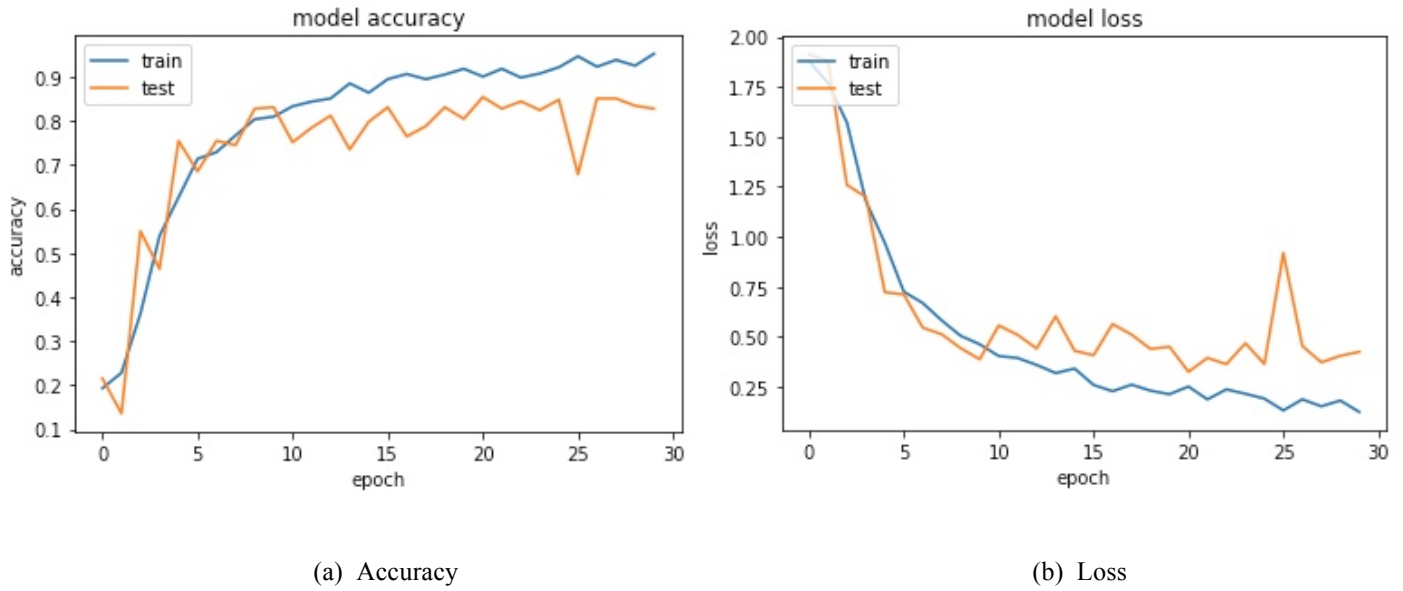


Fig. 3: Neural Network Results by Epoch

Predicted \ Actual	Empty	Faucets	Pacing	Shower	Shower & Faucets	Standing Water
Empty	41	0	5	1	0	1
Faucets	0	42	3	12	0	0
Pacing	0	0	42	0	4	2
Shower	0	14	1	49	0	0
Shower & Faucets	0	0	7	0	60	1
Standing Water	1	0	0	0	0	46

Fig. 4: Confusion Matrix of Predictions

VI. CONCLUSION

The neural network was definitely able to recognize patterns characteristic of water in the bathrooms. The results of this study show that there is great potential in the technology, as the running water in the room is a very minor change in the environment. Knowing what is specifically required to increase the accuracy is impossible, but that is the nature of black box neural networks. Yet, to see such a high accuracy with such little training data and such small test snippets is extremely promising.

VII. FUTURE WORK

When it comes to machine learning and neural networks, more data is always better. Additional experiments, such as more levels of running water, would be insightful in evaluating the technology. Collecting more data on the same experiments would be just as important, as an increased bank of training data would allow the neural network to “learn” more accurately. The original person identification study use about 40 minutes of data per experiment. If each of the experiments in this study were to be run for 40 minutes, around 736 gallons of water would be used (based on average water flow of 2.1 gallons/min for shower and 1.25 gallons/min for each sink). In drought-stricken California, I could not in good conscience find myself to leave water running for 40 minutes at a time, especially when considering that some experiments had to be run multiple times due to errors in data collection. In the right situation, where water is of abundance, collecting more data would be welcomed.

Finally, designing a model to be trained on CSV data instead of png images would circumvent a step in the training and prediction of the model, seeing as the CSV and png files essentially contain the same information.

REFERENCES

- [1] B.R. Bean and R. Abbott. (1957). Oxygen and water vapor absorption of radio waves in the atmosphere. *Geofiscia pura e applicata*.
- [2] R. Hulbert, L. Bowen, A. Strand, and A. Apitz. (2019). *Radio Tomography and Machine Learning for Person Identification*. California Polytechnic State University, San Luis Obispo
- [3] M. Schulz, D. Wegemer, and M. Hollick. (2017). Nexmon: The c-based firmware patching framework, [Online]. Available: <https://nexmon.org>.
- [4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 15. 1929-1958