Using Bluetooth RSSI Values to Classify Distance with Deep Neural Networks for COVID-19 Contact Tracing

Edward Jung   
edwardplusplus@gmail.com

*Abstract*— The Radio Signal Strength Indication (RSSI) obtained by Bluetooth can be used to estimate the proximity and duration of an individual’s exposure to patients diagnosed with COVID-19. However, due to the nature of the radiofrequency signal, fluctuations in the RSSI make it difficult to associate a measured RSSI with an exact distance. Therefore, I aim to measure the distance between two devices by mapping RSSI to distance using deep neural networks. I also display that there is a threshold value which can minimize the number of false positives.

Keywords: Contact Tracing, Bluetooth Devices, Machine Learning, COVID-19, Mobile App, Database, Privacy, Deep Neural Network, Google Tensorflow, Python

# Introduction

## Project Description

My project addresses a crucial task in contact tracing: predicting if two devices are closer than 6 feet apart. In order to correctly alert people of possible transmissions, an algorithm must be able to process the given RSSI values and determine if the two devices were close enough. If an inaccurate or faulty algorithm is implemented, numerous people could be infected or inconvenienced.

In this project, repeated measurements of RSSI between two devices at varying distances were collected for varying scenarios. After the data was collected, a conclusion regarding the possibility of an effective machine learning model to estimate distances was drawn. The procedures of the experiment were as follows:

1. Data was collected from two different locations: a) two Pis in the same room and b) two Pis in different rooms. When the two Pis were in the same room, data was collected with distances 3 feet, 6 feet, and 9 feet. When the two Pis were in different rooms, data was collected with distances 3 feet, and 9 feet.
2. Data was bootstrapped by factor of 100.
3. The initial Keras sequential model using 1 hidden dense layer with layer size 64 and adam optimizer was built.
4. Hyperparameter tuning was run on the dense layer size, the amount of dense layers, and different optimizers to find the best performing model.
5. Optimal model training was run with 128 hidden layer size, 1 Dense layer, and adam optimizer to 250 epochs.
6. Early stopping was implemented on the optimal model for the highest accuracy.
7. The RSSI threshold value was estimated from the 6 feet apart data and tested on the Experiment I dataset.

## Background Information

Contact Tracing:

Contact tracing is used to slow the spread of infectious diseases. [1] In general, contact tracing involves identifying index patients who have the disease and people who they came in contact with (contacts). [2] This includes asking people with positive diagnoses undergo isolation and their contacts to quarantine temporarily.

While contact tracing is strongly requested to prevent the spread of infectious diseases, traditional contact tracing can pose some challenges in accuracy, efficiency, and privacy. For example, manual contact tracing is subject to a person's ability to recall everyone contacted with over a certain period. Also, people tend to not reveal their private information including location data such as where they were or who they met. These difficulties request an automated, privacy-preserving contact tracing system. Today, most cell phones have Bluetooth modules integrated which can advertise their presence by the anonymous signal. Using these devices as automated tracing devices can solve most of the issues.

False Positives / False Negatives:

Distinguishing false positives and false negatives is very important. False positives are of concern since they mean that people may be led to unnecessarily quarantine. False negatives are of concern because they mean that infected people may unawarely spread the infection.

Bluetooth Advertisement:

[3] Bluetooth is a wireless technology standard used for exchanging data between fixed and mobile devices over short distances using short-wavelength UHF radio waves in the industrial, scientific and medical radio bands, from 2.402 GHz to 2.480 GHz, and building personal area networks (PANs). Advertisements using Bluetooth are recommended for contact tracing because these advertisement “chirps” can be privacy preserving and anonymous.

Machine learning:

Machine Learning is the study of computer algorithms that improve automatically through experience. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. After the model is trained, it can be evaluated using separate “test” and “validation” datasets.

Assumptions:

I assumed that people were in an indoor space with consistent advertising and scanning devices. I did not address the effects of weather conditions, clothing, or movement of the Pis. I also did not account for any interference due to other signals such as Wi-Fi routers or other Bluetooth devices.

# Hypothesis/Hypotheses

1. Using RSSI data values, it is possible to develop an effective deep neural network to classify distances between two raspberry Pis with above 90% accuracy.
2. There is an effective RSSI value threshold which enables the raspberry pi to conclude if the person is either too far away or separated by a wall or barrier with 95% accuracy.

Explanations:

As effectively recognizing possible transmissions is vital to contact tracing, a neural network which distinguishes safe distances from unsafe ones is relevant to PACT. A neural network can help identify possible transmissions with high speed, low cost, and across millions of devices simultaneously. These identification results can be used to alert people who might have had close contact with a person who has tested positive. The model requires the most investigation as there are a multitude of parameters that can be fine-tuned.

Finding an effective RSSI value threshold which automatically disregards a certain range of data as too far away or behind an obstacle could be valuable in the long term. The threshold value enables the model to process less data which allows for higher efficiency with larger datasets. Data collection requires the most investigation as the distance(s) at which the data is collected must be determined.

# Experiments and Data Collections

TABLE I.        Experiment Overview

|  |  |  |  |
| --- | --- | --- | --- |
| **Exp. #** | **Hypothesis** | **Reason** | **Repetition** |
| **# 1**  **RSSI Data Collection Experiment** | Using RSSI data values, it is possible to develop an effective deep neural network to classify distances between two Raspberry Pis with above 90% accuracy. | The RSSI data is needed to train the neural network and increase the accuracy of classification. | 4 |
| **# 2 Disregarding RSSI Threshold Experiment** | There is an effective RSSI value threshold which enables the raspberry pi to conclude if the person is either too far away or separated by a wall or barrier with 95% accuracy. | The RSSI values at 6 feet must be collected to determine the threshold that disregards data most accurately. | 1 |

## Plan and Execution

For the RSSI Data Collection Experiment, I collected data in four different scenarios. In total, I collected 2,000 data points with 500 RSSI values for each scenario. Throughout this experiment, the Raspberry Pis were stationary on the floor and experienced as little movement as possible between each scenario. The advertising and scanning Pis were also kept constant throughout the experiment.

1. Data collection with Pis 3 feet apart in an indoor open space.
2. Data collection with Pis 9 feet apart in an indoor open space.
3. Data collection with Pis 3 feet apart with an interior wall in between.
4. Data collection with Pis 9 feet apart with an interior wall in between.

TABLE II.        Experiment I Information

|  |  |  |
| --- | --- | --- |
| **Scenario #** | **Conditions** | **Target**  **Classification** |
| **#1**  **open**  **3 feet** | - 3 feet apart  - Indoor open space  - 500 data points | 1 |
| **#2**  **open**  **9 feet** | - 9 feet apart  - Indoor open space  - 500 data points | 0 |
| **#3**  **wall**  **3 feet** | - 3 feet apart  - Interior wall in between  - 500 data points | 0 |
| **#4**  **wall**  **9 feet** | - 9 feet apart  - Interior wall in between  - 500 data points | 0 |

Fig. 2.   The one target classification means that the Pis were close enough for a possible transmission while the zero target classification means that the Pis were too far away.

Bootstrapping also gave me an effective method to increase the amount of data points in a short period of time. I bootstrapped each scenario data by a factor of 100 which gave a total of 200,000 data points in the end.

For the Disregarding RSSI Threshold Experiment, I collected data in a single scenario for 650 data points. The raspberry Pis were placed 6 feet apart on the ground in an open indoor space. Throughout the experiment, the Pis were kept stationary on the ground with minimal surrounding movement. The advertising and scanning Pis used in the previous experiment were kept constant.

## Data Relevance

Collecting RSSI values from a distance of less than 6 feet and greater than 6 feet allows for the creation of a classification neural network. I also considered the effect of an interior wall on the RSSI values and added wall data to the 0th class (Table II). Using this categorized data, the model would be able to differentiate between RSSI values that are too low or too high. If the model is able to achieve at least 80% accuracy, I would be able to accept my first hypothesis. By collecting RSSI values with the Raspberry Pis 6 feet apart, I would be able to estimate a disregarding threshold value which could be used alongside my model. If the threshold value is able to disregard low RSSI values with an accuracy of at least 95%, I would be able to accept my second hypothesis.

## Examples

Experiment I:

A screen shot of a monitor

Description automatically generated

**Caption: Pandas Experiment I DataFrame for four scenarios RSSI data**

A screen shot of a monitor

Description automatically generated

**Caption: Pandas Experiment I Dataframe for four scenarios for bootstrapped RSSI data**

TABLE III.        Distribution Graphs

|  |  |  |
| --- | --- | --- |
| **Dist.** | **Original Data** | **Bootstrapped Data** |
| Open  3 Feet | A picture containing clock  Description automatically generated | A close up of text on a white background  Description automatically generated |
| Open  9 Feet | A picture containing clock  Description automatically generated | A picture containing clock  Description automatically generated |
| Wall  3 Feet | A screenshot of a cell phone  Description automatically generated | A picture containing looking, sitting  Description automatically generated |
| Wall  9 Feet | A screenshot of a cell phone  Description automatically generated | A picture containing clock  Description automatically generated |

Experiment II:

A screenshot of a computer

Description automatically generated

**Caption: Experiment II Pandas Dataframe All Data**

A close up of text on a white background

Description automatically generated

**Caption: Experiment II RSSI distribution**

# Analysis and Algorithms

## Description

Using the tensorflow feature classification documentation, I developed a feed forward sequential deep neural network to classify if two Raspberry Pis were less than 6 feet or greater than 6 feet apart. The model was created using keras, tensorflow, pandas, python, and Google Colaboratory as the online IDE.  The goal for this algorithm was to surpass 90% accuracy on the test dataset.

Initial Model Structure:

A screenshot of a cell phone

Description automatically generated

After training the initial model for 40 epochs, I achieved 74% accuracy. I then ran hyperparameter testing to fine tune the model’s parameters and scout for better performing model structures. The parameters that I tested were the amount of dense layers, the layer size, and the optimizer. In an attempt to further increase the accuracy, I trained the optimal model for 250 epochs and implemented early stopping.

With the 650 data points collected in Experiment II, I calculated the disregarding threshold by taking the minimum RSSI value. The methodology is that in order to have an RSSI value less than the lowest six feet value, the distance between the Pis must be farther away. This method of analysis also effectively covers the wall data as those RSSI values are dramatically lower than the Experiment II data.

## Results and Examples

TABLE IV.        Evaluation Accuracy

|  |  |  |
| --- | --- | --- |
|  | **RSSI Deep Neural Network** | **Disregarding Threshold** |
| **Final Accuracy** | 86.28% | 100% |

Result Description for Hypothesis 1:

After running hyperparameter testing, I concluded that the optimal model used one dense layer, 128 layer size, 0.25 dropout, and the adam optimizer. The final test accuracy of our model was 86.28%.

Result Description for Hypothesis 2:

The minimum RSSI value of the Experiment II data was -67. Testing the estimated threshold value on the entire Experiment I dataset led to 52,909 data points considered too low. As all 52,909 data points were in the 0th class, the threshold achieved 100% accuracy in disregarding insignificant data.

**Hypothesis I Results:**

Hyperparameter Testing Tables:

Optimizer: adam

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 Dense  Layer | 2 Dense Layers | 3 Dense Layers |
| Hidden  layer size 64 | A close up of text on a white background  Description automatically generatedA close up of a map  Description automatically generatedAccuracy: 0.8328 | A close up of a map  Description automatically generatedAccuracy: 0.8606A close up of a piece of paper  Description automatically generated | A close up of text on a white background  Description automatically generatedAccuracy: 0.8596A close up of text on a white background  Description automatically generated |
| Hidden layer size 128 | Accuracy: 0.8A close up of text on a white background  Description automatically generated624A close up of text on a white background  Description automatically generated | Accuracy: 0.8A close up of text on a white background  Description automatically generated583A close up of text on a white background  Description automatically generated | Accuracy: 0.83A close up of text on a white background  Description automatically generated09A close up of text on a white background  Description automatically generated |
| Hidden layer size 256 | A close up of text on a white background  Description automatically generatedAccuracy: 0.8574A close up of text on a white background  Description automatically generated | A close up of text on a white background  Description automatically generatedA close up of text on a white background  Description automatically generated Accuracy: 0.8608 | A close up of text on a white background  Description automatically generatedA close up of text on a white background  Description automatically generated Accuracy: 0.8597 |

Optimizer: rmsprop

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 Dense Layer | 2 Dense Layers | 3 Dense Layers |
| Hidden layer 64 | Accuracy: 0.83A close up of text on a white background  Description automatically generated56A close up of text on a white background  Description automatically generated | Accuracy: 0.83A close up of text on a white background  Description automatically generated19A close up of text on a white background  Description automatically generated | Accuracy: 0.8A close up of text on a white background  Description automatically generated350A close up of text on a white background  Description automatically generated |
| Hidden layer 128 | A close up of text on a white background  Description automatically generated Accuracy: 0.8342A close up of text on a white background  Description automatically generated | Accuracy: 0.A screenshot of a cell phone  Description automatically generated7592A close up of text on a white background  Description automatically generated | Accuracy: 0.7942A close up of a piece of paper  Description automatically generated |
| Hidden layer 256 | A close up of text on a white background  Description automatically generatedAccuracy: 0.8346A close up of text on a white background  Description automatically generated | Accuracy: 0.A close up of text on a white background  Description automatically generated7937A close up of text on a white background  Description automatically generated | Accuracy: 0.8A close up of a map  Description automatically generated248A close up of text on a white background  Description automatically generated |

Optimal Model Training (250 Epochs, Adam Optimizer, 128 Hidden Layer Size, 1 Dense):

|  |
| --- |
|  |



**Caption: Highest accuracy after early stopping implementation**

**Hypothesis II Results:**

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Caption: All the RSSI values that were below the threshold (True) were in the 0th class. This meant that the threshold achieved 100% accuracy in disregarding RSSI values that were too low.**

# Conclusions

## Hypothesis Evaluation

According to the Evaluation Accuracy Table (#4), the final accuracy of the model was 86.28%. I found the first hypothesis to be indeterminate as I achieved an accuracy of lower than 90% but perhaps another researcher might have achieved a higher accuracy.

According to the Evaluation Accuracy Table (#4), the final accuracy of the disregarding threshold on the RSSI data was 100%. Therefore, I found the second hypothesis to be true as the accuracy was above 95%.

## Noteworthy Conclusions

1. A deep neural network is able to classify distance via Bluetooth RSSI signals with at least 86.28% accuracy.
2. If two people(devices) are in different rooms and sharing a wall, a neural network is able to exclude this case from actual contacts even though they are within 6 feet away.
3. RSSI value thresholds and neural networks can effectively reduce the number of false-positive cases.

## General Lessons Learned

I learned that it is difficult to increase the accuracy of models with only Bluetooth RSSI values. Since RSSI values fluctuate even when the advertising and scanning devices are stationary, it is difficult to make reliable conclusions on the distance between them.

# Next Steps

In terms of data, I can incorporate external factors such as temperature, humidity, and the presence of other radiofrequency signals into my model. I can also test the effects of exterior walls on the RSSI values. If the exterior walls have higher attenuation than interior walls, I can assume that the model and threshold value can accurately classify the RSSI values as insignificant.

In terms of modeling methods, distance prediction formulas using RSSI value can be applied to further increase the accuracy of the classification model. I can also set up the Google Cloud GPU and run additional hyperparameter testing on the optimal model.

For the practical aspect in real world contact tracing, I can consider the differences in RSSI measurements during various conditions. For example, I can take into consideration the effects of device orientation, device location on the human body, model of the Bluetooth module, and surrounding articles of clothing.

Using the model, I can develop a mobile application that uses alert systems to notify the user of a potential disease transmission. I can create sql databases to hold Bluetooth RSSI information and user IDs which would make storing and processing data more organized and efficient. Lastly, I can implement privacy algorithms to preserve anonymity and ensure that only necessary information is given.

##### References

1. "Digital Contact Tracing for Pandemic Response." 2020. doi:10.1353/book.75831.
2. "COVID-19 Contact Tracing." Centers for Disease Control and Prevention. https://www.cdc.gov/coronavirus/2019-ncov/daily-life-coping/contact-tracing.html.
3. "Bluetooth." Wikipedia. September 08, 2020. https://en.wikipedia.org/wiki/Bluetooth.
4. "Classify Structured Data with Feature Columns  :   TensorFlow Core." TensorFlow. https://www.tensorflow.org/tutorials/structured\_data/feature\_columns.
5. Purdue Writing Lab. "Writing Report Abstracts // Purdue Writing Lab." Purdue Writing Lab. https://owl.purdue.edu/owl/subject\_specific\_writing/professional\_technical\_writing/technical\_reports\_and\_report\_abstracts/index.html.