The International Young Researchers' Conference
DOI: doi.org/10.34614/iyrc0044

Proximity Detection in Contact Tracing With Bluetooth Technology During Covid19 Pandemic

Alice Feng
Harker School, United States
alicezfeng@gmail.com

Abstract

The global pandemic COVID19 is spreading through close contact between an infected person and another person. Contact tracing is a useful tool in preventing the spread of the pandemic. Contact tracing identifies infected patients and their contacts who may have been exposed and asks them to quarantine. To automate contact tracing, Bluetooth Low Energy (BLE) technology is one option to identify the distance between two objects by correlating the attenuation of BLE signal strength with distance. Hence it can be used to automatically record how close two persons are and for how long. In this project, we studied the feasibility of correctly identifying the distance between two indoor objects using Raspberry Pi's and BLE technology. We conducted extensive experiments, correlated BLE signal strength attenuation with distance, and designed a simple lowpass filtering algorithm which optimized regression fitting and proximity decision, achieving accuracy over 90%. We also studied the impact of human obstruction, multipath, humidity, temperature, and down-sampling on proximity detection.

Keywords: COVID19, Contact tracing, Proximity Detection, Bluetooth

Introduction

Contact tracing (Centers for Disease Control and Prevention, 2020a) is the process of identifying people who have come into contact with an infected person and mapping out who they came into contact with. People who are in close contact with an infected person are at a higher risk of becoming infected themselves. In the current COVID19 pandemic, due to the large number of infected patients, an automated contact tracing approach is necessary. Several companies are developing exposure detection functions in smart phones using Bluetooth Low Energy (BLE) technology. BLE technology allows significantly lower power consumption and cost, and has been used in healthcare, fitness, and security. Apple and Google are working together on an Exposure Notification system (Apple & Google, 2020) which utilizes BLE technology on mobile devices to achieve automated contact tracing. The Private Automated Contact Tracing (PACT) (MIT & Massachusetts General Hospital, 2020; Rivest et al., 2020) is another automated contact tracing project lead by MIT and Massachusetts General Hospital Center for Global Health to develop exposure detection functions in personal digital communication devices using BLE. Studying the correlation of

BLE signal attenuation between two objects with their distance, or proximity detection, is of great importance and interest.

In this project, we studied the feasibility of correctly identifying the distance between two indoor objects using Raspberry Pi's and BLE technology. Raspberry Pi is used in this study as it has a standardized BLE protocol with identical antennas, while different cellphone brands and models use different BLE protocols and have different antenna designs, making it complicated to study the correlation of cellphone BLE signal strength attenuation with distance on a large scale. In addition, BLE signal attenuation can be affected by variants such as human shadowing, antenna patterns including angle and polarization, and deep multipath (Hatke et al., 2020). Through Raspberry Pi, we can more easily control the variants described above, and plot a clearer picture between BLE signal attenuation and distance. Then we study the impact of each variant one at a time.

In our experiments, one Raspberry Pi continuously emitted BLE beacons or "chirps," and another Pi periodically scanned the chirps and recorded the transmitting power that was sent along with the chirp and the received power, and then calculated the BLE signal attenuation in dBm (decibel-milliwatts), referred to as RSSI (Received Signal Strength Indication). We correlated RSSI with the distance between two Pi's, designed and compared data processing algorithms to improve the Receiver Operating Characteristics (ROC) curve and Area Under ROC (AUC), studied the impact of multipath, human obstruction, humidity, and temperature on RSSI, and experimented with down-sampling for power and memory saving benefits.

Methodology

Raw Data Collection

In this project, two Raspberry Pi's (Model 4-B) were placed at the same height, with one Raspberry Pi's antenna pointing towards the other Raspberry Pi's antenna. One Raspberry Pi advertised the BLE beacons every 250ms; the other Raspberry Pi scanned the chirps and recorded its RSSI at 1 second intervals for 10 minutes at each distance level increasing from 10cm to 400cm at 10cm increments. The ceilings of the rooms are 2.6m (8ft) tall. The floors are laminated floors. We designed a series of experiments to collect the raw data.

a) In experiment A, the Raspberry Pi's were placed in a 3 meter by 5 meter empty room without obstructions in between them. The empty room was used to reduce the impact of multipath. The temperature was 22^{o}C (72F), and humidity was 50%.
b) In experiment B, the Raspberry Pi's were placed in a 1m by 5m corridor with closed doors and walls on both sides to simulate a multipath environment. The temperature was 22^{o}C (72F), and humidity was 50%.
c) In experiment C, the Raspberry Pi's were placed in the 3m by 5m room with a human standing in between the two Pi's. The human is 163cm tall and weighs 120lb. The temperature was 22^{o}C (72F), and humidity was 50%.
d) In experiment D, the Raspberry Pi's were placed in the 3m by 5m room without obstructions. The temperature of the room was lowered to 17^{o}C (63F) through air-conditioning, or raised to 27^{o}C (81F) through heating. Humidity was 50%.
e) In experiment E, the Raspberry Pi's were placed in the 3m by 5m room without obstructions. The humidity of the room was lowered to 30% through a dehumidifier, or raised to 70% through a humidifier. The temperature was 22^{o}C (72F).

Data Processing Algorithms

In this project, the goal was to correctly determine "if the two Pi's are closer than 2m" (Centers for Disease Control and Prevention, 2020b) given an RSSI value. The raw RSSI data had lots of random noise (Figure 1) making it more difficult to distinguish between distances. It is usually desirable to preprocess raw data with a low-pass filter to suppress noise and improve signal to noise ratio. Hence, we designed a simple low-pass filter algorithm "Averaging Over Non-overlapping Intervals (AOI)" to optimize regression fitting and improve proximity detection accuracy. The algorithm was compared with four other standard low-pass filter algorithms.

a) Averaging Over Non-overlapping Intervals (AOI) keeps only the average for every interval size *K* of consecutive raw data. The $m[i] = \frac{a[iW] + a[iW+1] + \ldots + a[iW+W-1]}{W}$ computational complexity of AOI is *O(n)*, the lowest among all algorithms tested in this project. Let raw data sequence be $a[0], a[1], a[2], \ldots$, and let the data sequence after AOI of interval size *K* be $m[0], m[1], m[2], \ldots$. then,

$$m[i] = \frac{a[iK] + a[iK+1] + \ldots + a[iK+K-1]}{K} \quad (1)$$

b) Moving Median Filtering (MF) (MathWorks, n.d.a) with a sliding window of size *K*. The processed data is the median of the sliding window of consecutive *K* raw data. The computational complexity of MF is *O(nK*Log(K))*. Let the raw data sequence be $a[0], a[1], a[2], \ldots$, let the data sequence after MF with sliding window size *K* be $m[0], m[1], m[2], \ldots$, then

$$m[i] = median\{a[i], a[i+1], \ldots a[i+K-1]\} \quad (2)$$

c) Moving Average Filtering (AF) (MathWorks, n.d.b) with a sliding window of size *K*. The processed data is the average of the sliding window of consecutive *K* raw data. The computational complexity of AF is *O(nK).* Let the raw data sequence be $a[0], a[1], a[2], \ldots$, let the data sequence after AF with sliding window size W be $m[0], m[1], m[2], \ldots$, then

$$m[i] = \frac{a[i] + a[i+1] + \ldots + a[i+K-1]}{K} \quad (3)$$

d) Wiener Filtering (WF) (Mathworks, n.d.c) minimizes the mean squared error between the original data and the filtered data. The computational complexity of WF is *O(nK).* Let u_x and v_x be the mean and variance of the original data *x* with window size *K*, let v_n be the estimated variance of the noise. Then the filtered data y is

$$y = u_x + (x - u_x)\frac{v_x}{v_x + v_n} \quad (4)$$

e) Savitzky-Golay (SG) filter (Mathworks, n.d.d) is a digital filter to smooth the data by applying convolution to the data using specially designed low-degree polynomial convolution kernels. It uses local polynomial of degree *D* to fit a subset of the data of size *K* by means of linear least squares. The computational complexity of SG filter is $O(n^D)$.

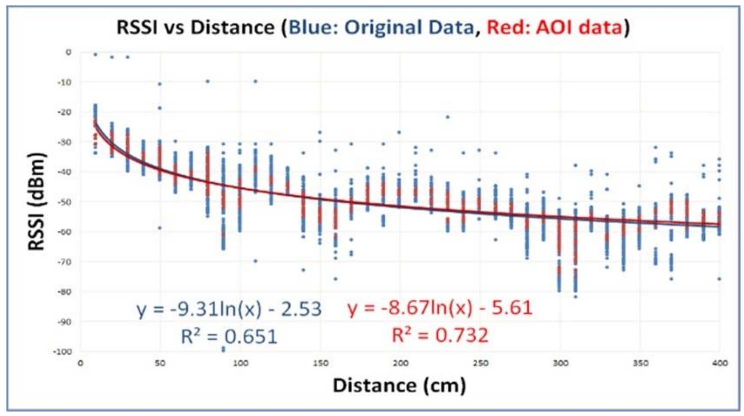

FIGURE 1: RSSI vs Distance, using raw data vs AOI data (interval=60).

Performance Metrics

For each RSSI vs distance correlation, we evaluated the following performance metrics.

a) Linear regression model coefficient of determination R^2 (also called *R*-square). We used linear regression model in log scale to fit the RSSI vs distance correlation. R^2 indicates how well the model fits the data.

b) The accuracy of a classifier at each threshold is defined below. The metrics of True Positive and True Negative in this project are summarized in Table 1.

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Samples} \quad (5)$$

c) Receiver Operating Curves (ROC). The ROC curve plots the true positive rate vs the false positive rate at various threshold settings. The area under ROC (AUC) reflects the accuracy of the classifier.

d) Computational complexity. It estimates the time required to compute the filtering algorithm in terms of *n*, the size of the input, and it is also an indication of battery consumption.

TABLE 1: Proximity decision measures

	True Positive	False Positive	True Negative	False Negative
Decision	< 2m	< 2m	> 2m	> 2m
Truth	< 2m	> 2m	> 2m	< 2m

Results and Discussion

RSSI vs Distance using Raw Data and Processed Data

Figure 1 compares the RSSI of raw data and AOI filtered data. The raw data contains significant random noise, and at each distance level the RSSI values can vary by as much as 40 dBm, making it more difficult to be used as a classifier. After applying AOI, the RSSI values vary by 5-10 dBm at most distance levels. The best regression fitting curve after AOI is a logarithm function

$$y = -\ 8.666\, ln(x) - 5.5611,\ R^2 = 0.7322 \qquad (6)$$

In Figure 2, the BLE attenuation $P = 10^{RSSI/10}$ fits well as a function inversely proportional to the square of distance, which is consistent with BLE power attenuation theory that signal strength deteriorates with distance at power of 2:

$$y = \frac{0.2779}{x^{1.995}},\ \ R^2 = 0.7322 \qquad (7)$$

The nicely fitted linear regression models and the relatively high R^2 show that RSSI can be correlated with distance with sufficient accuracy.

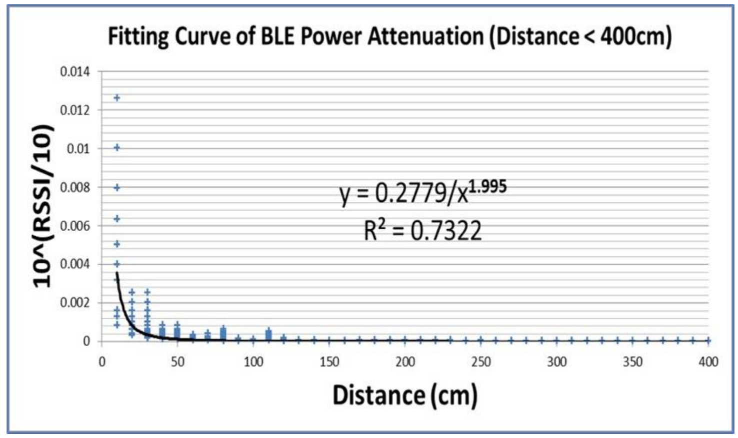

FIGURE 2: The fitting curve of BLE power attenuation after AOI.

Impact of Multipath on RSSI

Figure 3 shows that the empty room (experiment A) and corridor (experiment B) have close RSSI curves. The RSSI in corridor is stronger at lower distance but weaker at higher distance levels, but the difference is less than 2 dBm on average. Hence multipath due to the shape of a room slightly affects RSSI.

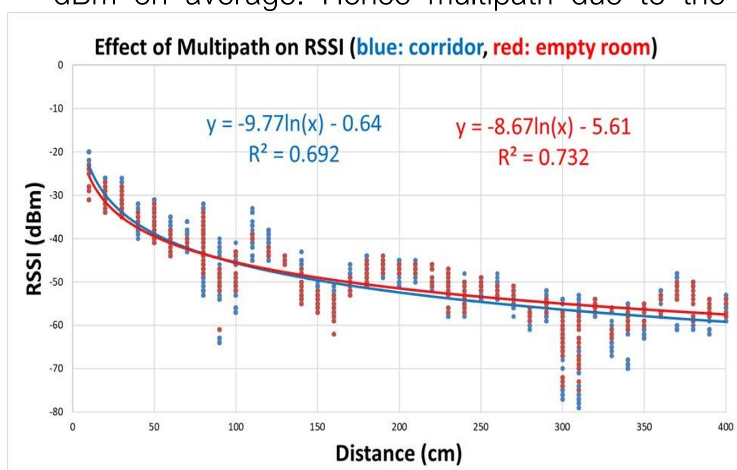

FIGURE 3: Multipath impact on RSSI raw data.

Impact of Human Obstruction (OB) on RSSI

With human obstruction, RSSI is around 5-10 dBm lower than without obstruction (Figure 4). RSSI with human obstruction also has significantly more noise and degraded regression fitting (Table 2). Applying AOI filtering greatly improved the degraded AUC, accuracy, and R^2 caused by human obstruction. However, RSSI no longer fits as a logarithm but a polynomial even after AOI filter or other low pass filters. Hence power attenuation with human obstruction no longer deteriorates with distance at power of 2. The optimal threshold derived from data with human obstruction is also 10 dBm lower than without human obstruction.

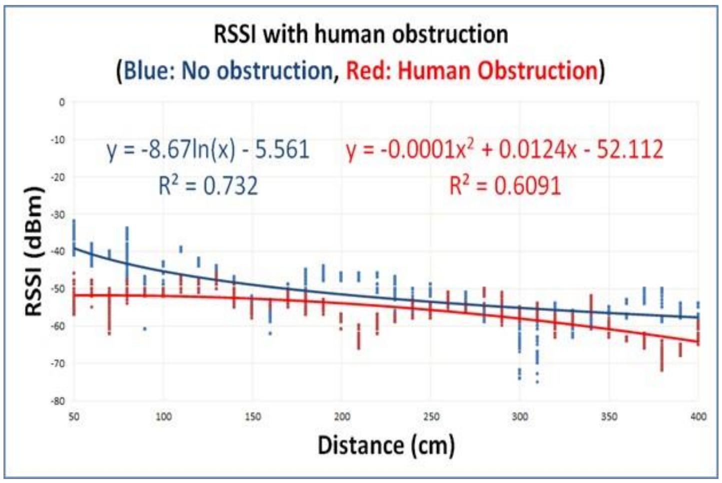

FIGURE 4: RSSI with human obstruction between the objects (after applying AOI with interval=60).

Impact of Temperature on RSSI

Figure 5 compares three temperature settings at 17°C (63F), 22°C (72F) and 27°C (81F). For every 5°C temperature increase, RSSI decreases by about 2 dBm. Similar observations have been reported in prior research using different devices and in different environment (Texas Instrument, 2008; Luomala, 2015; Guidara, 2018).

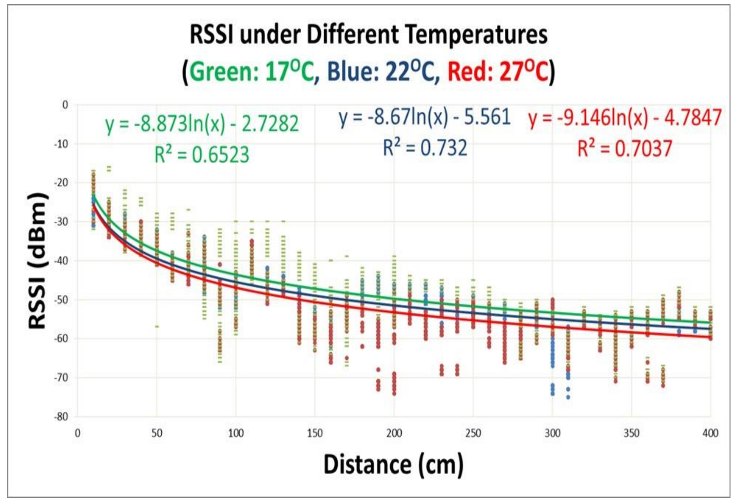

FIGURE 5: Impact of temperature on RSSI.

Impact of Humidity on RSSI

Figure 6 compares three humidity settings at 30%, 50% and 70%, but there is no noticeable impact of humidity on RSSI. This result is consistent with some prior research (Luomala, 2015), even though contradictory to some other results (Guidara, 2018). The exact device and signal used may explain the difference.

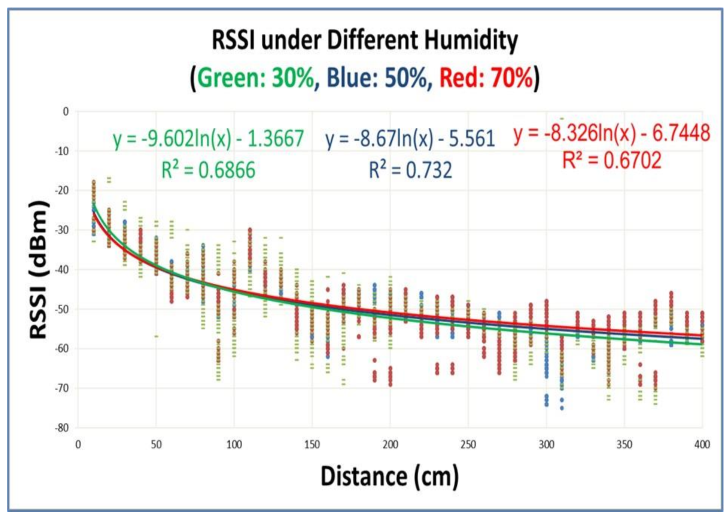

FIGURE 6: Impact of humidity on RSSI.

Proximity Decision – Closer than Two Meters? Yes or No

In COVID19 contact tracing, closer than 2m is considered 'too close'. To make a binary decision if the two Raspberry Pi's are less than 2 meters apart, we first plotted the RSSI histogram when the distance is less than 2 meters and the RSSI histogram when the distance is more than 2 meters (Figure 7). Since the two histograms overlap, for each RSSI threshold selected, the true positive, true negative, false positive, false negative values also change accordingly.

Our goal is to pick the Optimal Threshold - a threshold of RSSI value to maximize the decision Accuracy (Equation 5). Figure 8 shows the accuracy value vs the RSSI threshold. Here, RSSI threshold = -46 dBm leads to the highest accuracy 88.96% and is the Optimal Threshold in this example.

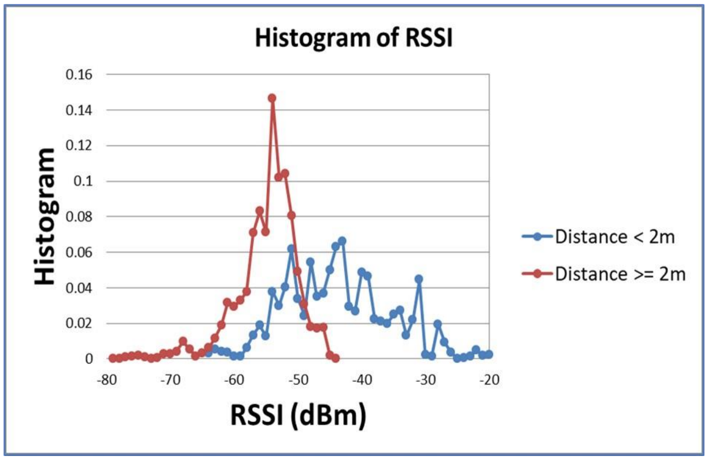

FIGURE 7: The histograms of RSSI when the distance is less than 2 meters or greater than 2 meters after AOI filtering.

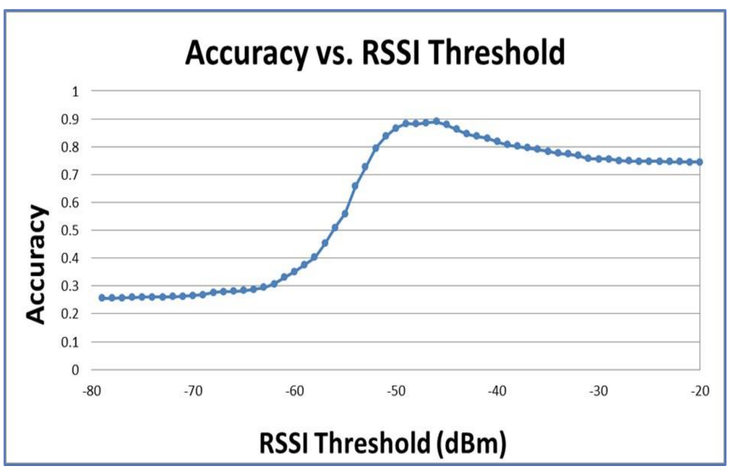

FIGURE 8: The accuracy of deciding if two Pi's are less than 2 meters apart given different RSSI thresholds after AOI filtering.

Comparing Different Data Processing Algorithms

Table 2 compares the performance of different filtering algorithms. Some discussions follow.

a) As an example, Figure 9 compares the ROC curves of AOI with varying interval sizes. The curves are way above the random decision line, i.e., the (0,0) to (1,1) diagonal line, hence they are good decision tests. The ROC curves show that even at about 60% True Positive rate, we can still eliminate nearly all False Positives. Yet to achieve 90% True Positive rate, we would have to allow a 50% False Positive rate. Overall, based on the RSSI values, we can make a rather accurate decision with about 89% accuracy to decide whether the two Raspberry Pi's are closer than 2 meters.
b) Using raw RSSI data without filtering can achieve relatively good decision performance, despite outliers and random noises.
c) To improve proximity decision performance, the RSSI raw data should be passed through a low pass filter.
d) Increasing the sliding window size or interval size of the low pass filter showed a noticeable improvement in AUC, accuracy and R^2, at the price of higher latency and CPU usage which translates to higher battery consumption.
e) AOI, MF or AF outperforms WF and SG. Table 2 shows the best performances of WF and SG.
f) AOI achieves the best AUC, accuracy, R^2 over all low pass filters we tested. It is also the easiest to implement, has the lowest computation complexity, and uses the least memory.

TABLE 2: Performance comparison

ALGORITHM OR SETTING	WINDOW SIZE OR DS RATE	OPTIMAL THRESHOLD (DB)	BEST ACCURACY	AUC	R^2
RAW	N/A	-46	0.874	0.87	0.651
OB	N/A	-55	0.717	0.75	0.325
OB+AOI	60	-56	0.787	0.87	0.609
AOI	5	-46	0.885	0.89	0.720
AOI	60	-46	0.887	0.90	0.741
AOI	300	-46	0.880	0.91	0.746
AOI	600	-51	0.915	0.91	0.890
MF	5	-46	0.890	0.89	0.689
MF	61	-46	0.891	0.90	0.732
MF	301	-47	0.884	0.91	0.730
MF	601	-47	0.874	0.91	0.718
AF	5	-46	0.885	0.89	0.721
AF	60	-46	0.886	0.90	0.739
AF	300	-46	0.878	0.91	0.739
AF	600	-51	0.884	0.92	0.735
WF	600	-46	0.887	0.89	0.738
SG	600	-50	0.889	0.91	0.745
DS	5	-46	0.884	0.87	0.651
DS	10	-46	0.884	0.87	0.652
DS	20	-46	0.884	0.87	0.652
DS+AOI	5	-47	0.887	0.88	0.68
DS+AOI	10	-49	0.890	0.89	0.732
DS+AOI	20	-46	0.892	0.92	0.791

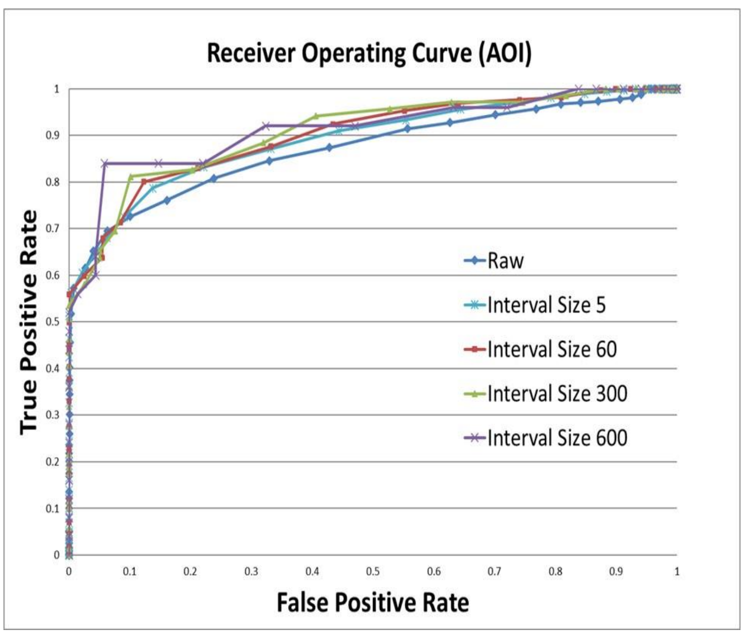

FIGURE 9: Comparing ROC of AOI with varying interval sizes.

Down-sampling

When the scanner scans every N (N>1) seconds, it saves power and memory. Down-sampling, i.e., keeping every N-th sample of the original data, is used to simulate less frequent scanning. Table 2 shows that down-sampling or down-sampling plus AOI (K=60) does not degrade performance. Hence it is reliable to scan every 5, 10 or 20 seconds.

Conclusion

In this project, extensive tests were conducted to correlate BLE RSSI with the distance between two indoor Raspberry Pi's. Our discoveries and contributes follow.

We discovered that the RSSI of two indoor Raspberry Pi has a strong correlation with the distance between the two Raspberry Pi's when they are less than 4 meters apart with all other variables being constant. Through linear regression modeling, BLE RSSI fits as a logarithm function of the distance, and achieve high R-square value.

BLE RSSI can be used as a fairly accurate indicator of whether two indoor objects are less than two meters apart. To improve ROC, the raw RSSI data should be passed through a low pass filter. The best accuracy ranges from 87.35% to 91.49% depending on the low pass filter.

The low pass filter algorithm AOI (Average Over Non-Overlapping Interval) outperforms other standard low pass filter algorithms in regression fitting, proximity detection, computational complexity, and memory requirement. When there is no obstruction between the objects, the RSSI after AOI filter fits well as a logarithm function of the distance and can indicate if two objects are closer than 2 meters with 91% accuracy.

Human obstruction introduces RSSI degradation. However, AOI can significantly improve the degradation caused by human obstruction.

The impact of environment variables should be considered when using BLE RSSI in proximity detection, since higher temperature reduces RSSI; multipath due to room shape has slight impact on RSSI as well; even though humidity does not affect BLE RSSI in our tests.

To save memory and CPU, the scanner can reliably scan at 20 seconds interval if AOI filter is used.

Through this project, we have shown that BLE technology is very promising in correctly detecting the distance between two objects. The impact of each environment variable was identified individually and can be used in constructing a holistic approach to correlate RSSI with distance. This project is a great starting point in developing automated contact tracing techniques for effective pandemic control, not only for COVID19 but also for other air-borne communicable diseases. The following topics are interesting for future work.

The data preprocessing algorithm can be made dynamic depending on the frequency of distance changes. Different antenna angles and locations such as placing the Raspberry Pi's inside a pocket should be tested to observe how they affect RSSI and distance detection. More Raspberry Pi's can be added to see how their BLE signals interfere with distance detection. All the tests should be repeated in more indoor and outdoor locations.

Besides BLE technology, ultrasound and wide band wireless technologies are being investigated for use in contact tracing. How these technologies compare to each other is also of great interest for future work.

Furthermore, some technique and ethnic issues still need more discussion. For example, the impact on RSSI when the two devices are not identical, such as an iPhone and an Android phone; protecting each user's privacy through proper protocol and encryption.

To summarize, this project successfully used BLE technology in proximity detection for COVID19 and paved the road for using BLE technology in automatic contact tracing of infectious diseases in a large scale.

Acknowledgments

I would like to thank MIT Lincoln lab for providing software that are used to advertise and scan Raspberry Pi's BLE signals and record RSSI values (MIT Lincoln Lab, 2020).

References

Apple & Google. (2020). *Privacy-Preserving Contact Tracing.* https://www.apple.com/covid19/contacttracing

Centers for Disease Control and Prevention. (2020a). *Contact Tracing: Frequently Asked Questions.* www.cdc.gov/coronavirus/2019-ncov/php/contact-tracing-faq.html

Centers for Disease Control and Prevention. (2020b). *Social Distancing*. https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/social-distancing.html

Hatke, G.F., Montanari, M., Appadwedula, M., Wentz, M., Meklenburg, J., Ivers, L., Watson, J., & Fiore, P. (2020). *Using Bluetooth Low Energy (BEL) signal strength estimation to facilitate contact tracing for COVID-19.* https://pact.mit.edu/wp-content/uploads/2020/11/TCFTL_paper_FINAL_V15_w_logos.pdf

Guidara, A., Fersi, G., Derbel, F., Jemaa, M.B. (2018), *Impacts of Temperature and Humidity variations on RSSI in indoor Wireless Sensor Networks*. Procedia Computer Science. Vol 126, pp1072-1081. Available from: https://doi.org/10.1016/j.procs.2018.08.044

Luomala, J., Hakala, I., (2015), *Effects of Temperature and Humidity on Radio Signal Strength in Outdoor Wirelss Sensor Network*. Proceedings of the Federated Conference on Computer Science and Information

System. Vol 5, pp1247-1255. Available from https://annals-csis.org/proceedings/2015/pliks/241.pdf

MathWorks. (n.d.a). *Moving Median.* https://www.mathworks.com/help/matlab/ref/movmedian.html

MathWorks. (n.d.b). *Moving Average Filter.* https://www.mathworks.com/help/econ/filtering.html

MathWorks. (n.d.c). *Wiener2 – 2D adaptive noise-removal filtering*. https://www.mathworks.com/help/images/ref/wiener2.html

MathWorks. (n.d.d). *Savitzky-Golay filter design.* https://www.mathworks.com/help/signal/ref/sgolay.html

MIT Lincoln Lab. (2020). *BWSI piPACT reference code.* https://github.com/BWSI-piPACT/reference_code

MIT & Massachusetts General Hospital. (2020). *PACT: Private Automated Contact Tracing.* https://pact.mit.edu/

Rivest, R. L., Weitzner, D. J., Ivers, L. C., Soibelman, I., & Zissman, M. A. (2020). *PACT: Private Automated Contact Tracing.* https://pact.mit.edu/wp-content/uploads/2020/05/PACT-Mission-and-Approach-2020-05-19-.pdf

Texas Instrument Inc., 2008, "CC2400 – 2.4GHz Low-Power RF Transceiver", Data Sheet.