A 2-stage Classifier for Contact Detection with BluetoothLE And INS Signals

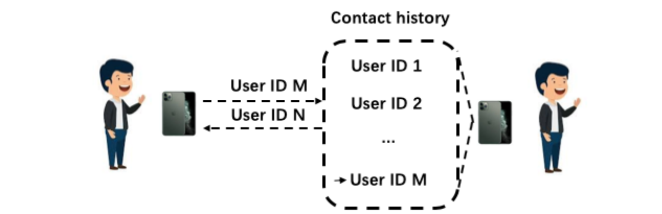
Tianlang HE
School of Engineering, Computer Science and Engineering Department
Hong Kong University of Science and Technology
Email: theaf@cse.ust.hk

Maximilian PRINTZ
School of Engineering, Computer Science and Engineering Department
Hong Kong University of Science and Technology
Email: mprintz@connect.ust.hk

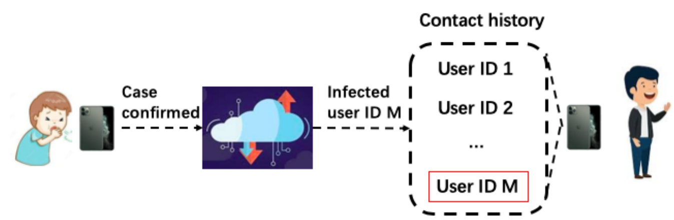
1. Introduction

Pandemic is one of the biggest threat of mankind. The outbreak of the Coronavirus disease (COVID-19) has nearly paralyzed the global economy and caused mass disruptions in people's daily lives. One of the methods to control the virus spread while optimally maintaining the normal social operation is contact tracing. Contact tracing is the process of identifying who has come into close contact with a infected case so that those suspected cases can be notified and possibly quarantined. Traditionally, manual contact tracing is conducted by interviewing infected patients about their recent tracks and possible contacts, which is labor-intensive with delayed notification. As the smartphone has become inseparable with people, automated contact tracing, which automatically collects contact event with prompt notification, was proposed.

The existing automated contact tracing technologies can be categorized as location-aware contact tracing and location-oblivious contact tracing. Location-aware contact tracing requires information that reveals users' location, such as GPS, Wi-Fi and QR code. GPS-based contact tracing determines contact event by matching the geolocations collected by smartphones. With Wi-Fi-based contact tracing, the smartphone periodically scans the surrounding Wi-Fi access points and determines contact event by matching the access point vector. QR code-based contact tracing requires the users to manually scan the QR code placed in public places and matches the QR code location to determine contact event. Although possessing several advantages over manual contact tracing, location-aware contact tracing poses severe privacy concern. The location-oblivious contact tracing directly detects the contacts between devices without knowing locations, which is privacy-preserving. The ultra-sound-based contact tracing uses the arrival time difference between Bluetooth chirps and ultra-sound waves to measure the contact distance, however, the takeover of the microphone causes inconvenience for smartphone users. As shown in figure 1, Bluetooth low energy (BLE)-based contact tracing assigns each user with a random ID and the users' smartphones exchange their IDs when encounter, contact event is determined by matching the infected person's ID with the contact history. As it is implicit, energy-saving and privacy-preserving, BLE-based contact tracing is commonly adopted.



(a) Contact Event Collection.



(b) Suspected case detection.

Figure 1. BLE-based contact tracing. (a) Each smartphone is assigned with a random ID and exchanges IDs when encounter with other smartphones. (b) The infected patient will publish his/her ID which is subsequently matched with other users' contact history to determine suspected cases.

Since the BLE signals could travel much farther than distance of close contact, BLE contact tracing needs to detect proximity based on the fact that the received signal strength (RSS or RSSI) attenuates as distance increases. However, as the BLE RSSI is very noisy, the existing BLE-based contact tracing approaches suffer the unreliable proximity detection. Generally, the main causes of the noises are complex building environment and phone carriage state. In complex building environments, the BLE signals may reflect, diffract and scatter before reaching to the receiver leading to the RSSI change, which is called multi-path effect. Also, various phone carriage states could render the signals covered by clothes or human body, causing different degrees of signal blocking. In reality, these two causes often co-exist which further complicates the scenarios. Therefore, how to accurately detect proximity using the noisy BLE

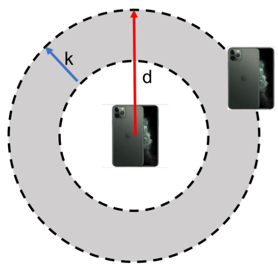


Figure 2. Ground truth description. d and $d - k$ respectively refers to the maximum and minimum distance between receiver and transmitter

RSSI poses a huge challenge for BLE contact tracing.

To overcome this issue, we propose a 2-stage deep learning-based classifier for BLE proximity detection. Our contributions are 1) we use BLE RSSI histogram representation to reduce the multi-path effect, 2) we introduce IMU data to compensate for the signal blocking from phone carriage states, 3) we leverage a deep learning-based classifier to model the impact of multi-path effect and phone carriage state on proximity detection, 4) our model is both storage efficient and computationally efficient due to the 2-stage structure, 5) we have verified our idea and got a very good result on Too Close for Too Long (TC4TL) Challenge held by The National Institute of Standards and Technology (NIST) in coordination with MIT PACT project.

The rest of the paper is organized as follows.

2. Dataset Description

We participated and verified our model in the Too Close for Too Long (TC4TL) Challenge, which is held by The National Institute of Standard and Technology (NIST) in coordination with the MIT PACT project, aiming for accurate proximity detection for BLE-based contact tracing.

As most of the data competition, a training dataset, a validation dataset and a testing dataset are given for model training, hyper-parameter tuning and model evaluation, respectively. To mimic the real world scenarios, the data are collected in diverse environment with different carriage states by different users. For example, the user may face to the transmitter or turn back to it, causing different degrees of signal blocking. As the users are free to move during the data collection, the distance may change, therefore, two parameters d and k are used to describe the ground truth. As shown in figure 2, d is the maximum distance between receiver and transmitter and $d - k$ refers to the minimum distance between receiver and transmitter, in other words, the possible location of receiver should be within the shaded region. Besides d and k, dataset also provides common smartphone embedded sensor readings collected in different time periods.

The goal of the challenge is to detect proximity on the testing data. The proximity threshold is denoted as D, if the maximum distance d is smaller than D, it is classified as a contact event, otherwise, it is classified as no contact. In the challenge, $k \in 0.9m, 2.1m$ and $D \in 1.2m, 1.8m, 3.0m$. The data are split into fine-grained dataset and coarse-grained dataset according to k. Therefore, the goal of the challenge is to determine contact event on fine-grained and coarse-grained dataset based on different proximity threshold.

3. System Description

3.1. System Overview

The system overview is shown in figure 3. The system input is the raw data from smartphone embedded sensors and output is the event type which is either contact event or no contact. Our model is consist of two stages. In stage 1, the raw data is converted to a fixed-length vector. In stage 2, a pre-trained deep learning classification model is employed to determine the event type using the fixed-length vector. In practice, smartphone keeps running stage 1 converting massive data into fixed-length vectors and stores them into the database, once the the user has the risk of exposure to virus, the stage 2 would fetch vectors from database and estimate the event type. Next, we will first introduce the data source selection, then explain how our approach reduces the multi-path effect and carriage state influence, in the end, the reason for selecting classification model is discussed.

3.2. Data Source Selection

Nowadays, smartphone has been embedded with several sensors that are available for proximity detection. In this section, we will focus on analyzing these data sources and their influences on BLE RSSI.

Overall, BLE RSSI, IMU data, magnetic field intensity and TxPower are selected as data sources for our model. BLE RSSI is the main source for proximity detection. As mentioned, higher BLE RSSI value indicates the closer distance, so RSSI values contain the distance information. IMU data is consist of acceleration and angular velocity, and acceleration can be further decomposed to linear acceleration and gravity, which are high and low frequency components of acceleration, respectively. As the gravity always points to ground, it is a good representation for phone pose. The angular velocity and linear acceleration capture the human dynamics since they are sensitive to phone movement. Therefore, IMU data represent phone carriage states. High intensity of magnetic field could rise the reflective index of air and increase the scattering probability of the BLE signals. Smartphone magnetometer can help to identify this scenario. TxPower is a value describing the transmission power of the signal transmitter, which is delivered along with the BLE handshaking signal. It serves as a very useful calibration for heterogeneity of devices.

Some other data sources are also provided by smartphone, we abandon them as they are either irrelevant to proximity detection or duplicated with the selected sources. For example, the correlation between altitude and BLE RSSI is not observed, and phone attitude is redundant since it is calculated by IMU data and magnetic field.

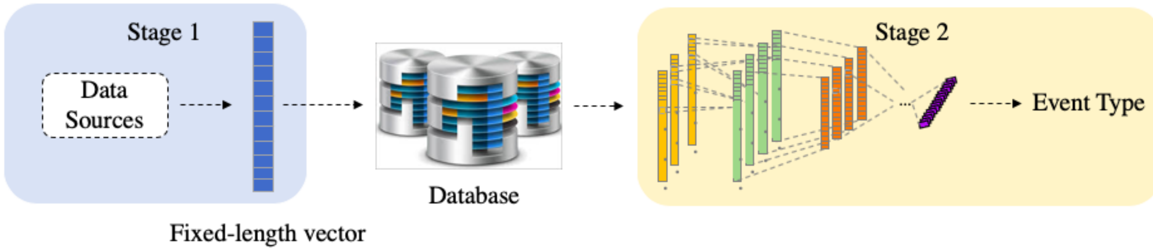
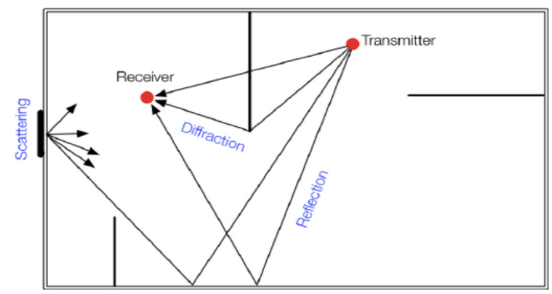


Figure 3. System overview. The input are raw data from smartphone embedded sensors. In practice, smartphone keeps running stage 1 converting massive data into fixed-length vectors and stores them into the database, once the user has the risk of exposure to virus, the stage 2 would fetch vectors from database and estimate the event type.

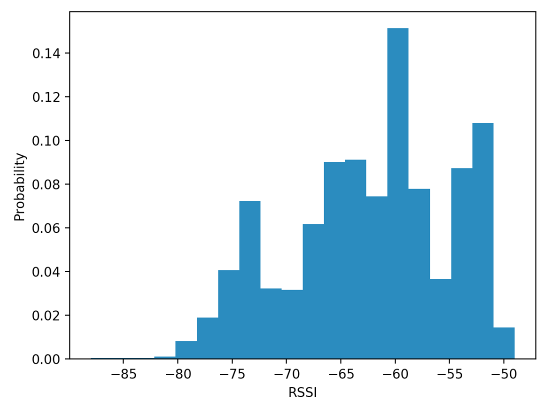
3.3. BLE Histogram Representation

As aforementioned, the higher BLE RSSI value indicates closer distance, but the multi-path effect renders the signal noises causing the unreliable proximity detection. The existing approaches try to eliminate these noises by employing filters, such as calculating mean value or Kalman filter, these approach can not adapt to diversity of environments in that the BLE RSSI variances caused by multi-path effect are not always centered or agree to Gaussian distribution.

In our model, instead of eliminating the noises, we utilize them to reduce the multi-path effect. As illustrated in figure 4, the transmitted signals go through several paths experiencing different degrees of signal attenuation before reaching to the receiver, causing the RSSI spread which can be represented by histogram. Therefore, we convert the BLE RSSI value to histogram representation to reflect the multi-path effect.



(a) Multi-path effect.



(b) BLE RSSI histogram representation.

Figure 4. BLE RSSI spread and histogram representation. (a) Multi-path renders the BLE RSSI spread. (b) RSSI spread can be represented by histogram.

3.4. Carriage State Feature Extraction

Many phone carriage states would cause the BLE signal covered by clothes or human body, such as phone carried in pocket or in bag, causing different degrees of signal blocking. Therefore, identifying these carriage states provides signal blocking information which contributes to better BLE proximity detection. As the phone carriage states are directly reflected by IMU data, we extract features from IMU data to represent the carriage states.

IMU data are consist of angular velocity, linear acceleration and gravity, each of them is a three-dimensional vector in the smartphone frame. The elements of gravity vector are gravitational projections on the three axes of smartphone frame. As the gravity always points to the earth's core, this gravity vector actually represents the phone inclination to the ground or phone pose. Therefore, phone pose reflected by gravity is a good representation of phone carriage state.

As shown in figure 5, however, similar phone poses may be correspondent to different carriage states, which means that the single phone pose is not enough for representing the carriage state. As aforementioned, linear acceleration is the high frequency component of acceleration, both linear acceleration and angular velocity are sensitive to phone movement. Figure 6 shows the different walking signal patterns of linear acceleration under different phone carriage states, even their phone poses are similar. Therefore, we could leverage the human dynamics, such as walking, to further distinguish more phone carriage states by extracting features from linear acceleration and angular velocity.

Specifically, we extract time domain features from gravity to represent phone pose and extract both of time and frequency domain features from linear acceleration and angular velocity to capture dynamic information.

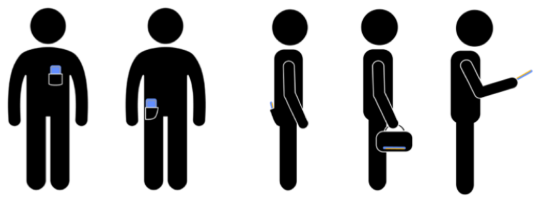


Figure 5. Phone poses and carriage states.

3.5. Deep Neural Network Classification Model

In the previous sections, we have discussed the usages of magnetic field intensity and TxPower, the histogram representation of BLE RSSI and features extraction from IMU data, however, how to use these processed sources to detect proximity remains unsolved. Unfortunately, the BLE signal propagation is an extremely sophisticated process which is impossible to manually analyze even if given the unchanged environment. Therefore, deep learning model is leveraged to detect proximity using these processed sources. Specifically, these sources are concatenated to a fixed-length vector which serves as the input of the deep learning model. Since the processed sources contain no spatial or temporal information, vanilla deep neural network with dropout layers is chosen.

As mentioned in section 3, the dataset provides different data sources, the ground truth distance and proximity thresholds. Both of the regression and classification model can be applied in proximity detection. In the training phase, the regression model uses the ground truth distance as label while the classification model converts the distance to event type by comparing it with the proximity threshold. In the testing phase, classification model directly outputs the event type while the regression model converts the predicted distance to event type. Basically, the main difference between these two models is when to convert distance to event type. We adopt classification model because the regression model has trouble dealing with the boundary points. Figure 7 illustrates the regression and classification model training process on a same boundary point. The coordinate represents the output distance and the red dotted line is the proximity threshold. Due to the classification model has two output units, we define the no contact score over contact event score to be pseudo-distance, in other words, greater pseudo-distance indicates greater probability of no contact. In the training phase, the regression model will optimize the prediction to the ground truth which is very close to the proximity threshold, therefore, it is likely to be wrongly classified with a little error. Whilst the classification model's ground truth has been converted from distance to event type which will optimize the prediction to be away from the proximity threshold.

4. Evaluation

Besides high accuracy, our design is also storage and computationally efficient.

Generally, it is infeasible for smartphone to employs deep learning model with IMU data in contact tracing. IMU data directly reflect the phone carriage state, though, smartphone could produce several gigabytes of IMU data in two weeks, which is a huge burden for smartphone storage. So, IMU data need to be processed once produced. Deep learning model is able to extract high level features, however, at the cost of computational efficiency due to its complex structure, frequently running deep learning model is impossible for smartphones. This dilemma, however, can be solved by our 2-stage model. As shown in figure 3, we leverage the light-weight algorithm to convert massive data into fixed-length vector in stage 1 and employ the complicated deep learning model to detect proximity in stage 2. Therefore, the smartphone can keep running the stage 1 and stores the vector into database. Only if the smartphone owner has the risk of exposure to virus, our model will fetch the vector from database and detect proximity. In this way, our design is both storage efficient and computationally efficient.

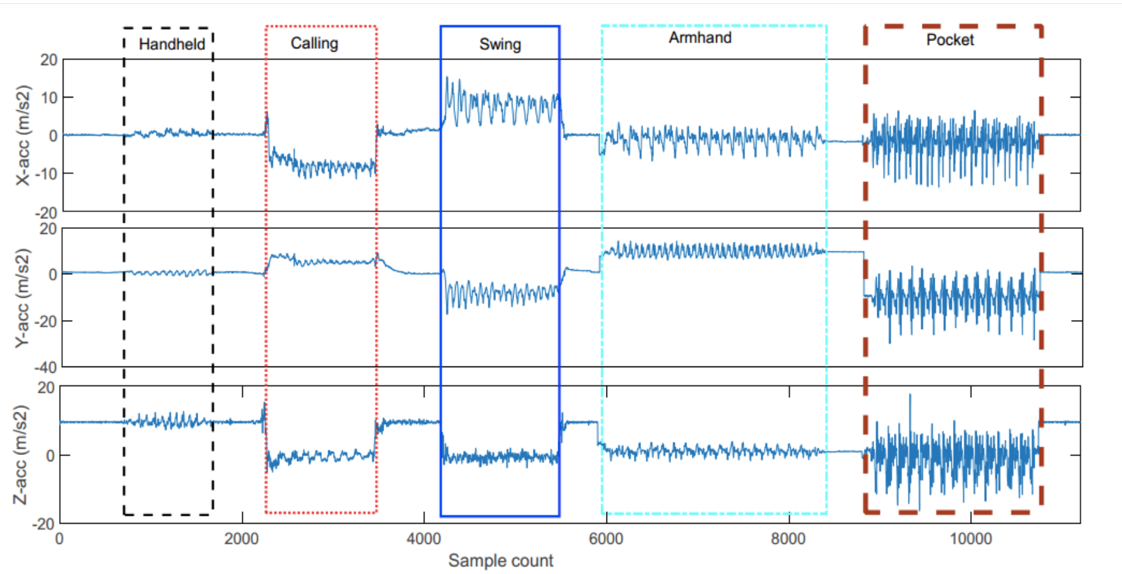


Figure 6. Linear acceleration under different phone carriage states.

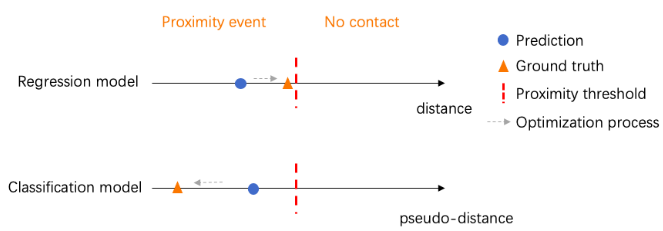


Figure 7. Illustration of the regression and classification model training process on a same boundary point. Pseudo-distance is defined as the probability of no contact over the probability of contact event.