DISTERNING: Distance Estimation using Machine Learning Approach for COVID-19 Contact Tracing and Beyond

Bohan Zhang, Ruiyu Wang, Hao Xu, *Member, IEEE*
Xiaoshuai Zhang, and Lei Zhang, *Senior Member, IEEE*

***Abstract*—Since the coronavirus disease 19 (COVID-19) outbreak, the epidemiological analysis has raised a strong requirement for more effective and accurate contact tracing solution. However, the existing contact tracing solutions either lacked the evaluation of tracing proximity or the features used for the tracing proximity evaluation were susceptible to certain negative environmental factors (e.g., body shielding). In this article, we propose a novel distance estimation algorithm based on machine learning for contact tracing: DISTERNING, where we leverage machine learning algorithms including Learning Vector Quantization, Regression, and Deep Feed-forward (DFF) Neural Network, data processing methods, and digital filters to process the Bluetooth signal information collected by the mobile phone for contact distance estimation. A contact tracing scheme based on edge computing is also proposed for algorithm deployment due to the requirements of the computational power. Compared with the existing contact tracing solutions, our algorithm considers the factors that have significant negative influence on the Bluetooth signal for distance estimation in reality. The evaluation results show that when the collected Bluetooth signal is influenced by real-world negative environmental factors, employing our proposed algorithm DISTERNING can keep the accuracy of the estimated distance reliable. The output distance can be combined with some medical models to conduct infection risk assessments.**

***Index terms*—** Distance estimation, contact tracing, machine learning, Bluetooth, pandemic, COVID-19.

I. Introduction

Since the outbreak in the winter of 2019, the coronavirus disease 19 pandemic (COVID-19) has been going on for over 2 years around the world. The enduring COVID-19 has caused 290,519,852 cases of infection and 5,445,804 death across 186 countries and territories in addition to tremendous economy shrink (including US by an annual rate of 4.8% in the first quarter of 2020 and by a shocking 32.9% in the second quarter) [1][2] to the date of 4th January, 2022. While the vaccine has been widely used, other non-pharmaceutical interventions (NPIs) including social distance [3], contact tracing, etc., are adopted by many countries to prevent the widespread COVID-19. As the primary method to alleviate the COVID-19 epidemic, strict NPIs such as quarantine can reduce the diffusion of COVID-19 by 99% under the synergy of vaccines [4]. However, such strict quarantine and lock down NPI measures may threat industrial production and people's work and daily life, which leads to global economy recession [5]. Compared with strict NPIs, social distance and contact tracing as milder NPIs are much more acceptable to prevent the spread of most respiratory infectious diseases [6].

Contact tracing is considered as an effective tool for monitoring social distance and as one of the most powerful public health interventions feasible in public places [7]. Digital contact tracing (DCT), i.e., using mobile phone apps to implement contact tracing and notification between individuals, has recently been proposed to be a plausible complement of manual contact tracing within the Test, Trace, and Isolate (TTI) containment strategies [8]. Existing DCT solutions usually depend on modules embedded in mobile phone (e.g., Bluetooth module) or extra hardware (e.g., intelligent reflecting surfaces) [9]. At present, many countries (e.g., United Kingdom, Singapore, and China, etc.) have deployed their digital contact tracing applications to assist epidemiological investigations which track the infection chain of COVID-19 and implement control measures to prevent further infections [10].

Current DCT applications deployed in various countries, including Bluetooth, GPS, and QR code technologies are mainly applied to realize proximity estimation. TraceTogether [11] of Singapore uses Bluetooth low energy (BLE) [12] to track the close contacts of infected persons. TraceTogether can be installed on mobile phones to scan the surrounding BLE devices in real time. It records the scanned Bluetooth signal strength (RSSI, Received Signal Strength Indication) in dBm for the proximity estimation of distance. Google Apple Contact Tracing [13] also relies on Bluetooth on mobile phones. Different from TraceTogether, this system utilizes Bluetooth to exchange keys representing users identities and considers all contacts in the Bluetooth broadcasting range as the close contacts. Once a person is confirmed to be infected, other close contacts can be notified. This system improves the privacy security with the help of individual keys but lacks relatively accurate proximity. China's health code system is built up on GPS, cellular base station, WiFi, Bluetooth, and the QR code provided by the health code app. In this system, a person's real-time health status is labelled by the three-color algorithm (i.e., green, yellow, and red) [14]. The system extracts the data from GPS, cellular base station, WiFi, and Bluetooth to monitor the locations and the time periods to deduce the activities of the infected person, and then adjusts the color of the health code of the close contacts who have been in the same place at the same time.

However, the existing digital contact tracing solutions cannot provide enough proximity estimation accuracy for contact tracing [15]. Commonly, GPS-based proximity estimation has a positioning accuracy of 10 meter (m) [16]. The possible 10

B. Zhang, R. Wang, H. Xu, X. Zhang, and L. Zhang are with the James Watt School of Engineering, University of Glasgow, UK; E-mail: 2429642z@student.gla.ac.uk; {R.Wang.1, H.Xu.2}@research.gla.ac.uk; {Xiaoshuai.Zhang, Lei.Zhang}@glasgow.ac.uk. This work is supported in part by the PETRAS National Centre of Excellence for IoT Systems Cybersecurity, which has been funded by the UK EPSRC under grant number EP/S035362/1. Corresponding Author: Lei Zhang.

TABLE I
A COMPARISON BETWEEN PREVIOUS SOLUTIONS AND PROPOSED NEW ALGORITHM

Name	Dependent technology	Environmental factors consideration	Proximity estimation	Distance conversion
TraceTogether [11]	Bluetooth	No	Yes, but not accurate	No
Google Apple Contact Tracking [13]	Bluetooth	No	No	No
China Health Code System [14]	GNSS, QR code, Cellular base station, WiFi, Bluetooth	No	No	No
DISTERNING	Bluetooth	Yes	Yes, accurate	Yes

m error is too large for contact tracing to detect the proximity between contacts. Bluetooth-based proximity estimation solutions also cannot provide enough proximity estimation accuracy for contact tracing under the consideration of certain negative environmental factors leading to inaccuracy of proximity estimation in real-world (e.g., body shielding) [17]. The inaccuracy of proximity estimation introduces significant errors in infection likelihood estimation, and hence, may lead to the failure of stopping epidemics [15]. Therefore, how to utilize mainstream mobile devices and their sensors (such as Bluetooth modules in smartphones) to implement contact tracing with enough proximity estimation for social distance detection should be further explored and addressed.

Since most previous Bluetooth-based solutions do not consider negative environmental factors in practice, our research focuses on Bluetooth-based distance estimation that includes the consideration of negative environmental factors between contacts to realize more accurate contact tracing. The estimated social distance can be combined with some medical research about the relationship between infection risk and distance (e.g., G. Cortellessa et al. [18]) to rank the potential cases for contact tracing. In Table I, several exemplars of mainstream contact tracing solutions, TraceTogether, Google Apple Contact Tracking, and China Health Code System, are compared with our algorithm DISTERNING in terms of the performance of proximity estimation from three aspects including environmental factors consideration, proximity estimation, and distance conversion (i.e., whether the solution estimates physical distance or not). Compared with the existing contact tracing solutions, DISTERNING provides a more accurate distance estimation between contacts while considering real-world environmental factors. Therefore, our proposed distance estimation algorithm possibly can be combined with the existing Bluetooth-based contact tracing solutions as a more reliable data processing and distance estimation component.

Considering the limitations of mobile devices in computation, we propose a contact tracing scheme based on cloud training and edge computing. In the scheme, we distribute the responsibilities of distance estimation and data collection to edge and users mobile phone respectively. The distribution of responsibility overcomes the high latency of cloud computing and low computational power of mobile phone while reducing the probability of private information leakage [19][20].

Our main contributions in this paper are highlighted as follows:

1) A machine learning powered Bluetooth distance estimation algorithm with two features (distance classification and conversion selection) for contact tracing called DISTERNING is proposed. The proposed algorithm not only considers the negative environmental factors caused by body shielding in contact tracing, but also outputs the estimated distance for the infection risk evaluation. The result shows that the average estimated distance absolute error can be reduced to around 0.3 m with the consideration of shielding when no noise is added. After noise is involved, the absolute errors of most distance ranges can be still kept within 1 m. Since there are far more medical studies on the relationship between infection risk and distance of infectious diseases than on the relationship between infection risk and Bluetooth signal, the concrete distance value estimated by our proposed algorithm DISTERNING can be applied to conclude infection risk by distance directly. Hence, DISTERNING is more universal and fundamental than other proximity estimation solutions thus can be used in preventing many other infectious diseases beyond COVID-19.
2) In view of the different data abundance caused by contact time, there are several metrics to evaluate the algorithm's performance. In this study, accuracy (absolute error between estimated distance and ground truth distance), latency (execution time on mobile devices e.g., smartphones), and stability (estimated distance absolute error variance for different distance ranges and body shielding categories) of the algorithm are considered as key performance indicators (KPIs). Quadrant diagrams representing different levels of these indicators provide a reference for users to judge the performance of the algorithm for different contact duration, and these results are discussed in Section VI-D.
3) For the proposed Bluetooth-based contact tracing algorithm, short contact time might lead to low data abundance and large algorithm error, this paper provides a solution to overcome this problem in Section VI-E. This provides a reference solution for fulfilling the high-accuracy requirement with limited data.
4) Due to the needs of machine learning model training and testing, we collect the mobile phone BLE RSSI data of two contacts at different distance with considering different mobile phone positions (under different body shielding categories). The collected data could be uploaded to Github as a new BLE RSSI data set for other studies in the future.

The rest of this paper is organized as follows. Some recent studies about contact tracing are introduced and analyzed in Section II. Then, the deployment scheme of our proposed DISTERNING algorithm is depicted in Section III. An overview of data collection, data labeling and algorithm architecture for DISTERNING is presented in Section IV. After that,

we illustrate the functions and implementations of our new algorithm DISTERNING in Section V and demonstrate our experimental results in the real environment in Section VI. In the next Section VII, the challenges and some potential countermeasures are discussed, which is followed by the final section to conclude our work.

II. Related Work

In previous studies, researchers analyzed the main negative environmental factors that Bluetooth signals undergo in real environments i.e., human body shielding and multi-path effect [21][22][23]. Some researchers suggest using some sensors embedded in mobile phones to mitigate the impact of these negative environmental factors [24], but these sensors can misjudge the actual negative environmental factors or fail to report the actual negative environmental factors (e.g., the light sensor of the mobile phone might not distinguish whether the mobile phone is in the backpack or in the pocket). Therefore, a more accurate solution of distance estimation is needed for COVID-19 contact tracing based on Bluetooth.

[17][25][26] have proposed machine learning based solutions for contact tracing or distance estimation. Su et al. [17] applied several machine learning algorithms including Support Vector Machines (SVM), Random Forest, and Gradient Boosted Machines (GBM) to realize distance estimation based on Bluetooth RSSI. Furthermore, they improved the accuracy of distance estimation by extracting a variety of Bluetooth signal characteristics. In their solution, the result is measured by the likelihood that the estimated distance and the actual distance between the two mobile phones are both greater than or less than 1.5 m. Even though the use of machine learning algorithms significantly improves the likelihood, and the distance range of the contact is accurately determined, there still exists a maximum possible error of 1.5 m. Meanwhile, this solution needs to obtain numerous Bluetooth signal characteristics, and some of them (such as the received Bluetooth signal frequency) have to be collected by extra tools. Such drawbacks prevent this solution from being applied for Bluetooth-based contact tracing currently. Sattler et al. [25] proposed a model that uses a linear classifier to assess infection risk of contacts. They performed a linear regression on the risk of infection with RSSI, time characteristics, and specified risk thresholds to deduce high or low infection risk on contacts. However, this model does not consider the shielding effect of the body when the mobile phone is put in a pocket or other different positions. In [26], optimized support vector machine and Kalman filter are applied for distance regression from BLE RSSI without considering negative environmental factors. Although the estimated distance absolute error can be reduced to 0.4 m, an obvious accuracy drop of distance estimation is observed when the BLE signal is blocked by objects.

As investigated above, these BLE-based solutions either lack of the consideration of different negative environmental factors in the real world or fail to mitigate the impact of negative environmental factors. However, the above results show a promising potential of machine learning in distance estimation based on BLE. Inspired by previous work, we proposed DISTERNING.

III. A Deployment Scheme of Distance Estimation algorithm on Edge

When machine learning methods are integrated in our proposed algorithm, the execution time for distance estimation increases as the amount of data to be processed increases (this result will be demonstrated in Section VI-B). Deploying such an algorithm for distance estimation on a mobile phone could occupy a large amount of memory. Meanwhile, the computational power of the mobile phone cannot meet the demand of fast distance estimation. A cloud deployment requires cloud computation component which is usually limited by the transmission bandwidth due to the remote deployment of workload. Compared to cloud computing, edge computing has a lower latency and could mitigate the bandwidth limit. Moving the workload closer to the user reduces the effect of limited bandwidth at a location [27]. Additionally, in terms of privacy, the cloud is a potential breaking point if the user data used for distance estimation is not managed properly. The edge computing can mitigate the privacy requirement [27] by providing alternative jurisdiction over the privacy-sensitive data, and a flexible model of computing and storage solutions. There are explorations on using user-centric edge computing [28] and resource sharing scheme powered by blockchain [29] with mutual authentication between users and the targeted edge computing instances. By employing the decentralized safety features, users can better protect their privacy while using the edge computing for speeding up data processing.

As an infrastructure that could improve the capabilities of personal portable devices effectively, mobile edge computing (MEC) [30] could provide sufficient computational resources, latency, responsiveness, and data security simultaneously when compared to local and cloud computing [31]. In addition, MEC reduces the occupation of mobile phone memory sigficantly. Therefore, we propose a scheme to deploy machine learning based algorithms on the edge-based computing platform to reduce the computing latency and ensure the data security.

Fig. 1 describes the deployment scheme that includes the privacy-preserving DCT solution and the proposed distance estimation machine learning algorithm. In the deployment scheme, the distance estimation model is trained on cloud by developers in the lab using the RSSI and distance data measured in real-world and then deployed in the edge server as requested by users, as shown in step 1. When a person is diagnosed by COVID test institution, a COVID test report is transmitted to the patient, as shown in step 2. Next, the patient sends his/ her encrypted Bluetooth ID to the privacy-preserving DCT solution and transmits encrypted Bluetooth ID and distorted proximity data (RSSI) to edge in step 3. Then, in the fourth step, the DCT solution transmits the exposure notification to contacts. After that, the contacts transmit their encrypted Bluetooth IDs and distorted proximity data to the edge, and the data is inputted into the distance estimation model on the edge for distance estimation as shown in step 5. Finally, the estimated distance is converted to the infection risk

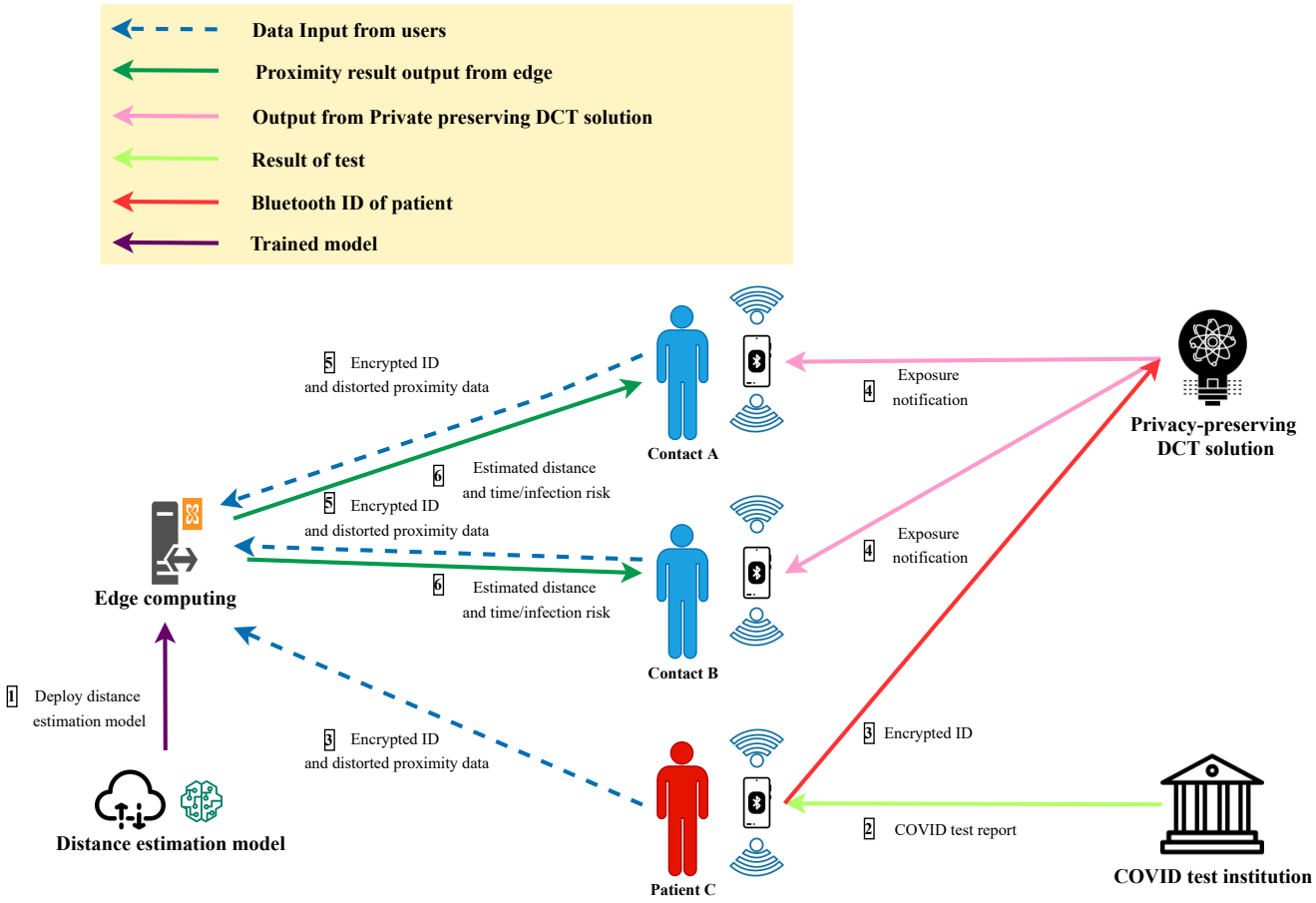


Fig. 1. Contact tracing schematic diagram.

by combining with the appropriate medical distance-infection risk model, and the infection risk is transmitted to contacts as denoted in step 6. Through the cloud training and the edge deployment, the computation speed for distance estimation is much faster compared with the local deployment (on the mobile phone). Furthermore, since the edge server only provides better computational power for distance estimation using distorted proximity data from encrypted ID, the protection of user privacy is also improved.

The data transmission between the edge and mobile phones of users can be supported by Long Term Evolution (LTE), which is supported by nearly all smart phones. It provides 75 Mbps upload bandwidth and 300 Mbps download bandwidth within 5 km coverage [32]. The transmitted BLE broadcasting packet is 31 bytes including the RSSI and timestamp [33]. Assuming the packet is transmitted 4 times per second, and the walking speed is 1 m per second, the contact duration is 9 seconds within the distance range 4.5 m (i.e., go close and then away), and 8928 bits are stored temporarily and then uploaded to the edge for one contact in the contact duration. The time required to upload the 8928 bits data is in milliseconds (0.12 milliseconds), which is negligible to a smart mobile phone that is connected to 4G communication networks. The notification from the DCT solution and institution can be realized by the mobile phone APP. The communication between the cloud and the edge is only required for the initial model deployment and the model update.

In this paper, we focus on the development of the distance estimation algorithm deployed on the edge, and use the collected data to train and test the algorithm.

IV. A Distance Estimation Algorithm Integrated by Learners

In DISTERNING, we combine clustering with supervised information and classification algorithms and use the data collected through experiments in the real environment to train and test an integrated learner algorithm. The data collected in the experiment is divided into two parts: a training set for training the algorithm of distance estimation and a testing set for testing our algorithm. After testing, Gaussian noise is added to the original data of the testing set to simulate the effect of other negative environmental factors to verify the robustness of our algorithm to verify the robustness of algorithm. Considering the overall operating equipment of the

algorithm, the trained machine learning algorithm model is deployed on the mobile phone to examine the actual running time recorded in Section VI-B.

As discussed in Section II, for the environmental factors, the body shielding and multi-path effect significantly influence the Bluetooth signal. The result of [22] shows that multi-path effect has a positive effect on reducing the signal attenuation caused by shielding and thus increases distance detection accuracy. Therefore, multi-path effect is not considered as a negative environmental factor and hence, is not considered in the algorithm. This algorithm mainly considers the signal absorption caused by body shielding. For other minor factors such as shielding caused by pass-by between contacts and the small difference on RSSI for different BLE chips with the same setting power since the transmitting power might not be exactly the same, Gaussian noise is added to the testing data to simulate these random factors [34][35]. For larger objects between contacts (such as walls), these objects can block the droplets from contacts and reduce the risk of infection. The significant shielding effect of the wall on RSSI will make a larger distance estimation by algorithm resulting in a low infection risk assessment, which is aligned with the real situation. Meanwhile, for other smaller objects on the signal propagation path, their influence is simulated by adding noise to the data set.

A. Raw Data Collection

For the contact tracing algorithm based on the Bluetooth signal strength, the most important environmental factor is the human body's shielding on the Bluetooth signal strength when the mobile phone is placed in different positions relative to the human body, as shown in Table II. During the collection of the original data, the human body's shielding effect on the Bluetooth signal is summarized in the following three conditions:

- There is no human body shielding between the two mobile phones, which is marked as 'no shielding'.
- One of the two mobile phones is on one side of the human body, and the other mobile phone is on the other side. This condition is marked as 'half shielding.
- One of the two mobile phones is located on the front (or back) of the human body, and the other mobile phone is located on back (or front). This is marked as 'full shielding'.

TABLE II
THE CATEGORIES OF BODY SHIELDING

Shielding category	Phone 1 position	Phone 2 position	Body between two phones
No shielding	—	—	No
Half shielding	Left/Right side	Right/Left side	Yes
Full shielding	Front/Back	Back/Front	Yes

Since two people need to use two mobile phones, the three possible body shielding categories for each person can be combined to form six shielding categories (combinations) shown in Fig. 3.

- No shielding-no shielding (e.g., two phones are held by two persons' hands separately).
- No shielding-half shielding (e.g., one phone is in hand, and the other one is placed in the pocket).
- No shielding-full shielding (e.g., one phone is in hand, the other one is placed in the backpack).
- Half occluded-half occluded (e.g., two phones are put in two persons' pockets separately).
- Half shielding-full shielding (e.g., one phone is put in the pocket, the other one is put in the backpack).
- Full shielding-full shielding (e.g., two phones are placed in two persons' backpacks separately).

In current studies, maintaining a social distance of more than 2 m is considered as the effective distance to reduce the infection risk of COVID-19 [36]. For this reason, the distance estimation range of the algorithm should be greater than 2 m. In free space, the effective transmission distance of BLE is usually 10 m, and the signal strength decays logarithmically with the distance. In the experiment, when the BLE transmitter is set to the maximum power, RSSI changes significantly with the distance in the range of less than 4.5 m. However, when the distance range is greater than 4.5 m, the relationship between RSSI and distance falls into the stable part of the logarithmic curve, hence, there is no obvious change in RSSI for different distances. Therefore, we choose the distance range at which the Bluetooth signal strength attenuates the fastest to satisfy the social distance requirement, that is, 0 to 4.5 m as the distance range of the design. Within this distance range, starting from 0 m, one experimenter stands still while the other moves at 0.5 m intervals in the direction shown in Fig. 2. Two people use two mobile phones to collect the Bluetooth signal strength and phone timestamps when the Bluetooth broadcast packet is received under the six shielding combinations mentioned above at each distance. For the RSSI collected on two mobile phones in each body shielding category at each distance, a total of 50 pairs of RSSI data are collected. In order to consider the fluctuation of RSSI in reality and get as much data as possible for testing and training, each RSSI recorded by one mobile phone is combined with each RSSI recorded by another mobile phone to form a new pair of data. Therefore, the total number of data in each body shielding category at each distance is therefore expanded into 2500 pairs.

B. Generation of Training Set and Testing Set

When the electromagnetic wave encounters an obstacle on the propagation path, a fan-shaped shadow area is formed behind the obstacle, in which the signal strength is attenuated. When the obstruction is close to the receiver or the transmitter, the signal strength received by the receiver is different. Therefore, we use the RSSI difference collected by the two mobile phones to characterize the influence caused by the position of the obstruction. This difference is utilized as an input to the machine learning algorithm to convert RSSI features to distance features. In order to show the overall impact of shielding and distance on RSSI, the average value of each pair of RSSI data is used to characterize this impact as another input.

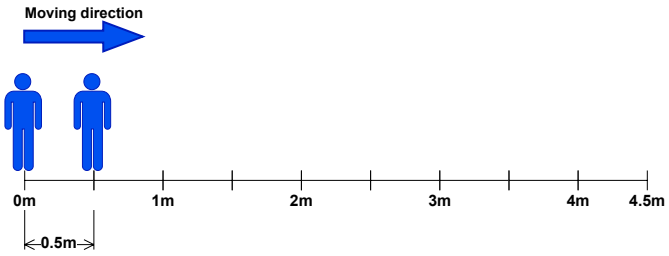


Fig. 2. Distance arrangement of data collection.

(a) no shielding-no shielding (b) no shielding-half shielding (c) no shielding-full shielding (d) half shielding-half shielding (e) half shielding-full shielding (f) full shielding-full shielding

Fig. 3. The demonstration of different body shielding categories.

The absolute difference and mean value of each pair of RSSI data are calculated as the training and testing data. For each body shielding category at each distance, 2500 pairs of new data containing these two features are generated. Meanwhile, each pair of data also includes the distance and the body shielding category label when it is measured. In these newly generated data, 2000 pairs are used as the training set, and 500 pairs are used as the testing set.

C. Design of DISTERNING Algorithm Architecture

The design of DISTERNING is illustrated as four algorithm components in Fig. 4, including input generation, RSSI data classification, conversion from RSSI features to distance features and final distance result selection. In the above four components, the machine learning algorithms and data processing methods are applied. The input of the algorithm is the RSSI difference and the mean value. Meanwhile, the output is the estimated average contact distance and minimum contact distance during the contact time between the patient and the contacts. The different components of DISTERNING adopt a cascading structure consisting of a classifier, distance regression, bias correction, a distance data filter, and distance feature selection. Since the applied machine learning algorithm is relatively complex as it involves multiple levels, this brings difficulties to the training of the machine learning algorithm as an entirety. Therefore, in the process of training and testing, each component is trained and tested separately. After each component is tested, they are integrated together to construct the entire algorithm. Finally, the test set is inputted into the algorithm to test its overall performance in terms of latency, distance estimation accuracy, and stability.

When the trained machine learning algorithm is applied, the user's mobile phone as a data collector only needs to collect the RSSI from the received Bluetooth broadcast packets, the corresponding Bluetooth name or assigned ID and the mobile phone timestamp. This process does not require reading other information on the mobile phone, which is beneficial for privacy-preserving. When contact tracing is required, the collected RSSI data, Bluetooth name or assigned ID and the timestamp are uploaded to edge, and the inputs of DISTERNING are generated in the same way as the training/testing set generation. Finally, some distance features of the corresponding Bluetooth ID are deduced from the inputs.

V. DISTERING Implementation

This section discusses the concrete function design and implementation of each block. As shown in Fig. 4, machine learning methods such as learning vector quantization (LVQ), regression, and DFF neural networks are applied to realize the component functions of the algorithm. In addition, other data processing methods like filtering are introduced to improve the distance estimation accuracy. The performance of each block is reflected on its output to the testing set as the input, and the overall performance is reflected on the final output of the algorithm.

When the algorithm is deployed in practice, the collected RSSIs from receivers and transmitters in the contact duration of two contacts are divided into several raw data pairs. Each raw data pair contains one RSSI from the receiver and one RSSI from the transmitter. The two RSSIs have the closest timestamp in the time scale, and their absolute difference and mean are calculated as one pair of the input. Assuming the BLE broadcasts at the maximum rate i.e., 4 times per second, the time interval between two broadcasting packets is 0.25 seconds. The timestamp provided by the mobile phone has a millisecond precision [37], which can meet the time matching requirement. The input set consists of all generated input pairs. Note that the input pairs have the same form as the pairs in the training set and the testing set.

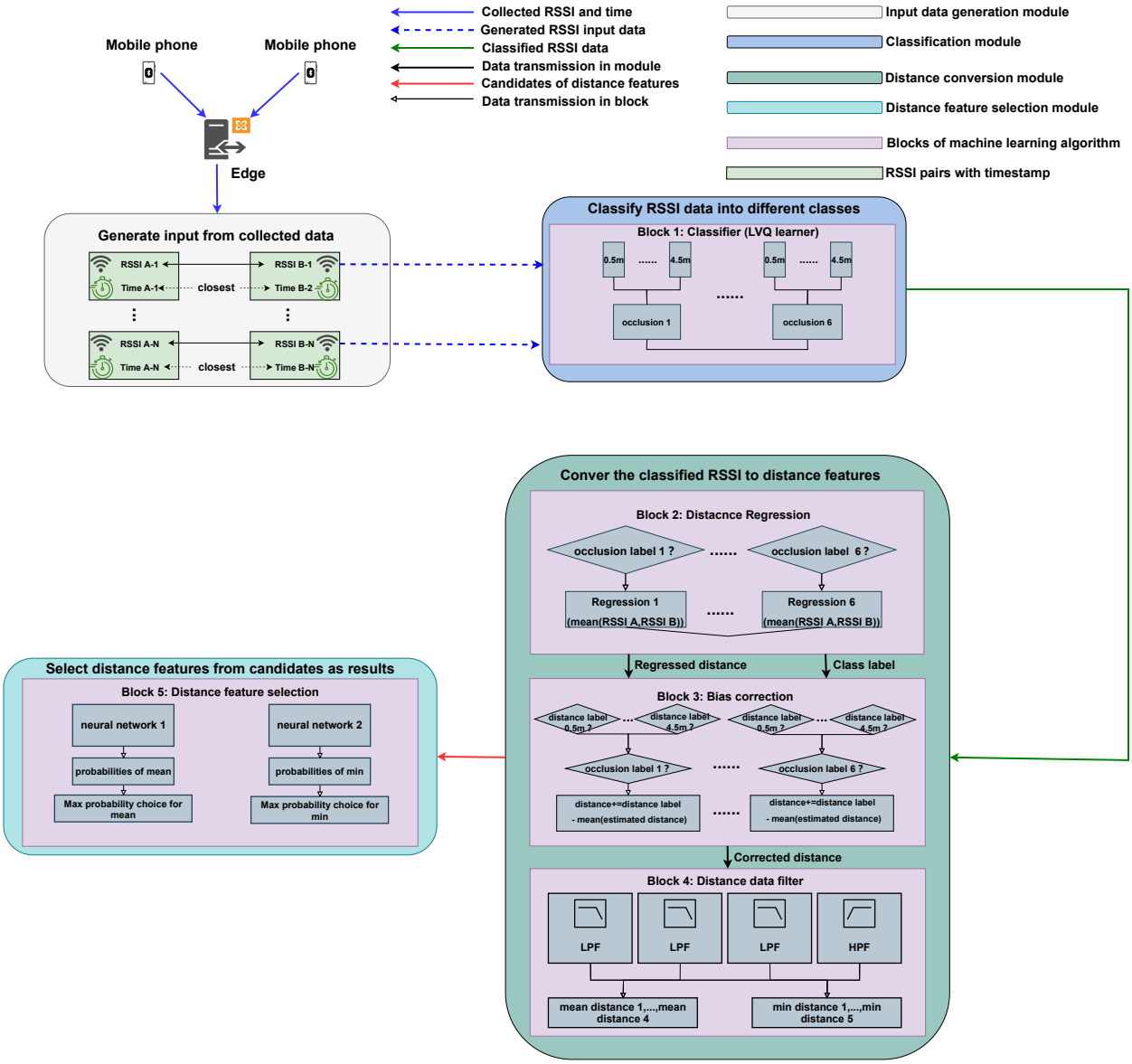


Fig. 4. Algorithm architecture of DISTERNING.

A. *Classifier*

The classifier shown in Block 1 of Fig. 4 is used to classify the input set into 54 categories corresponding to different body shielding and ground truth distance. Then classifier outputs the RSSI mean with the classification label to the next block. As a widely used machine learning algorithm for multiple classification, LVQ has the advantage of powerful classification ability and relative low requirements of storage space and calculation especially when it is applied to classify multi-dimensional data. Because of LVQ's characteristics of high efficiency and low computing cost, it is applied to realize the classification function.

The distance label in outputs will not be used in the next stage, but it will be used in the bias correction stage and can improve the performance of LVQ for the body shielding classification compared to classifying the input data into 6 body shielding categories directly (i.e., single prototype vector for each shielding category). This is because instead of using single prototype vector for each shielding classifications, multiple prototype vectors are applied for each body shielding category. Therefore, there are 54 categories corresponding to all combinations of 6 shielding and 9 distance range (i.e., 54 categories). After training, the input data is classified according to the prototype vector with the smallest Euclidean distance as shown in Equation (1.1)-(1.5):

$$g_j \in G = \{0.5m, \cdots, 4.5m\}, \tag{1.1}$$

$$h_k \in H = \{h_1, \cdots, h_6\}, \tag{1.2}$$

$$e_{i,g_j,h_k} = \left\| \mathbf{d}_i - \mathbf{p}_{g_j,h_k} \right\|, \ i = 1, \cdots, n, \tag{1.3}$$

$$e_{i,min} = \min \bigcup_{g_j}^{G} \bigcup_{h_k}^{H} \{ e_{i,g_j,h_k} \}, \tag{1.4}$$

$$\mathbf{d}_{i,g_i=g_j,h_i=h_k} \Leftarrow \mathbf{p}_{g_j,h_k},\ e_{i,g_j,h_k} = e_{i,min}, \tag{1.5}$$

where $\mathbf{d}_i$ is the two-dimensional input data vector (represented by bold and operate as column vectors) consisting of RSSI difference and RSSI average. $\mathbf{p}_{g_j,h_j}$ is the two-dimensional prototype vector including the shielding category and the classified distance label. g_j and h_j represent the ground truth distance and shielding category label while g_i and h_i are the labels of the input data vector $\mathbf{d}_i$. n is the size of the input data. G and H are the sets of classified distance and shielding category where h_1 to h_6 are the shielding categories. e_{g_j,h_j} and $e_{i,min}$ are the Euclidean distance and minimum Euclidean distance between the input vector and prototype vector, and Equation (1.5) represents the category map from prototype vector to data vector with classified label.

The performance of the LVQ learner is reflected by ROC (Receiver Operating Characteristic) and AUC (Area Under the Curve) of the learner. Compared with the overall error rate of the learner, the classification performance of each shielding category is more worthy of consideration. ROC, as a common performance metric of two-class learners, can reflect the generalization ability of the learner for classification problems. Therefore, this concept is applied to our design to indicate the classification performance of the LVQ learner for each shielding class. Since the classification of body shielding categories is a multi-classification problem, in order to enable ROC to be applied to the performance measurement for each shielding classification, we rearrange all the RSSI data in the testing set according to the Euclidean distance from each prototype vector from small to large (i.e., in the order from the most likely to the least likely to be this category) as shown in Equation (1.6):

$$D = \{\mathbf{d}_1, \cdots, \mathbf{d}_n\},\ e_{1,min} < \cdots < e_{n,min}. \tag{1.6}$$

If the classification result of an input data vector in testing set is consistent with its shielding label, the vector is considered to be true positive sample; otherwise, the vector is considered to be a false positive sample as in Equation (1.7)-(1.10):

$$TP = \bigcup_{i=1}^{n_{tp}} \left\{\mathbf{d}_{i,g_i=g_{i_t} \cap h_i=h_{i_t}}\right\}, \tag{1.7}$$

$$FP = \bigcup_{i=1}^{n_{fp}} \left\{\mathbf{d}_{i,g_i\neq g_{i_t} \cup h_i\neq h_{i_t}}\right\}, \tag{1.8}$$

$$TN = \bigcup_{i=1}^{n_{tn}} \left\{\mathbf{d}_{i,g_i=g'_{i_t} \cup h_i=h'_{i_t}}\right\}, \tag{1.9}$$

$$FN = \bigcup_{i=1}^{n_{fn}} \left\{\mathbf{d}_{i,g_i\neq g'_{i_t} \cap h_i\neq h'_{i_t}}\right\}, \tag{1.10}$$

where TP and FP are the true and false positive input data sets, and TN and FN are the true and false negative sets. g_{i_t} and h_{i_t} are the ground truth distance and truth shielding categories while g'_{i_t} and h'_{i_t} are the complement of g_{i_t} and h_{i_t} with respect to G and H. n_{tp}, n_{fp}, n_{tn} and n_{fn} are the size of sets.

In order to verify the performance improvement of multiple prototype vectors compared to single prototype vector, the AUCs of two types of learners are shown in Table III, where the six body shielding categories are marked as A-F. The two axes of the ROC curve are calculated according to Equation (1.11-1.12):

$$TPR = \frac{TP}{TP + FN}, \tag{1.11}$$

$$FPR = \frac{FP}{TN + FP}, \tag{1.12}$$

where TPR and FPR are the true positive rate and false positive rate.

For most body shielding categories, the AUCs of the two learners are similar. However, for the shielding categories D and E, there is relatively significant difference on AUC for two types of learner. The comparison of the ROC curves for two LVQ learners in terms of shielding class D and E is depicted in Fig. 5. Furthermore, the average AUC of the learner with multiple prototype vector for different shielding categories is greater than the average AUC of the learner with single prototype vector. The results imply that the distance label can effectively improve the performance of the classifier for body shielding classification.

TABLE III
AUC REUSULTS OF TWO LVQ LEARNERS CONTAINING MULTIPLE OR SINGLE PROTOTYPE VECTORS FOR EACH BODY SHIELDING CATEGORY

Shielding category	AUC-multiple prototype vector	AUC-single prototype vector
A	0.8250	0.8185
B	0.7498	0.7929
C	0.8444	0.8815
D	0.9447	0.8793
E	0.7295	0.6622
F	0.8985	0.9094
Average	0.8320	0.8239

B. Distance Regression

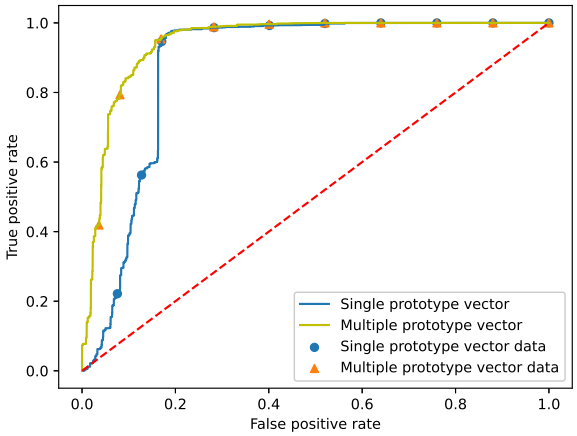
The distance regression shown in Block 2 of Fig. 4 converts RSSI into distance through regression. In this process, the input is the vector $\mathbf{r}_{i,h}$ consisting of mean RSSI with the classified shielding category label of the input data vector as shown in Equation (2.1)-(2.2):

$$\mathbf{d}_{i,g_i,h_i} = (|R_{i,A} - R_{i,B}|, \frac{1}{2}(R_{i,A} + R_{i,B}), g_i, h_i), \tag{2.1}$$

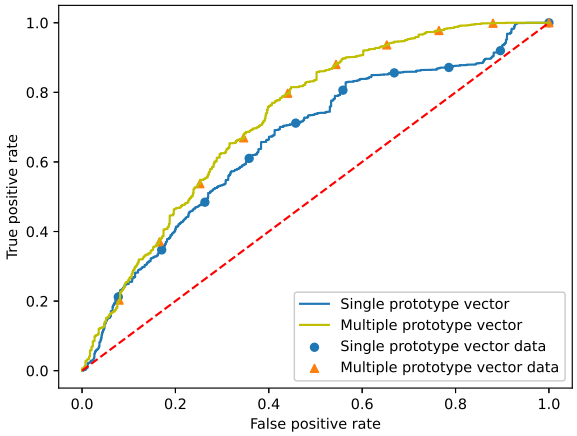
$$\mathbf{r}_{i,h_i} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{d}_{i,g_i,h_i} = (\frac{1}{2}(R_{i,A} + R_{i,B}), h_i), \tag{2.2}$$

where $R_{i,A}$ and $R_{i,B}$ are the received RSSI from two mobile phones.

For the 6 body shielding categories, each shielding category has a corresponding distance regression equation, which is selected from three candidate regression equations according



(a) ROC comparison of the body shielding category D.



(b) ROC comparison of the body shielding category E.

Fig. 5. ROC comparison of the categories D and E of body shielding with two types of LVQ learner.

to the maximum correlation coefficient on the training set by the method of ordinary least squares as in Equation (2.3):

$$f_{h_i}(x) = \begin{cases} ax + b, & R_l^2 = \max(R_l^2, R_i^2, R_p^2) \\ \frac{a}{x} + b, & R_i^2 = \max(R_l^2, R_i^2, R_p^2) \\ ax^2 + bx + c, & R_p^2 = \max(R_l^2, R_i^2, R_p^2) \end{cases}, \quad (2.3)$$

where a, b, and c are constants to be determined. $f_{h_i}(x)$ is the regression equation of the shielding category h_i, and R_l^2,R_i^2,R_p^2 are the correlation coefficients of linear, inverse proportional and polynomial regression equations.

The first dimension of input vector is considered as the independent variable of the trained regression equation to convert RSSI to distance as shown in Equation (2.4):

$$dist_{i,g_i,h_i} = f_{h_i}(\begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{r}_{i,h}), \quad (2.4)$$

where $dist_{i,g_i,h_i}$ is the regressed distance.

To ensure that the regression equation obtained by training has high accuracy, Kalman filter is used to eliminate the noise before training, shown in Algorithm 1. In the filtering process, the RSSI mean $r_i, i = 1, \ldots, n$ in all pairs of training set forms the observation matrix $O = (r_1, \ldots, r_n)$. Its average value and variance are regarded as the expectation μ_o and the variance σ_o^2 of the observation matrix normal distribution $N(\mu_o, \sigma_o^2)$. The differences between every two adjacent RSSIs in observation matrix consist of the measure matrix $M = (r_2 - r_1, \ldots, r_n - r_{n-1})$. The average value and variance of M are used as the expectation μ_m and variance σ_m^2 of the measure matrix normal distribution $N(\mu_m, \sigma_m^2)$. The variance $\sigma_{p_i}^2$ of these two distributions is considered as the error of the corresponding quantity. The expectations μ_{p_i} and errors of their distributions are used to derive the prediction matrix distribution $N(\mu_{p_i}, \sigma_{p_i}^2)$ which is formed by predicted RSSIs denoted by $P = (Predict_1, \ldots, Predict_n)$. The variance of the prediction matrix is initialized to 0 and iterated under the influence of the observation matrix and the measurement matrix. The state matrix $S = (State_1, \ldots, State_{n-1})$ is composed of intermediate variables generated in the filtering process, which is used to connect the observation matrix, the measurement matrix and the prediction matrix as shown in Equation (2.5)-(2.6)

$$\begin{aligned} N(\mu_{s_i}, \sigma_{s_i}^2) &= N(\mu_{p_i}, \sigma_{p_i}^2) + N(\mu_m, \sigma_m^2) \\ &= N(\mu_{p_{i-1}} + \mu_m, \sigma_{p_{i-1}}^2 + \sigma_m^2), \end{aligned} \quad (2.5)$$

$$\begin{aligned} N(\mu_{p_i}, \sigma_{p_i}^2) &= N(\mu_{s_{i-1}}, \sigma_{s_{i-1}}^2) \times N(\mu_o, \sigma_o^2) \\ &= N(\frac{\sigma_{s_{i-1}}^2 \mu_o + \sigma_o^2 \mu_{s_{i-1}}}{\sigma_{s_{i-1}}^2 + \sigma_o^2}, \frac{\sigma_{s_{i-1}}^2 \sigma_o^2}{\sigma_{s_{i-1}}^2 + \sigma_o^2}). \end{aligned} \quad (2.6)$$

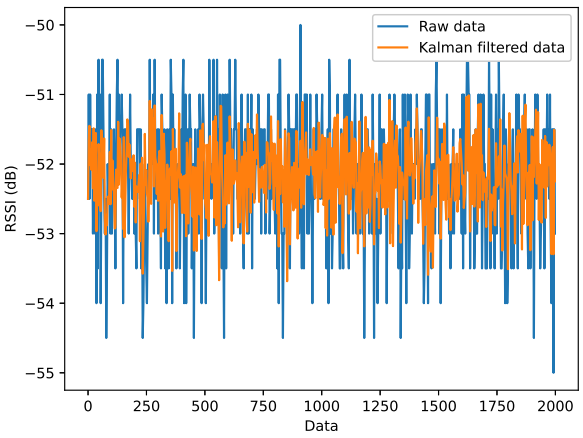
After the Kalman filter, the prediction matrix (i.e., the filtered RSSIs) is outputted as in Fig. 6, and these filtered RSSIs are regressed with their ground truth distance labels.

Algorithm 1 Kalman filter for RSSI

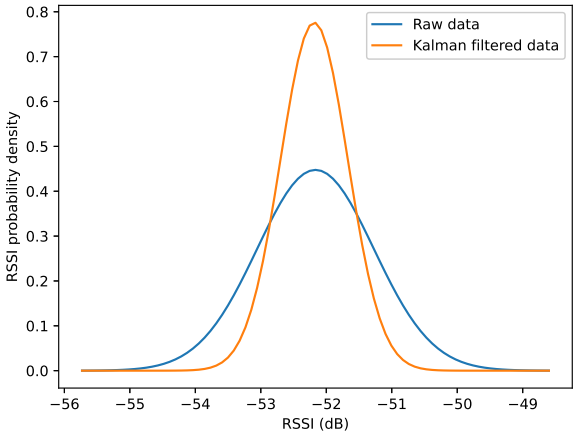
1: Initialization: O, M, P, S;
2: $Predict_1 = r_1$
3: $\sigma_{p_1}^2 = 0$
4: $i = 2$
5: **repeat**
6: $State_{i-1} = Predict_{i-1} + r_i - r_{i-1}$;
7: $\sigma_{s_{i-1}}^2 = \sigma_{p_{i-1}}^2 + \sigma_m^2$; based on Equation (2.7)
8: $Predict_i = State_{i-1} \times \frac{\sigma_0^2}{\sigma_0^2 + \sigma_{s_{i-1}}^2} + r_i \times \frac{\sigma_{s_{i-1}}^2}{\sigma_0^2 + \sigma_{s_{i-1}}^2}$;
9: $\sigma_{p_i}^2 = \frac{\sigma_{s_{i-1}}^2 \times \sigma_o^2}{\sigma_{s_{i-1}}^2 + \sigma_o^2}$; based on Equation (2.8)
10: i+=1;
11: **until** $i > n$
Output: predict matrix P

C. Bias Correction

In an ideal situation, there is no classification error in the LVQ classifier and the regressed distance fluctuate around its ground truth distance label (i.e., the average value is very close to the ground truth distance label). However, classifier and regression are not perfect. The errors caused by classification, regression, and environmental factors in real world make the average regressed distance have a certain bias compared with



(a) The comparison of the raw RSSI data and the filtered RSSI data.



(b) PDF comparison of raw RSSI data and filtered RSSI data.

Fig. 6. Example of using Kalman filter to preprocess RSSI data for distance regression.

its ground truth distance label. The bias changes with the distance range and the body shielding categories so that a filter could not adapt to this varied bias. Therefore, bias correction is introduced as a dynamic mechanism to reduce this bias.

The workflow of the bias correction shown in Block 3 of Fig. 4 is demonstrated in Fig. 7. The bias correction takes the input vector $\mathbf{c}_{i,g_j,h_k}$ consisting of regressed distance of the input data vector and the distance label as its input. The regressed distance is adjusted by adding bias which is the difference between the average regressed distance and the distance label of data of each category as shown in Equation (3.1)-(3.6):

$$\mathbf{c}_{i,g_j,h_k} = (dist_{i,g_j,h_k}, g_j), \tag{3.1}$$

$$\mathbf{C}_{g_j,h_k} = \bigcup_{i=1}^{n_{jk}} \left\{\mathbf{c}_{i,g_j,h_k}\right\}, \tag{3.2}$$

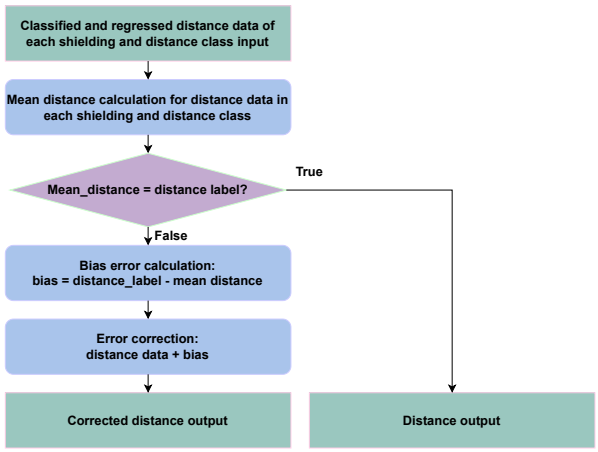


Fig. 7. Workflow of the bias correction.

$$b_{g_j,h_k} = \frac{1}{n_{jk}} \begin{bmatrix} -1 & 1 \end{bmatrix} \mathbf{C}_{g_j,h_k} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{n_{jk}\times 1}, \tag{3.3}$$

$$\mathbf{B} = \begin{bmatrix} dist_{1,g_1,h_1} & \cdots & dist_{n_{11},g_1,h_1} & b_{g_1,h_1} \\ \vdots & \cdots & \vdots & \vdots \\ dist_{1,g_9,h_6} & \cdots & 0 & b_{g_9,h_6} \end{bmatrix}, \tag{3.4}$$

$$\mathbf{M} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & 1 \\ 1 & \cdots & 1 \end{bmatrix}, \tag{3.5}$$

$$\mathbf{A} = \mathbf{BM} = \begin{bmatrix} a_{1,g_1,h_1} & \cdots & a_{n_{11},g_1,h_1} \\ \vdots & \cdots & \vdots \\ a_{1,g_9,h_6} & \cdots & b_{g_9,h_6} \end{bmatrix}, \tag{3.6}$$

where $\mathbf{C}_{g_j,h_k}$ is the set of the input vector corresponding to the classified distance g_j and the shielding category h_k, and n_{jk} is its size. b_{g_j,h_k} represents the bias of the category, and the last matrix in Equation (3.3) is an all-ones matrix with a size of $n_{jk} \times 1$. $\mathbf{B}$ is the bias matrix including the regressed distance of all categories and the calculated bias with the size of $54 \times (\max(size(\mathbf{C}_{g_1,h_1}), \cdots, size(\mathbf{C}_{g_9,h_6})) + 1)$. The size of regressed distance of each category might be different due to the different size of the classification for each category, and the vacancy in each row is occupied by 0. $\mathbf{M}$ is the matrix to add the regressed distance with bias including a unit vector and all-ones in the last row. $\mathbf{A}$ is the corrected distance matrix involving the corrected distance a_{i,g_j,h_k}.

D. Distance Data Filter

After the bias in the distance data is corrected, other noises in the distance data still affects the final distance feature extraction. Therefore, four Butterworth filters including three low-pass filters and one high-pass filter shown in Block 4 of

Fig. 4 are applied to reduce these noises. An example of a Butterworth low-pass filter is shown in Equation (4.1):

$$H(s) = \frac{\Omega_c^N}{\prod_{l=1}^{N}(s - \Omega_c e^{j[\pi(N+2l-1)/2N]})}, \tag{4.1}$$

where Ω_c, $H(s)$, s are the cut-off frequency, system function, and frequency while N is the order of the filter. Additionally, the cut-off frequencies of the four used filters are normalized to unit, and trial-and-error method is implemented on training set to determine the cut-off frequency of each filter by reducing the cut-off frequency from unit. In the process of changing cut-off frequency, the mean and minimum values of the filtered distance are calculated until the mean and minimum of the filtered distance have the least difference from the mean and minimum of the ground truth distance.

In this stage, the input is the corrected regression distance as shown in Equation (4.2):

$$a(t) = \bigcup_{k=1}^{6}\bigcup_{j=1}^{9}\bigcup_{i=1}^{n_{jk}}\left\{a_{i,g_j,h_k}\right\}, \tag{4.2}$$

where $a(t)$ is the input sequence consisting of the corrected distance.

Since the noise frequency mainly affects the accuracy of distance estimation varies with the distance between the contacts, the cut-off frequency of the filter is also different. This varied noise is caused by multiple reasons, such as the fluctuation of RSSI and the error caused by the processing in former blocks. Furthermore, the logarithmic relationship between RSSI and distance also introduces different frequency noise on distance regression. For the fixed fluctuation on RSSI, the regressed distance changes gently when the contacts are close due to the rapidly changing part of the logarithm relationship between RSSI and distance. The regressed distance changes significantly when the contacts are far apart due to the stable part of the logarithm relationship. The significant change in regressed distance leads to more high-frequency noise with a higher smallest frequency, whereas the gentle change in regressed distance leads to more high-frequency noise with a lower smallest frequency.

Because of the different frequency noise, four situations of the distance relationship between two contacts are discussed to choose different cut-off frequencies for the applied filters. Firstly, when two contacts are relatively close (e.g., 0.5-1.5 m), the distance regression of the algorithm contains more high-frequency noise with a lower smallest frequency, which has a major influence on the distance characteristics. Secondly, when the contacts are at a middle distance (e.g., 1.5-2.5 m), the smallest frequency of the high-frequency noise having the main influence on the distance characteristics increases. Thirdly, with the distance between contacts increasing (e.g., 2.5-3.5 m), the smallest frequency of the high-frequency noise in the regressed distance that has the main influence further increases. Lastly, when the contact persons are far away (e.g., 3.5-4.5 m), the distance filtered by the low-pass filter cannot directly extract the required distance features. Therefore, a pair of filters including a high-pass filter and a low-pass filter are used to obtain more accurate estimated distance features from the distance data.

In our proposed algorithm, the three low-pass filters have three cut-off frequencies marked as low, medium, and high, respectively. Meanwhile, the high-pass filter uses the same cut-off frequency with the highest cut-off frequency of the three low-pass filters. When the distance between contacts are far away, the average values of the filtered distance through the low-pass filter with the highest cut-off frequency and the high-pass filter are added together to perform the overall frequency distance estimation so that it is large enough to approximate the large ground truth distance. To calculate the estimated distance characteristics i.e., mean distance and minimum distance, the average values of the distance data filtered by three low-pass filters, respectively, the sum of the average distance value filtered by the low-pass filter with the highest cut-off frequency and the high-pass filter are considered as four candidates for the mean distance. Five candidates of the minimum distance are composed of the average values of the distance data through the three low-pass filters respectively subtracting the standard deviations, the average value of the distance data filtered by the low-pass filter with the highest cut-off frequency, the sum of the average value of the distance data filtered by the low-pass filter with the same highest cut-off frequency and the average value of the distance data filtered by the high-pass filter with the same highest cut-off frequency. These candidates of the mean and minimum distance are the outputs of this stage as shown in Equation (4.3)-(4.5):

$$f \in \mathbf{F} = \{low, middle, high\}, \tag{4.3}$$

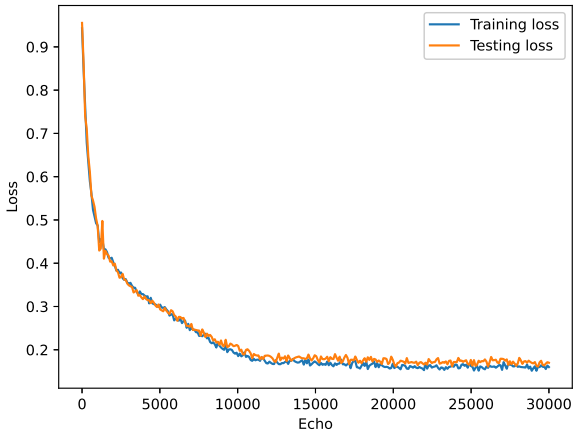
$$\begin{aligned}\mathbf{Q} = \bigcup_{f\in\mathbf{F}} &\left\{\frac{1}{n}\sum_{t=1}^{n} a(t) * h_{LP,f}(t)\right\} \cup \\ &\left\{\frac{1}{n}\sum_{t=1}^{n} a(t) * (h_{LP,high}(t) + h_{HP}(t))\right\},\end{aligned} \tag{4.4}$$

$$\begin{aligned}\mathbf{R} = \bigcup_{f\in\mathbf{F}} &\left\{\frac{1}{n}\sum_{t=1}^{n} a(t) * h_{LP,f}(t) - \right. \\ &\left.\sqrt{\frac{1}{n}\sum_{m=1}^{n}(a(m) * h_{LP,f}(m) - \frac{1}{n}\sum_{t=1}^{n} a(t) * h_{LP,f}(t))^2}\right\} \cup \\ &\left\{\frac{1}{n}\sum_{t=1}^{n} a(t) * h_{LP,high}(t)\right\} \cup \\ &\left\{\frac{1}{n}\sum_{t=1}^{n} a(t) * (h_{LP,high}(t) + h_{HP}(t))\right\},\end{aligned} \tag{4.5}$$

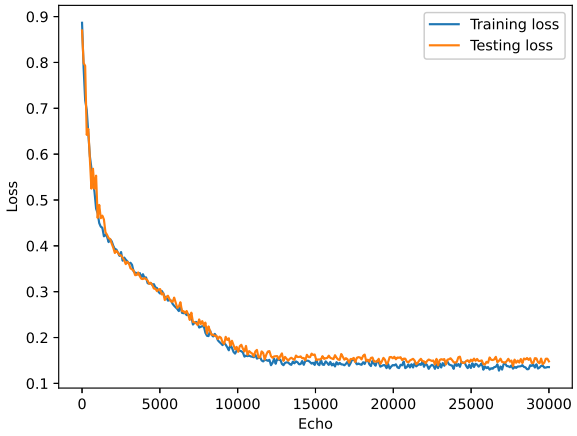
where $\mathbf{F}$ is the set of marks indicating the cut-off frequency of filters. $\mathbf{Q}$ and $\mathbf{R}$ are the sets of mean distance candidates and minimum distance candidates. $h_{LP,f}(t)$ represents the low-pass filter with its mark while $h_{HP}(t)$ is the high-pass filter.

E. Distance Feature Selection

The distance feature selection shown in Block 5 of Fig. 4 is used to select one value from the four candidates of average distance and one value from the five candidates of



(a) Loss of the applied DFF neural network for the mean distance estimation.



(b) Loss of the applied DFF neural network for the minimum distance estimation.

Fig. 8. Loss of the applied DFF neural network for the mean and minimum distance estimation.

minimum distance as the final estimated average distance and minimum distance respectively. However, this selection is established on finding the inherent pattern of these distance candidates, which might change with the distance between contacts, so such a pattern is quite complicated to be analyzed. As an effective data analysis tool, the neural network can be adopted to construct a structure conforming to any data change patterns. Therefore, in this algorithm, two Deep Feed-forward (DFF) neural networks adopting the error Back Propagation (BP) training algorithm for fitting the data change patterns of the average distance candidates and the minimum distance candidates are built to select the estimated average and minimum distance from the distance candidates. The input set $\mathbf{X} = \mathbf{Q} \cup \mathbf{R}$ consists of all the candidates of average distance and minimum distance. Since the average distance candidates and the minimum distance candidates have two common elements, there are a total of seven inputs imported into the two neural networks. The two neural networks output the probabilities corresponding to the four average distance candidates and the probabilities corresponding to the five minimum distance candidates. Then, the value with the largest probability from the four average distance candidates and the value with the largest probability from the five minimum distance candidates are selected as the estimated average and minimum distance, respectively as shown in Equation (5.1)-(5.2):

$$mean = q_i,\ P(q_i) = \max \bigcup_{q \in \mathbf{Q}}^{\mathbf{Q}} \{P(q)\}, \tag{5.1}$$

$$minimum = r_i,\ P(r_i) = \max \bigcup_{r \in \mathbf{R}}^{\mathbf{R}} \{P(r)\}, \tag{5.2}$$

where q_i and r_i are the estimated average and minimum distance with the largest probability $P(q_i)$ and $P(r_i)$.

The DFF neural network structure used for the judgment of the average distance and the minimum distance is composed of 4 layers of neurons, including an input layer, two hidden layers and an output layer (Table IV). The neurons in each layer adopt a fully connected structure. The tanh function is used as the activation function of the hidden layer since the tanh function has high sensitivity. The activation function of the output layer is softmax which can output a probability value in the range between 0 and 1.

TABLE IV
THE STRUCTURE OF THE DFF NEURAL NETWORKS

Structure	Mean distance network	Minimum distance network
Input layer nodes	7	7
Hidden layer 1 nodes	28	35
Hidden layer 2 nodes	20	20
Output layer nodes	4	5
Hidden layer activation	tanh	tanh
Output layer activation	softmax	softmax

Since the input of the neural network depends on the output of the previous stage, after the previous stage is trained, the training set is inputted to the previous stage, and the corresponding output is obtained as the training set of this stage. In the training process, the output labels of the training set for judging the average distance and the minimum distance are composed of 4 and 5 bits one-hot codes, respectively. Each bit corresponds to the candidates of the average distance and the minimum distance. In each set of one-hot codes, the mean and minimum distance value marked as 1 is the candidate with the least difference from the mean and minimum ground truth distance among the distance candidates.

The training result of the neural network for mean distance estimation is reflected by the loss function on the testing set according to Equation (5.3):

$$Loss = \frac{1}{M} \sum_{i=1}^{M} (\mathbf{y}_i - \mathbf{p}_i)(\mathbf{y}_i - \mathbf{p}_i)^{\mathbf{T}}, \tag{5.3}$$

where $\mathbf{y}_i$ is the one-hot code vector corresponding to the distance candidate with the least difference from the ground truth distance with the size of $1 \times c$ where c is the number of distance candidate. $\mathbf{p}_i$ is the selection probability matrix with

size $1 \times c$ for each candidate outputted by the neural network. M is the number of all combinations for each body shielding category at each distance, and the superscript $\mathbf{T}$ means the transpose of the matrix. By calculating the loss function on the training set and the testing set shown in Fig. 8, the neural network can converge to a low loss without obvious overfitting. It means that the neural network can select the final values of average and minimum distance from the distance candidates precisely.

VI. Results and Discussion

A. Robustness Testing

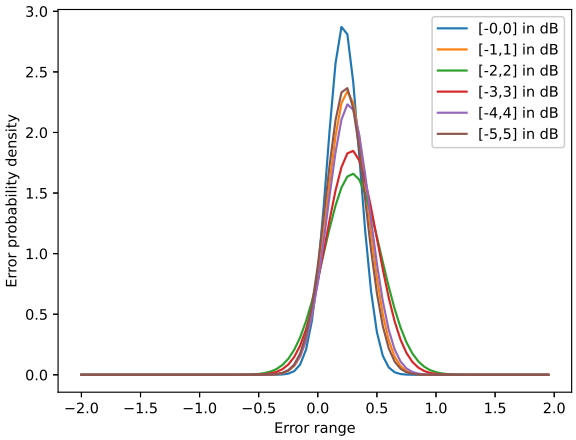
In practice, other possible negative environmental factors other than human body shielding. As discussed before, some researchers have verified the possibility of modeling obstacles by Gaussian noise. Meanwhile, the transmitting power of BLE chips can be set to the same while the actual power might not be exactly identical. The small difference in RSSI as a result of the inconsistent power is also simulated by the Gaussian noise. Therefore, Gaussian noise is added to the collected raw data of the testing set. Then, the raw data is used to generate the input of the DISTERNING algorithm to test the robustness of the algorithm. Additionally, some existing researches have proposed solutions for the RSSI difference of BLE chips [25].

As shown in Fig. 9, standard normal noise with different truncating ranges is added to the original data and inputted into the input data generation module of the DISTERNING algorithm. The absolute differences between the estimated average distance and the average ground truth distance and between the estimated minimum distance and the minimum ground truth distance are calculated as errors. The errors for various shielding categories and different distance ranges are plotted as a probability density distribution function (PDF). We apply Truncated Normal Distribution (TND) with different bounds, such as $[-3, 3]$ means the truncating range for TND is from -3 dBm to 3 dBm, and the range $[-0, 0]$ means there is no noise added. Overall, our machine learning-based algorithm shows strong robustness to negative environmental factors. When the noise truncating range increases, the errors of the estimated mean distance and minimum distance are almost kept within 1 m, and for most cases, the errors are in the range of 0-0.5 m. Since above 1 meter is considered as a safe social distance according to the particle image of droplets from contacts at different distances [36][38], the algorithm can provide an accuracy that can meet the application requirements. Meanwhile, the noise immunity of the algorithm also proves the feasibility of adding noise to distort data to enhance privacy protection.

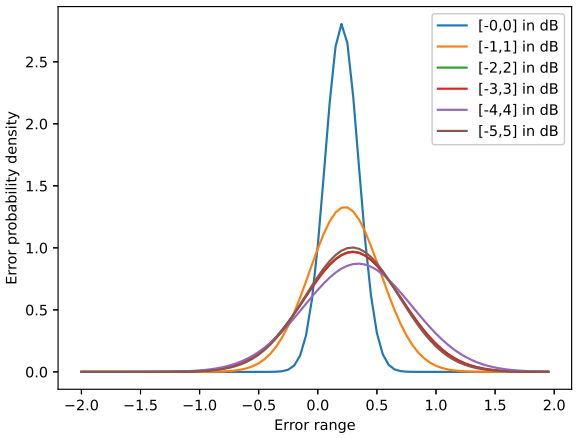
B. Execution Time and Calculation Complexity

The calculation complexity of the algorithm (i.e., the number of calculations) is determined by the number of additions and subtractions and the number of cycles that the algorithm performs for the distance estimation. The calculation complexity $C(n)$ of the algorithm is represented by Big-O time complexity notation [39] as Equation (6.1):

$$C(n) = 405n + 2097, \tag{6.1}$$



(a) PDF for mean distance.



(b) PDF for minimum distance.

Fig. 9. Probability density distribution function of estimated distance error with truncated standard normal noise.

which includes $216n+1039$ additions or subtractions, $135n+1054$ multiplications or divisions and $54n + 4$ square roots. Note that we have assumed that each addition, multiplication and square root operations have the same complexity for simplifying the analysis [40].

To test the execution time of the machine learning algorithm on the mobile device, we run the algorithm on a mobile phone (HUAWEI Honor V10) by a phone-based python interpreter QPython to fit the running time with the calculation complexity. The result shown in Fig. 10 illustrates that the running time of the proposed algorithm on the mobile phone is positively proportional to the calculation complexity. With the calculation complexity increasing, the execution time of our machine learning algorithm on the mobile phone might increase to ten seconds which is unacceptable for some quick responsive demands. Therefore, it is recommended to deploy the machine learning algorithm in edge and only use the mobile phone as a data collection device to achieve the low latency and high execution speed (Section III).

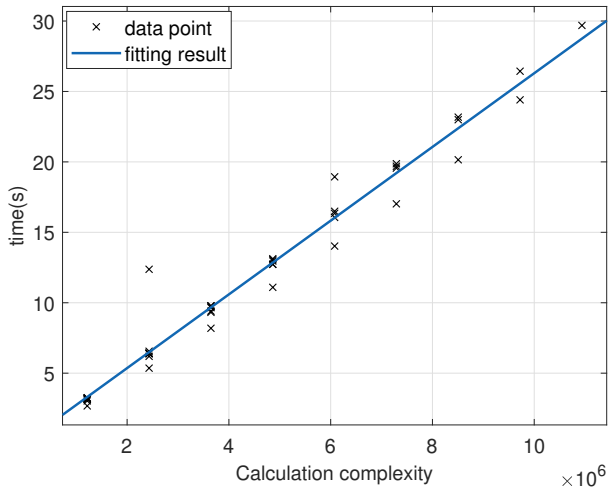
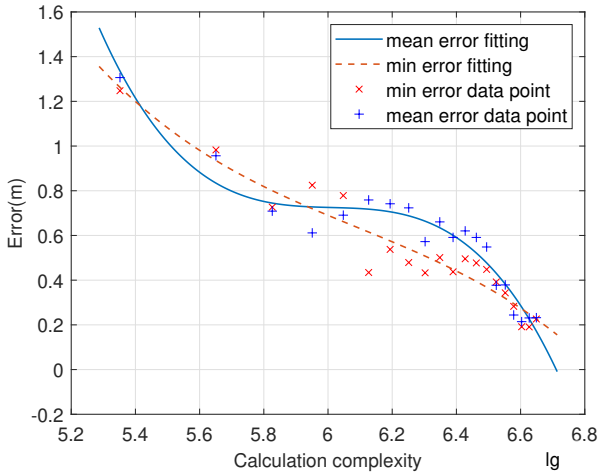


Fig. 10. Fitting result of the relationship between execution time and calculation complexity.

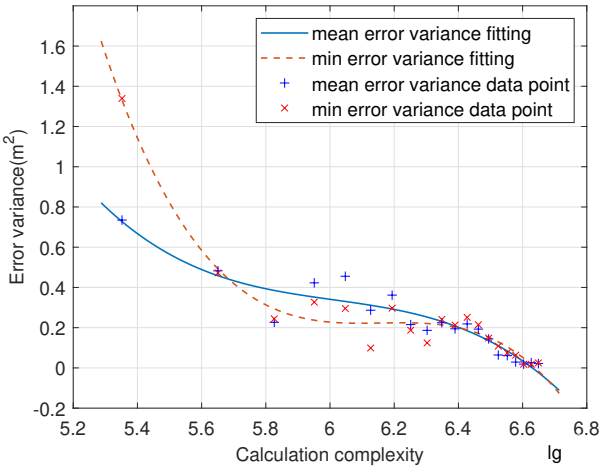
C. Distance Estimation Error and Data Abundance

In addition to noise, the abundance of data amount for each body shielding category at each distance also affects the accuracy of the estimated distance. This relationship is shown in Fig. 11 through the fitting curve of the errors and calculation complexity, as higher data abundance can result in higher calculation complexity. Since the calculation complexity is related to the number of the distance ranges where the input data is collected and the number of body shielding categories included at each distance, it is possible that the total data amount of the two sets of data is the same, but the data abundance is different. For example, one group of input data contains the data collected at three distances, each distance includes three shielding categories, and the amount of data for each shielding category is 60. Meanwhile, the other group of input data contains the data collected at two distances, each distance includes three shielding categories, and the amount of data for each shielding category is 90. These two sets have the same total amount of data, but the data abundance is different i.e., the first group has 60 data sets for each shielding category at each distance, while the other group has 90 data sets for each shielding category at each distance.

In order to characterize the influence of data abundance on errors, we consider all possible distance ranges and body shielding categories. We assume that all body shielding categories are included at each distance, and the data amount for each shielding category at each distance is fixed. Firstly, we fix the lower limit of the distance range at 0 m and decrease the upper limit from 4.5 m at an interval of 0.5 m. When the upper limit is changed, we calculate the error (the difference between the estimated distance and the ground truth distance) and calculation complexity. Therefore, for one lower limit, there are several errors and complexities corresponding to different upper limits. Then, we increase the lower limit by 0.5 m and repeat the same process until the lower limit gets to the largest distance i.e., 4.5 m, so that there are several errors and complexities for a fixed data amount. Finally, we increase the



(a) Fitting for error with number of calculation.



(b) Fitting for error variance with calculation complexity.

Fig. 11. Fitting for error and error variance with calculation complexity.

data amount and repeat the same process mentioned above to calculate the average error, error variance, and average calculation complexity for each data amount. The average error represents the accuracy of the algorithm, and the error variance shows the stability of the algorithm for different distance ranges and shielding categories. The relationship between the average error and the average calculation complexity is shown in Fig. 11a. Meanwhile, the variance of the error is fitted with the average calculation complexity to show the stability of the algorithm depicted in Fig. 11b. As the data abundance gradually increases (i.e., the amount of data for each distance and shielding category increases from 50 to 500 with the interval of 50), the estimated distance error and error variance gradually decrease. It indicates that the accuracy and stability of the estimated distance converge with the data abundance.

D. Quadrant of Latency, Accuracy and Stability

Since the running time is positively correlated with the calculation complexity, and the data abundance affecting the accuracy of the algorithm is also related to the calculation

complexity, the user might need to weigh the estimated distance accuracy of the algorithm, the stability and delay of the estimated distance to figure out a balance in actual use. Therefore, the estimated mean and minimum distance error (accuracy), as well as distance error variance (stability) and calculation complexity (latency) are divided into three levels: low, middle and high (labeled as L, M and H) as shown in Fig. 12. The three levels corresponding to the three parameters respectively represent low, middle, high accuracy, low, middle, high stability, and low, middle and high latency.

When the amount of data for each shielding category at each distance is increased gradually, the maximum and minimum errors and the error variance are obtained. The boundary values of the three levels are calculated by adding the minimum error, minimum error variance and minimum calculation complexity by one-third of the difference between the maximum and minimum values. Since these boundary values are not fixed, users can define the boundaries of these levels by themselves when they use the algorithm. Here we just show Fig. 12 as a reference.

E. Data Expansion for Lower Error

Since the error of the distance estimated by the algorithm is negatively correlated with the data abundance, a higher data abundance is required to obtain a higher precision of the estimated distance. However, in real environments, the time for two contacts to meet might be short, which makes it possible to obtain an unsatisfactory estimated distance by directly using the machine learning algorithm. Due to the robustness of the algorithm for truncating Gaussian noise, when the amount of data is insufficient, adding Gaussian noise to the collected data to generate new data can expand the amount of data and hence to improve the accuracy of the estimated distance. In our experiments, the original data volume is 25, 100, 200 and 300 for each shielding category and distance in the testing set. When new data is generated by adding truncating standard normal noise in the range of $[-3, 3]$, the estimated average distance and minimum distance, average distance variance and minimum distance variance tend to converge as shown in Fig. 13. In actual use, if the data abundance is low, the data can be expanded by adding noise until the average distance and minimum distance estimated by the algorithm no longer change significantly.

VII. Challenges and Countermeasures

A. Battery Power Consumption and Time Cost

When using our proposed DISTERNING algorithm, the mobile phones BLE needs to be set in the state of scanning and broadcasting. Although there is no need to establish a link and the BLE greatly optimizes its power consumption, it still drains the battery. However, the power consumption of the hardware can be reduced by changing the broadcasting and scanning frequency in different situations (e.g., reducing the frequency of scanning and broadcasting when the surrounding devices are not scanned). When the machine learning algorithm is performed on a mobile phone, the calculation process also reduces the battery life. Therefore, distributing data collection and data processing to mobile phone and edge respectively can effectively improve the battery life of mobile phones. The calculation can be done elsewhere such as mobile edge, while the mobile phone is only utilized to collect RSSI data. When an exposure notation is received, the mobile phone uploads the collected data to the edge for distance estimation.

In addition, the machine learning algorithm takes time to be run on the mobile phone. When the data is large, the calculation might take a long time. Therefore, for contact tracing that has some constraints on time, it is recommended to apply the division strategy to data collection and data processing. This can be solved by edge/cloud deployment (see Section III).

B. Security and Privacy

Another major challenge is the information transmission and information storage between different organizations in DISTERNING deployment scheme. The information (including Bluetooth signal information and estimated distance information) and its transmission might introduce some privacy issues. Although encrypting the users' ID and distorting transmitted data by adding noise are used in this paper to solve this problem, further research is needed for the data protection between users and edge devices. Additionally, the noise added introduces some errors to the algorithm thus a trade-off between accuracy and privacy security is required. The research also serves as a foundation for distributed DCT solutions such as BeepTrace [41][42][43], which provides a highly secured and privacy-preserving solution through blockchain technology for data sharing.

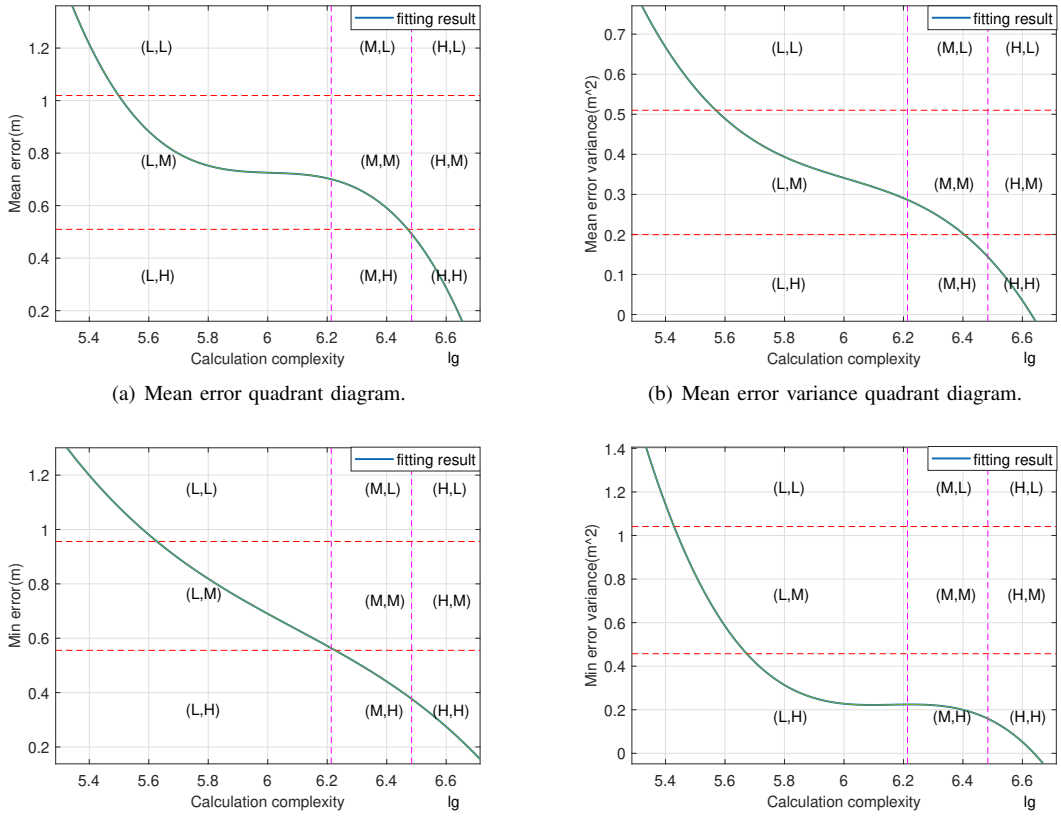
C. Data Storage Capacity

When the strategy of separating the data collection terminal from the data storage terminal is adopted, as the number of users increases, more data needs to be stored temporarily on the edge. Moreover, COVID-19 has a certain incubation period, so the Bluetooth signal data collected by mobile phones of contacts should be stored in the mobile phone for a period of time. During this period, the mobile phone might continue receiving new data, which requires the phone to have enough reserved storage space for new data. If both data collection and data processing are completed by mobile phone, except for Bluetooth signal information, the mobile phone also needs to store the estimated distance and algorithm. This might occupy a large amount of storage space on the mobile phone.

Therefore, as mentioned in Section III, adopting the strategy of separating data collection and processing would significantly reduce the storage usage of mobile phones. However, a more efficient data storage strategy is required for both edge and mobile phones to temporarily store Bluetooth signal data.

D. Deployment Scope of BLE

As a new generation of Bluetooth technology, BLE has been widely deployed on mobile phones by major manufacturers in the sector of mobile communication, but a small number of old phones might not support BLE. This problem might trouble



(a) Mean error quadrant diagram.

(b) Mean error variance quadrant diagram.

(c) Minimum error quadrant diagram.

(d) Minimum error variance quadrant diagram.

Fig. 12. Quadrant diagram for mean error, mean error variance, minimum error, minimum error variance.

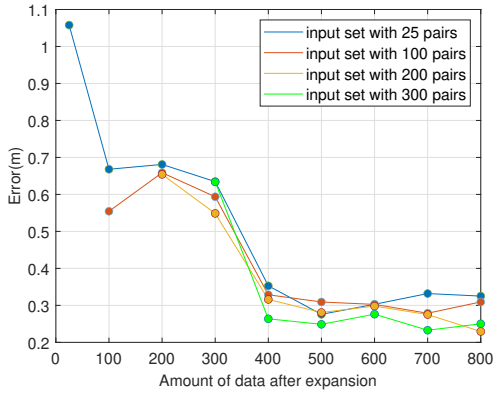
the elderly and minors since they might be less concerned about the update of mobile phones.

However, our solution can not only use the BLE on the mobile phone for data collection but also fit other more general communication modules with broadcasting and scanning functions on different mobile phones. Contact tracing based on these communication modules (e.g., WiFi) can be trained using the same architecture of the machine learning algorithm proposed in DISTERNING to generate corresponding applications, which provides a solution for the scope challenge. In addition, other hardware or wearable devices with short-distance communication functions can also be developed as front-end data collection devices.

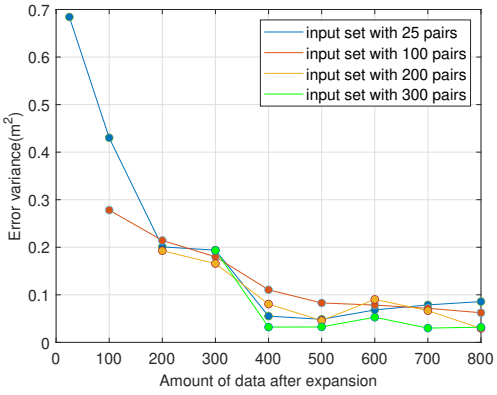
VIII. Conclusion

In this paper, DISTERNING - a contact distance estimation algorithm based on machine learning and Bluetooth low energy (BLE) is proposed to solve the infection risk assessment problem of COVID-19 digital contact tracing, and a scheme based on edge computing and cloud training is designed for the deployment of the algorithm. By training the distance estimation model in the cloud and deploying on edge, the scheme can provide stronger computational power than mobile device deployment and lower transmitting latency than cloud deployment. The edge server only computes the estimated distance using encrypted and distorted data transmitted from users, and there is no user data storage on the edge so that privacy security is improved. Our DISTERNING algorithm can extract distance features from RSSI data to calculate the contacting distance value between two contacts. Meanwhile, our algorithm considers the shielding on RSSI caused by the human body when the mobile phone is placed in different positions of the human body in the real situation. Furthermore, other potential negative environmental factors in reality are simulated by adding noise to the original data. After a series of machine learning algorithms and data processing methods such as filtering, the algorithm finally outputs the estimated average distance and minimum distance between contacts. Then the two values are transmitted together with the contact time recorded by the mobile phone to the hospital or other medical professional institutions for contact infection risk assessment. The experimental result of the estimated distance shows that when the input data set contains shielding and noise, the error of the estimated distance for most cases could still be kept within 1 m. In the case of a small amount of input data, new data can be generated by adding noise to improve the accuracy of the estimated distance.

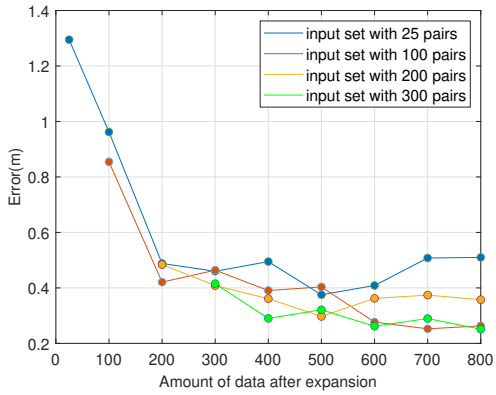
The proposed algorithm provides a low-cost and portable approach to distance estimation between contacts using Bluetooth. Since the hardware to underpin our solution is universal smartphones, and the Bluetooth RSSI is easy to be obtained, the applied scope of our proposed algorithm is not only



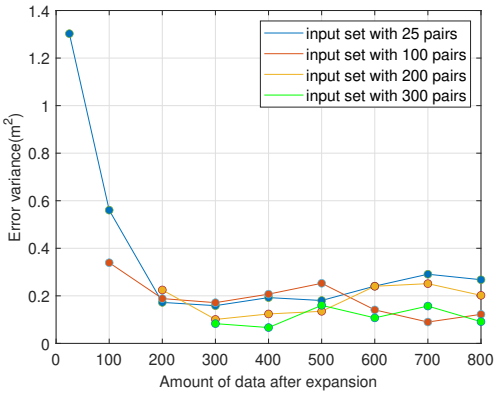
(a) Input RSSI data expansion for mean error.



(b) Input RSSI data expansion for mean error variance.



(c) Input RSSI data expansion for minimum error.



(d) Input RSSI data expansion for minimum error variance.

Fig. 13. Input RSSI data expansion for mean, minimum error and mean, minimum error variance.

the contact tracing of COVID-19 but also other infectious diseases to serve for the infection risk assessment of contacts. Furthermore, for other distance estimation tasks such as labor intensity estimation and population density evaluation, the proposed algorithm DISTERNING can also be considered as a cost-effective tool.

The main challenges to the deployment of the algorithm are the power consumption of the hardware BLE it relies on, the encryption and data distortion method for privacy protection in data transmission and storage, the data storage capacity, and the deployment scope of BLE. In future work, we plan to solve the problems we face and further optimize our current design. The DISTERNING algorithm might be developed into a new application or combined with other existing solutions of contact tracing for COVID-19 and beyond.

REFERENCES

[1] COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU). [Online]. Available: https://coronavirus.jhu.edu/map.html

[2] J. Chen, A. Vullikanti, J. Santos, S. Venkatramanan, S. Hoops, H. Mortveit, B. Lewis, W. You, S. Eubank, M. Marathe *et al.*, "Epidemiological and economic impact of covid-19 in the us," *medRxiv*, 2020.

[3] Centers for Disease Control and Prevention (CDC) Nonpharmaceutical Interventions. [Online]. Available: https://www.cdc.gov/nonpharmaceutical-interventions/workplace/index.html

[4] J. Yang, V. Marziano, X. Deng, G. Guzzetta, J. Zhang, F. Trentini, J. Cai, P. Poletti, W. Zheng, W. Wang *et al.*, "Despite vaccination, china needs non-pharmaceutical interventions to prevent widespread outbreaks of covid-19 in 2021," *Nature Human Behaviour*, pp. 1–12, 2021.

[5] M. McKee and D. Stuckler, "If the world fails to protect the economy, covid-19 will damage health not just now but also in the future," *Nature Medicine*, vol. 26, no. 5, pp. 640–642, 2020.

[6] M. Qian and J. Jiang, "Covid-19 and social distancing," *Journal of Public Health*, pp. 1–3, 2020.

[7] W. J. Bradshaw, E. C. Alley, J. H. Huggins, A. L. Lloyd, and K. M. Esvelt, "Bidirectional contact tracing could dramatically improve covid-19 control," *Nature communications*, vol. 12, no. 1, pp. 1–9, 2021.

[8] P. Rodríguez, S. Graña, E. E. Alvarez-León, M. Battaglini, F. J. Darias, M. A. Hernán, R. López, P. Llaneza, M. C. Martín, O. Ramirez-Rubio *et al.*, "A population-based controlled experiment assessing the epidemiological impact of digital contact tracing," *Nature communications*, vol. 12, no. 1, pp. 1–6, 2021.

[9] H. Šiljak, N. Ashraf, M. T. Barros, D. P. Martins, B. Butler, A. Farhang, N. Marchetti, and S. Balasubramaniam, "Evolving intelligent reflector surface toward 6g for public health: Application in airborne virus detection," *IEEE Network*, vol. 35, no. 5, pp. 306–312, 2021.

[10] CHAPTER V:THE EPIDEMIOLOGIC INVESTIGATION in Foodborne Illness Investigation and Control Manual. [Online]. Available: https://www.mass.gov/doc/the-epidemiologic-investigation/download

[11] J. Bay, J. Kek, A. Tan, C. S. Hau, L. Yongquan, J. Tan, and T. A. Quy, "Bluetrace: A privacy-preserving protocol for community-driven contact tracing across borders," *Government Technology Agency-Singapore, Tech. Rep*, 2020.

[12] S. Bluetooth, "Sig introduces bluetooth low energy wireless technology, the next generation of bluetooth wireless technology," *press release, Dec*, vol. 17, 2009.

[13] T. Sharon, "Blind-sided by privacy? digital contact tracing, the apple/google api and big techs newfound role as global health policy

makers," *Ethics and Information Technology*, vol. 23, no. 1, pp. 45–57, 2021.

[14] W. J. Buchanan, M. A. Imran, M. Ur-Rehman, L. Zhang, Q. H. Abbasi, C. Chrysoulas, D. Haynes, N. Pitropakis, and P. Papadopoulos, "Review and critical analysis of privacy-preserving infection tracking and contact tracing," *Frontiers in Communications and Networks*, vol. 1, p. 2, 2020.

[15] M. Cebrian, "The past, present and future of digital contact tracing," *Nature Electronics*, vol. 4, no. 1, pp. 2–4, 2021.

[16] E. D. Kaplan and C. Hegarty, *Understanding GPS/GNSS: principles and applications*. Artech house, 2017.

[17] Z. Su, K. Pahlavan, and E. Agu, "Performance evaluation of covid-19 proximity detection using bluetooth le signal," *IEEE Access*, vol. 9, pp. 38 891–38 906, 2021.

[18] G. Cortellessa, L. Stabile, F. Arpino, D. Faleiros, W. Van Den Bos, L. Morawska, and G. Buonanno, "Close proximity risk assessment for sars-cov-2 infection," *Science of The Total Environment*, vol. 794, p. 148749, 2021.

[19] P. V. Krishna, S. Misra, V. Saritha, D. N. Raju, and M. S. Obaidat, "An efficient learning automata based task offloading in mobile cloud computing environments," in *2017 IEEE international conference on communications (ICC)*. IEEE, 2017, pp. 1–6.

[20] N. Pathak, P. K. Deb, A. Mukherjee, and S. Misra, "Iot-to-the-rescue: A survey of iot solutions for covid-19-like pandemics," *IEEE Internet of Things Journal*, 2021.

[21] M. Almalki and A. Giannicchi, "Health apps for combating covid-19: Descriptive review and taxonomy," *JMIR mHealth and uHealth*, vol. 9, no. 3, p. e24322, 2021.

[22] B. Etzlinger, B. Nußbaummüller, P. Peterseil, and K. A. Hummel, "Distance estimation for ble-based contact tracing–a measurement study," in *2021 Wireless Days (WD)*. IEEE, 2021, pp. 1–5.

[23] D. J. Leith and S. Farrell, "Coronavirus contact tracing: Evaluating the potential of using bluetooth received signal strength for proximity detection," *ACM SIGCOMM Computer Communication Review*, vol. 50, no. 4, pp. 66–74, 2020.

[24] S. Liu, Y. Jiang, and A. Striegel, "Face-to-face proximity estimationusing bluetooth on smartphones," *IEEE Transactions on Mobile Computing*, vol. 13, no. 4, pp. 811–823, 2013.

[25] F. Sattler, J. Ma, P. Wagner, D. Neumann, M. Wenzel, R. Schäfer, W. Samek, K.-R. Müller, and T. Wiegand, "Risk estimation of sars-cov-2 transmission from bluetooth low energy measurements," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–4, 2020.

[26] C. H. Lam, P. C. Ng, and J. She, "Improved distance estimation with ble beacon using kalman filter and svm," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[27] S. Parikh, D. Dave, R. Patel, and N. Doshi, "Security and privacy issues in cloud, fog and edge computing," *Procedia Computer Science*, vol. 160, pp. 734–739, 2019, the 10th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2019) / The 9th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2019) / Affiliated Workshops. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919317181

[28] X. Deng, Z. Sun, D. Li, J. Luo, and S. Wan, "User-centric computation offloading for edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 559–12 568, 2021.

[29] H. Xu, P. V. Klaine, O. Onireti, B. Cao, M. Imran, and L. Zhang, "Blockchain-enabled resource management and sharing for 6G communications," *Digital Communications and Networks*, vol. 6, no. 3, pp. 261–269, aug 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2352864820300249

[30] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal *et al.*, "Mobile-edge computing introductory technical white paper," *White paper, mobile-edge computing (MEC) industry initiative*, vol. 29, pp. 854–864, 2014.

[31] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[32] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE communications surveys & tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[33] A. Arvanitopoulos, J. Gialelis, and S. Koubias, "Energy efficient indoor localization utilizing bt 4.0 strapdown inertial navigation system," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE, 2014, pp. 1–5.

[34] A. Patri and S. P. Rath, "Elimination of gaussian noise using entropy function for a rssi based localization," in *2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*. IEEE, 2013, pp. 690–694.

[35] K. Pahlavan and A. Levesque, "Modeling and simulation of narrowband signal characteristics," *Wireless Information Networks*, pp. 94–122, 2005.

[36] N. R. Jones, Z. U. Qureshi, R. J. Temple, J. P. Larwood, T. Greenhalgh, and L. Bourouiba, "Two metres or one: what is the evidence for physical distancing in covid-19?" *bmj*, vol. 370, 2020.

[37] Timestamp. [Online]. Available: https://developer.android.google.cn/reference/kotlin/java/sql/Timestamp?hl=en

[38] Prevention for Coronavirus Disease (Covid-19) by World Health Organization (WHO). [Online]. Available: https://www.who.int/health-topics/coronavirus$#$tab=tab$_$2

[39] S. G. Devi, K. Selvam, and S. Rajagopalan, "An abstract to calculate big o factors of time and space complexity of machine code," 2011.

[40] K. A. Frenkel, "Evaluating two massively parallel machines," *Communications of the ACM*, vol. 29, no. 8, pp. 752–758, 1986.

[41] H. Xu, L. Zhang, O. Onireti, Y. Fang, W. J. Buchanan, and M. A. Imran, "Beeptrace: Blockchain-enabled privacy-preserving contact tracing for covid-19 pandemic and beyond," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3915–3929, 2020.

[42] P. V. Klaine, L. Zhang, B. Zhou, Y. Sun, H. Xu, and M. Imran, "Privacy-preserving contact tracing and public risk assessment using blockchain for covid-19 pandemic," *IEEE Internet of Things Magazine*, vol. 3, no. 3, pp. 58–63, 2020.

[43] H. Kang, Z. Zhang, J. Dong, Y. Ji, H. Xu, and L. Zhang, "Beeptrace for covid-19 pandemic: A demo," in *2021 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. IEEE, 2021, pp. 1–2.