Hybrid Learning for Mobile Ad-Hoc Distancing/Positioning Using Bluetooth Low Energy

Yik Him Ho, Yunfei Liu, Caiqi Zhang, Yerkezhan Sartayeva, and Henry C. B. Chan, *Member, IEEE*

Abstract—With the advent of Bluetooth low-energy (BLE)-enabled smartphones, there has been considerable interest in investigating BLE-based distancing/positioning methods (e.g., for social distancing applications). In this article, we present a novel hybrid learning method to support mobile ad-hoc distancing (MAD)/positioning (MAP) using BLE-enabled smartphones. Compared to traditional BLE-based distancing/positioning methods, the hybrid learning method provides the following unique features and contributions. First, it combines unsupervised learning, supervised learning, and genetic algorithms (GAs) for enhancing distance estimation accuracy. Second, unsupervised learning is employed to identify three pseudo channels/clusters for enhanced RSSI data processing. Third, its underlying mechanism is based on a new pattern-inspired approach to enhance the machine learning process. Fourth, it provides a flagging mechanism to alert users if a predicted distance is accurate or not. Fifth, it provides a model aggregation scheme with an innovative 2-D GA to aggregate the distance estimation results of different machine learning models. As an application of hybrid learning for distance estimation, we also present a new MAP scenario with an iterative algorithm to estimate mobile positions in an ad-hoc environment. Experimental results show the effectiveness of the hybrid learning method. In particular, hybrid learning without flagging and with flagging outperforms the baseline by 57% and 65%, respectively, in terms of mean absolute error. By means of model aggregation, a further 4% improvement can be realized. The hybrid learning approach can also be applied to previous work to enhance distance estimation accuracy and provide valuable insights for further research.

Index Terms—Bluetooth low energy (BLE), COVID-19, machine learning, mobile ad-hoc distancing (MAD)/positioning (MAP), social distancing.

I. Introduction

ACCORDING to the statistics in [1], there are more than five billion smartphones worldwide with an annual growth rate of approximately 10% in recent years. Most of today's smartphones are equipped with Bluetooth low
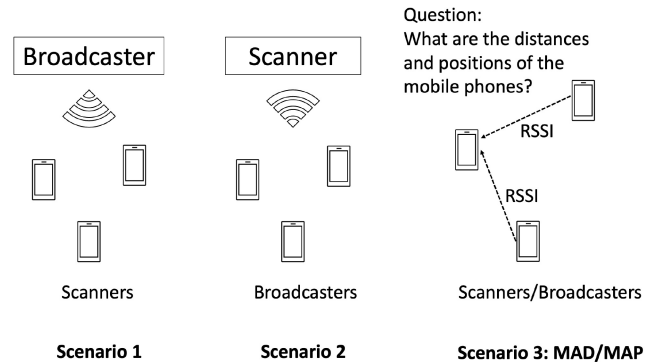
Fig. 1. Three possible BLE-based positioning scenarios.

energy (BLE) [2]. Recently, due to COVID-19, there has been considerable interest in investigating and developing social distancing-related applications (e.g., to monitor distance between people) [3], [4]. BLE-enabled smartphones are well suited to support these applications, and BLE is the most popular technology of choice for contact tracing applications [5]. The aim of this article is to contribute to this important current development with a new distancing/positioning paradigm called mobile ad-hoc distancing (MAD)/mobile ad-hoc positioning (MAP) using machine learning.

Similar to WiFi, BLE also operates over the ISM (2.4 GHz) band [6]. There are forty channels, and three of them (channels 37, 38, and 39) are advertisement channels [6]. There are two types/modes of BLE devices: 1) broadcaster and 2) scanner. During the advertisement phase, a broadcaster sends advertisement packets periodically over the three channels. A scanner can receive the packets and hence detect the radio signal strength index (RSSI), which is one of the most popular types of data for estimating the distance between two BLE devices [7].

Fig. 1 shows the three possible BLE-based positioning scenarios. In the first case, the broadcasters are fixed and the scanners are mobile. This is the common case in use. It is also possible to have the broadcasters moving and the scanners fixed (i.e., second case). For example, people can wear BLE tags which are detected by scanners. In the third case, both the broadcasters and scanners can be mobile (e.g., people with BLE phones monitor the distances between each other). We call this MAD/MAP scenario. Relatively less work has been conducted for this new scenario, which is closely related to

TABLE I
COMPARISON BETWEEN THE TRADITIONAL APPROACH AND NEW APPROACH

Traditional approach	New approach
Supervised learning	Hybrid learning with unsupervised learning, supervised learning and genetic algorithms (GA)
Aggregated channel	Three pseudo channels based on clustering
More data/statistic oriented	Data/statistic and pattern oriented
No flagging	With flagging
Single model	Model aggregation with two-dimensional GA

social distancing applications. In fact, there are new research challenges and opportunities. For example, researchers have been investigating the use of fingerprinting for BLE-based positioning, e.g., [2]. However, this cannot be used for the MAD/MAP scenario because of the ad-hoc nature (i.e., fingerprints cannot be obtained when both broadcasters and scanners are mobile in an ad-hoc environment). On the other hand, the use of smartphones (i.e., for both broadcasters and scanners) allows more complex methods to be employed and machine learning-based methods should play an important role in the MAD/MAP scenario. As shown in Fig. 1, the basic question is based on the RSSIs, what are the distances and positions of the mobile phones?

To support MAD/MAP, there are five important research questions. First, what are the major RSSI-related issues that hinder the distance estimation accuracy? In other words, the core question is that based on available RSSI data, how can we enhance the distance estimation accuracy? Second, is there any new approach for supporting this new MAD/MAP scenario? Third, how can we predict whether a distance estimate is accurate or not? Fourth, how can we combine the results of different machine learning models to enhance the distance estimation accuracy? Fifth, how can we tackle the new positioning problem in an ad-hoc environment (i.e., without infrastructure, such as the conventional trilateration problem with, for example, three fixed beacons)? As shown in Table I, this article aims to address these research questions using a novel approach with the following new contributions.

1) We conduct an RSSI analysis. One major finding is that RSSI varies over discrete levels. In nearly all previous work, an aggregate channel is employed for training purposes as channel-related information is usually not available through the mobile phone application programming interface (API) (i.e., it is assumed that RSSI varies over one channel [8]). This is in fact one major cause of inaccuracy. As will be explained later, to tackle this issue, we employ an innovative unsupervised learning (clustering) method to cluster RSSI over three pseudo channels so as to enhance the distance estimation accuracy.

2) We present a new hybrid learning method to support MAD/MAP. Traditionally, supervised learning is used for training RSSI-based data for distance/positioning estimation purposes. In the new approach, we investigate hybrid learning by combining unsupervised learning (i.e., for clustering RSSI data), supervised learning, and genetic algorithms (GAs) to enhance distance estimation accuracy based on RSSI data. As shown by later experimental results, the hybrid learning method can achieve significant improvement over traditional methods.

3) In addition to using data analysis and statistical methods, the hybrid learning approach also makes use of a pattern-inspired mechanism (i.e., inspired by image recognition). With traditional methods, it is not easy to determine whether an estimated distance is good. By using the pattern-inspired mechanism, the estimated distance can be flagged more effectively (i.e., to predict whether a prediction is good or bad). This allows further accuracy improvements.

4) Unlike conventional methods in which a single machine learning model is used, we propose a new model aggregation mechanism based on an innovative 2-D GA method so that prediction results from different machine learning models can be combined to achieve better accuracy.

5) We discuss the MAP scenario with a new iterative algorithm to estimate the positions of mobile users in a mobile ad-hoc environment. This is a new problem with little prior study.

6) We present extensive experimental results to study the proposed hybrid learning approach, as compared to traditional methods. Experimental results show that by using hybrid learning, the mean absolute error (MAE) can be reduced to 0.66 m. Compared to the baseline method and basic machine learning method, the improvement percentages are 57% and 25%, respectively. With the flagging option for hybrid learning and the model aggregation method, further improvements can be achieved at the expense of excluding some estimated distances.

To the best of our knowledge, the hybrid learning approach with unsupervised learning, supervised learning, and GA for supporting MAD/MAP has not been studied in the literature. This new approach can also be applied to previous work to improve performance and provide new insights for future research.

The remaining sections of this article are organized as follows. Section II presents related work. Section III presents the RSSI analysis and the hybrid learning method to support MAD/MAP. Section IV presents and discusses the experimental results. Section V gives the conclusion and future work.

II. RELATED WORK

A. BLE Positioning Methodologies

Most BLE positioning techniques rely on radio signal strength indicator (RSSI) values, but RSSI is an unreliable indicator of distance because it is sensitive to obstructions

TABLE II
SUMMARY OF RELATED WORK

Work	Positioning Approach	Data Collection & Processing	Distance and Position Estimation
[9]	Proximity	RSSI	Offline Linear Least-Squares (LLS) Regression, Offline Nonlinear Gaussian Process Regression, Online Gaussian Process Regression, Fundamental lower bounds proximity report-based position estimator
[10]	Proximity and device motion	RSSI and device motion	Log-Distance Path Loss model, Motion model and particle smoothing algorithm
[11]	Proximity	RSSI and signal inter-arrival time	Log-Distance Path Loss model and Adaptive scanning heuristic algorithm with differential evolution
[12]	Signal propagation	RSSI	Log-Distance Path Loss model for distance estimation, Trilateration for positioning
[13]	Signal propagation	RSSI, Gaussian filter, weighted sliding windows	Log-Distance Path Loss model for distance estimation, Taylor series expansion based collaborative positioning, Triangle trilateral relations theorem for position correction
[14]	Signal propagation	RSSI, Kalman filter, moving average	Log-Distance Path Loss model for distance estimation, Weighted centroid localization algorithm for positioning
[15]	Signal propagation	RSSI, smoothing filter, wavelet filter	Log-Distance Path Loss model for distance estimation, Weighted off-set triangulation algorithm for positioning
[16]	Signal propagation	RSSI, broadcasting channel information, Kalman filter	Log-Distance Path Loss model for distance estimation, International Telecommunication Union (ITU) model, empirical model, Weighted trilateration for positioning
[17]	Signal propagation	RSSI	Log-Distance Path Loss for distance estimation, Inter Ring Localization Algorithm (iRingLA) for positioning
[18]	Signal propagation and dead reckoning	RSSI, average, median, Kalman filter, device sensors readings	Log-Distance Path Loss model for distance estimation, Trilateration for positioning
[19]	Fingerprint	RSSI, maximum, median, mean	Bayesian Likelihood estimator
[20]	Fingerprint	RSSI, channel and orientation fingerprinting	Weighted kNN
[21]	Fingerprint	RSSI	Autoencoder, kNN
[2]	Fingerprint	RSSI mean and standard deviation	kNN and weighted kNN with Chebyshev and Euclidean metrics.
[22]	Fingerprint	RSSI	Eight-Neighborhood Template Matching (ENTM)

and is thus subject to noise [23]. This is referred to as the multipath problem [24]. Estimating distances/positions based on RSSI is challenging, and most BLE-positioning methods seek to tackle this problem by different means.

As shown in Table II, there are three major BLE positioning approaches, namely, proximity-based, signal propagation-based with trilateration, and fingerprint-based. They are primarily based on RSSI. The most basic approach is proximity-based, which is designed to provide a proximity-based service in particular. For example, Apple's iBeacon provides a proximity-based mechanism, indicating whether a beacon is close to or far from a mobile terminal (e.g., based on an RSSI threshold) [25]. To enhance the proximity-based method, researchers have studied more advanced techniques. For example, [9] studied a threshold optimization framework. Three RSSI-related models have been studied: 1) offline linear least-squares regression; 2) offline nonlinear Gaussian process regression (GPR); and 3) online GPR. The optimal threshold is determined utilizing an optimization problem. Zhao et al. [10] investigated a positioning method by considering the proximity reports sent by a mobile terminal over time and using a motion model. Particle filtering and smoothing methods are used for position determination. Ng et al. [11] investigated a proximity-based method for dense deployment of beacons, as the conventional method may not work effectively

when a large number of beacons is close to one another. It employs an adaptive scanning mechanism together with a spontaneous differential evolution algorithm to provide high detection accuracy [11].

Trilateration/multilateration is one of the simplest RSSI-based positioning methods. RSSI values from multiple anchors are collected, and the distance to each anchor is estimated using the signal attenuation model [26]. These distance values are then used to estimate the position of an object, typically based on a least-squares approach. As the RSSI values are often noisy with large variations, filtering methods can be employed to address this issue. For example, [13] developed a positioning system with a weighted sliding window and Gaussian filtering to smoothen the RSSI curve. Furthermore, they used a Taylor series expansion-based algorithm to further enhance positioning accuracy.

Another popular BLE positioning method is RSSI fingerprinting. An RSSI fingerprint is a vector of RSSI values of multiple BLE beacons collected at a single reference point. To find the closest fingerprint in a database, the k-nearest Neighbor (kNN) algorithm is commonly used. For example, [2] used both kNN and weighted kNN. Another study used kNN [27] for a smart domestic power management system, where positioning was needed to calculate energy consumption. To address the issue of high variance in raw RSSI values,

they filtered the stream of values first and then extracted maximum and minimum values from each reference point. Yadav et al. [28] extended kNN with Bayesian estimation into a single algorithm called the trusted *k*-nearest Bayesian estimation (TKBE). They coupled TKBE with dead reckoning to reduce positioning errors, and incorporated Kalman filtering based on fuzzy logic to further enhance performance. Apart from kNN, there are other schemes for matching fingerprints in a database. For example, [29] proposed a novel ranking-based RSSI fingerprinting method with Kendall Tau correlation coefficient (KTCC) for fingerprint matching. One of the main disadvantages of fingerprint-based positioning is that an extensive database of fingerprints must be compiled in advance before real-time positioning can take place. Ho and Chan [30] designed a decentralized indoor positioning protocol to build an RSSI database on the fly by allowing broadcasters to exchange signals with each other. This database can then be used to estimate the distance from an object to anchors.

Some studies have been conducted to investigate the use of the newly released Angle of Arrival (AoA) feature of the Bluetooth 5.1 standard for indoor positioning [31] using multiangulation and have demonstrated promising results. For example, [32] experimented with using AoA alone and combining multiangulation with multilateration, and found that both methods performed slightly better than current methods. Another interesting study [33] has been done on the use of the AoA technology for 3-D positioning, where the authors used the least-squares approach for position estimation.

According to [34], BLE-positioning methods can be combined with other methods to enhance accuracy and performance. Among these complementary methods are 1) sensor-based positioning using sensor data from gyroscopes, magnetometers, and accelerometers, available in most smartphones; 2) magnetic field fingerprinting matching known magnetic field distributions with readings for unknown locations; and 3) map-matching (an indoor map is taken into consideration during positioning). For example, [35] combined dead reckoning with trilateration and Kalman filtering, thus proposing a hybrid positioning method with better performance. Röbesaat et al. [18] used a similar approach, achieving promising performance. Another study in [36] developed different signal attenuation models for three BLE advertising channels instead of relying on a single channel for distance estimation. Measurements from the three channels are then assembled to calculate a more stable distance value. Ye et al. [37] designed a positioning system to remove the need for multilateration and multitriangulation as only one anchor is used for positioning. Their system combines data from the anchor and a pedestrian dead reckoning (PDR) method, and further improves positioning performance using Kalman filtering.

B. Indoor Positioning Using Machine Learning

As the trajectory of BLE signal propagation is difficult to model, there has been growing interest in exploring machine learning to support positioning. Li et al. [38] provided a survey of the use of machine learning for indoor positioning. The study in [39] reiterated the advantages of using machine learning in positioning methods. Bozkurt et al. [40] compared multiple machine learning models for fingerprint-based BLE indoor positioning and established that kNN outperformed other models such as decision trees. They found that even after using ensemble learning models, such as AdaBoost and Bagging, decision trees still performed more poorly. Prasad et al. [41] used GPR in their RSS fingerprint-based positioning system. They used two types of GPR: 1) conventional GPR (CGPR) and 2) numerical approximation GPR (NaGPR). Their proof-of-concept solution involves training a machine learning model in a noise-free indoor environment, to be deployed later in real-world conditions, and their experimental analysis demonstrated that this approach improved the root-mean-square error (RMSE) of NaGPR. Homayounvala et al. [42] compared GPR to other ML methods like support vector machine (SVM) and kNN and found that the former outperformed the latter in positioning accuracy. Another common machine learning model used for real-time indoor positioning is the extreme learning machine (ELM), which features a single hidden layer. Lu et al. [43] compared multiple types of ELMs in real-world and simulated conditions and achieved promising results. Zhao and Wang [44] also used SVM, but to mitigate the noise issue, they used kernel direct discriminant analysis (KDDA) and relevance vector regression (RVR). Their proposed method outperformed existing methods, such as weighted kNN. Tran and Ha [45] investigated a creative machine learning approach by using both classification and regression models for a light-based positioning system. In essence, classification was used to divide the indoor environment into two distinct areas, and then regression was used for position estimation. Last but not least, there has been growing interest in studying deep learning in indoor positioning research. For example, [46] built a fingerprint-based positioning framework called DeepFi, where channel state information (CSI) is used. In the offline phase, fingerprints are fed into a deep learning net, which is then used in the online phase for real-time positioning. Another study [21] used an autoencoder net to capture high-level features and solve the problem of high feature dimensionality. Their system can use new data to learn on the fly, making it a flexible positioning solution. Zou et al. [47] also trained an online deep learning network that continuously accepts new training data at a rapid speed. They built a sequential feedforward net with one hidden layer and used WiFi RSS fingerprints for the input layer. BelMannoubi and Touati [48] also utilized autoencoders, aiming to address the multipath problem. Their experimental results showed their model performed better than kNN and SVM. Long short-term memory (LSTM) nets have also found a use in indoor positioning [49], with one study employing a hybrid network with LSTM elements [39]. While convolutional neural networks (CNNs) are commonly used for image processing, there were also attempts to use them for indoor positioning [50].

III. HYBRID LEARNING METHOD

In this section, we first present an analysis of RSSI, and then the hybrid learning method. In particular, the hybrid learning
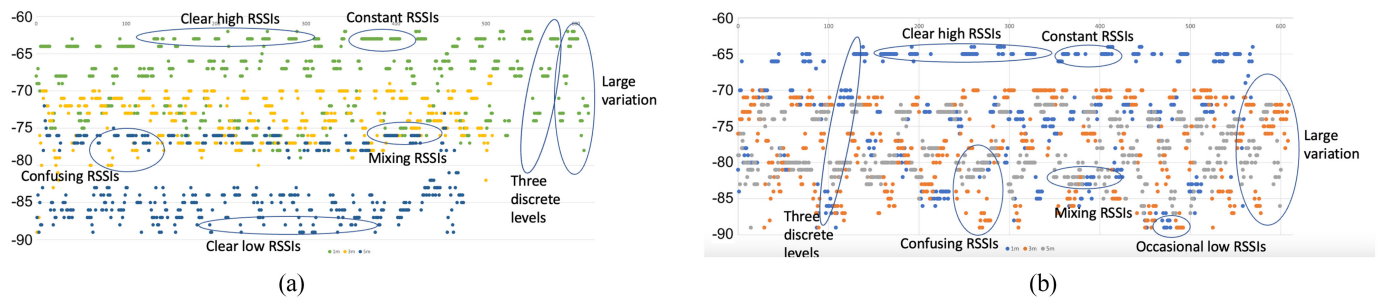
Fig. 2. RSSI analysis. (a) Phone 1: blue: 5 m, yellow: 3 m, green: 1 m. (b) Phone 2: gray: 5 m, orange: 3 m, blue: 1 m.

method is designed based on findings from the RSSI analysis. Finally, we discuss the MAP scenario with an iterative algorithm for estimating mobile user positions in an ad-hoc environment.

A. RSSI Analysis

To develop the hybrid learning method, we have conducted many RSSI-related experiments intending to identify the key issues affecting distance estimation accuracy. Fig. 2 shows the RSSIs when two different mobile phones receive BLE broadcast packets (i.e., during the broadcast phase) at 1, 3, and 5 m from the same broadcaster (i.e., a sending mobile phone), respectively. It can be seen that the RSSIs are highly varied over time. Although the two mobile phones receive the same BLE packets or signals (i.e., from the same sender with the same transmission power), the measured RSSIs can be quite different. The figure shows that the patterns have both similarities and differences. Even with the same mobile phone, there can be a large RSSI variation for the same distance. For example, as shown in Fig. 2(a), the RSSI for 1 m can vary between −60 and close to −80. Note that the unit of RSSI is in dBm. For the other mobile phone, the RSSI for 1 m can sometimes drop to nearly −90, even lower than the RSSI for 3 and 5 m. The two figures also show three discrete levels of RSSI with distinctive RSSI variations. One possible reason is that the BLE packets are sent over three separate channels. However, on most mobile phones, channel-related information is currently not available (e.g., through the APIs of the phone's operating system). Hence, in most previous work, the RSSIs are considered over one aggregate channel, although the RSSIs can vary within each individual channel. We believe that this may lead to inaccurate distance estimation. Possibly within the same channel, the RSSIs for the same distance can occasionally remain relatively constant [see both Fig. 2(a) and (b)]. On the other hand, the RSSIs for 1, 3, and 5 m can sometimes have the same values. Note that for example, the RSSIs for the worst channel for 1 m may be similar to those for the best channel for 3 or 5 m. In some cases, there may even be confusing RSSIs. For example, as shown in Fig. 2(b), the RSSIs for 3 m may even be much lower than those for 5 m. Fig. 2 also shows there are clear high RSSIs for 1 m. In other words, these RSSIs cannot be reached at 3 or 5 m, irrespective of the channels. That means, if these RSSIs are detected, the distance range can be more certain. Fig. 2(a) shows some clear

low RSSIs as well. However, Fig. 2(b) does not show similar patterns for the other mobile phone. For low RSSIs, it can be a result of other factors.

B. Hybrid Learning—Training Phase

Based on the aforementioned RSSI analysis, the proposed hybrid learning method seeks to enhance RSSI-based distance estimation with the following unique features. First, unlike previous work in which an aggregate channel is considered [51], unsupervised learning is employed to identify three pseudo channels (i.e., clusters) for enhanced RSSI processing. Second, as there are confusing or mixing RSSIs, a flagging method is proposed to alert users as to whether an estimated distance is predicted to be accurate or not (e.g., so that it can be excluded or remeasured). Third, apart from using data analytics, a pattern-inspired mechanism is employed to enhance the machine learning process. Fourth, instead of using one machine learning model, a new model aggregation method (with further flagging) is proposed to further enhance distance estimation accuracy. As shown by the experimental results in the next section, the hybrid learning method with the aforementioned features can provide a significant improvement over conventional methods. Fig. 3 shows the flow diagram of the hybrid learning method. In the training phase, there are six main steps: 1) RSSI data collection; 2) clustering of RSSI data; 3) range determination; 4) RSSI cluster assignment; 5) distance estimation training; and 6) elimination model training. These steps are explained as follows.

1) RSSI Data Collection: In order to build a distance estimation model for a pair of mobile phones (i.e., one as a broadcaster and the other one as a scanner) for different distances, RSSI data are collected at various sampling distances, such as 1 m, 2 m, . . . , 6 m. Note that although the sample distance is discrete, regression models are used for machine learning. As an example, 5 min of RSSI data are collected at each sample distance. With a common broadcasting interval setting at 100 ms, there are approximately 2000–3000 RSSI measurements for every sample distance. Twenty data samples or RSSI records are grouped into one data set, called RSSI record group for further processing. In other words, one RSSI record group (i.e., the fundamental unit for hybrid learning) associated with a distance comprises $r = 20$ RSSI records.

2) Clustering of RSSI Data: As mentioned, during the BLE advertising phase, BLE packets are sent over three advertising
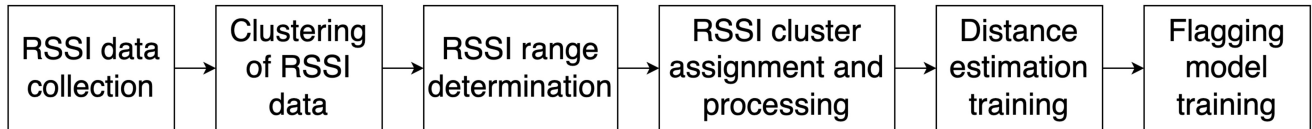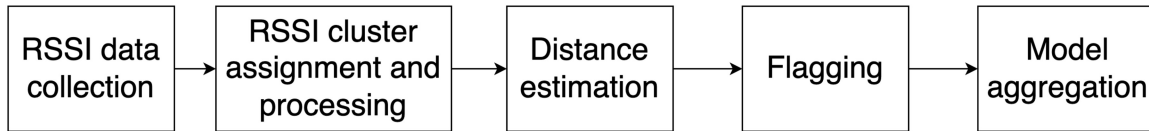
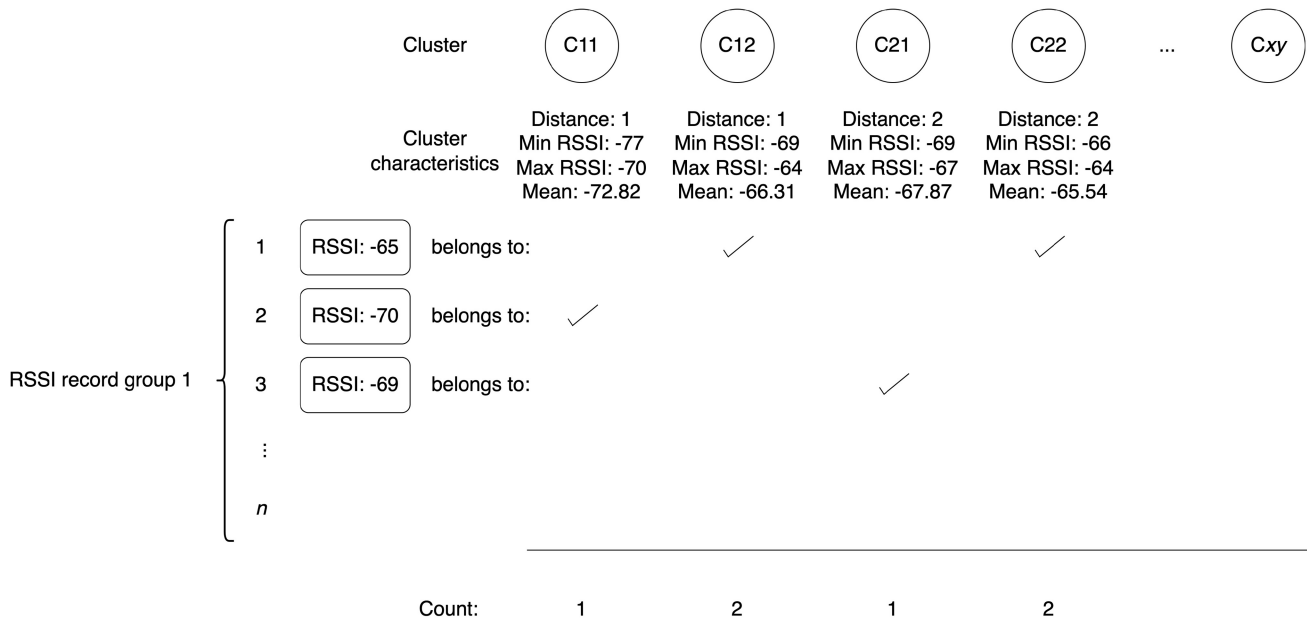Fig. 3. Flow diagram of the hybrid learning method.

Fig. 4. Example of cluster assignment for one RSSI record group.

channels (i.e., RSSI varies over three channels, possibly with different characteristics). However, as channel information is usually not available through the mobile phone's API, only one aggregate channel is considered in conventional approaches (i.e., considering that RSSI varies over one rather than three separate channels). Here, we use unsupervised learning (i.e., clustering) to cluster RSSI data for each distance into three groups or pseudo channels by means of k-means clustering. Suppose there are n_d sample distances, there will be $3n_d$ clusters. For example, if there are six sample distances, there will be 18 clusters. Within each channel, it should be reasonable to assume that RSSI should vary more consistently. The use of clustering seeks to identify the separate channels with the aim of enhancing the later machine learning process. Note that the clusters can be viewed as pseudo channels.

3) RSSI Range Determination: After generating the clusters or pseudo channels, the RSSI range (minimum and maximum) of each cluster is determined. The range information is to facilitate the cluster assignment in the next step. Fig. 4 shows a cluster rule example as determined by k-means clustering. For distance x, Cxy denotes cluster y of distance x.

For each distance, there are three clusters or pseudo channels. Note that like frequency channels, clusters for different distances may overlap with one another, but clusters for the same distance are distinctive. For example, as shown in Fig. 4, C12 (cluster 2 at distance 1) is overlapped with both C21 (cluster 1 of distance 2) and C22 (cluster 2 of distance 2).

4) RSSI Cluster Assignment and Processing: In an RSSI record group, each RSSI record is assigned to the clusters based on the aforementioned rule or RSSI ranges. For example, an RSSI record of -65 should belong to cluster 2 of 1 m (C12) and cluster 2 of 2 m (C22). Note that an RSSI record may be assigned to multiple clusters. The cluster assignment example can be found in Fig. 4. As an analogy, the RSSI records look like "pixels" in an image (i.e., they can be viewed as an image-inspired innovative approach).

After the assignment, the count of each cluster is determined (see an example in Table III). The cluster counts are then normalized as shown in Table IV. This normalization process seeks to establish the distribution or pattern to facilitate the data training. As an analogy, the normalized cluster count gives the "image intensity."

TABLE III
CLUSTER COUNT EXAMPLE

Distance	C11 Count	C12 Count	C13 Count	C21 Count	C22 Count	C23 Count
1	0	5	15	1	4	0

TABLE IV
NORMALIZED CLUSTER COUNT EXAMPLE

Distance	C11 Normalized Count	C12 Normalized Count	C13 Normalized Count	C21 Normalized Count	C22 Normalized Count	C23 Normalized Count
1	0	0.2	0.6	0.04	0.16	0

5) Distance Estimation Model Training: The normalized cluster information (Table IV) is used as the features (input) of the machine learning model for distance estimation training purposes (i.e., with the actual distance as the label). Compared to other RSSI-based training models (e.g., regression) which normally use RSSI as the input, our proposed method uses the normalized count of the 18 clusters as the estimation model input. In the example, there are 18 features corresponding to the normalized count of the 18 clusters. In essence, they provide the distribution/pattern for the corresponding distance. Four machine learning regression models are used for the training/prediction, namely, random forest (RF), kNN, SVM, and neural network (NN). They will also be used for the model aggregation to be explained later. For the training, the cross-validation method is used to build the model. The model is then saved for the testing/estimation phase.

6) Flagging Model Training: Flagging is an optional feature that seeks to identify uncertain distance estimation with the aim of further enhancing distance estimation accuracy. Basically, after building the distance estimation training model, the distance of each RSSI record group is predicted based on the model using the corresponding cluster pattern. The predicted distance is then compared to the corresponding actual distance. If the absolute error is greater than a certain comparison threshold, the RSSI record group is flagged. That means, the corresponding cluster pattern may lead to a poor or uncertain prediction/estimation. Unless otherwise specified, we shall use 1 m as the comparison threshold as an example. A flagging model training is then conducted using all RSSI record groups with the label "flag" or "not flag". The trained model is used to predict whether or not the input cluster pattern is flagged (i.e., for the testing data). If an RSSI record group is flagged, the predicted result may be disregarded because the estimated distance may not be accurate (e.g., due to confusing RSSI). In this case, additional measurement(s) can be taken until a better prediction is obtained. Of course, there is a tradeoff. If the threshold is too small, many remeasurements are required. The investigation of this tradeoff will be presented in the next section.

C. Hybrid Learning—Testing/Prediction Phase

As shown in Fig. 3, most steps for the testing/prediction stage follow that of the training stage. Similarly, to estimate the distance from another phone, 20 RSSI records are collected (i.e., forming one RSSI record group for processing). The RSSI records are assigned to each cluster based on
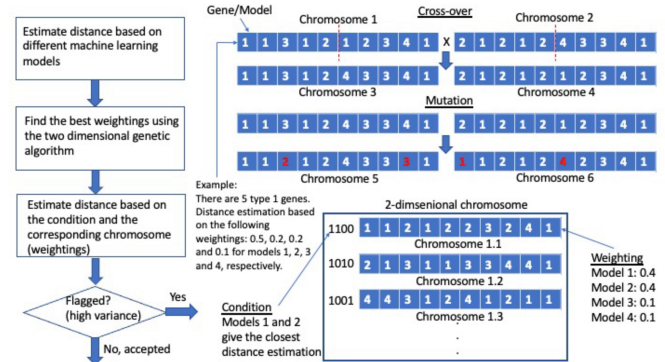

Fig. 5. Model aggregation using GA.

the predetermined clustering rule. The cluster counts and normalized cluster counts are determined similarly to the above. Based on the normalized cluster counts and the trained machine learning model (i.e., the distance estimation training model in Section III-B5), the estimated distance is predicted. Furthermore, based on the flagging model (i.e., Section III-B6), the estimated distance may be flagged (i.e., good or bad prediction) based on the training cluster patterns. If it is flagged (i.e., potentially inaccurate distance estimation), the user can retake the measurement until it is not flagged. Optionally, model aggregation can be used to further enhance accuracy, which is explained in the next section.

D. Model Aggregation

To further enhance the hybrid learning method, we present a novel GA-based model aggregation scheme. In essence, the aim is to aggregate the models (i.e., to determine the best weighting for the machine learning models, such as RF, SVM, kNN, and NN such that the overall MAE can be minimized). The best weighting is found by a GA. As shown in Fig. 5, a chromosome has ten genes, each represented by one of the models. Each gene contributes to 10% (or 0.1) of the overall weighting. Hence, in the example, as there are four genes for model 1, model 1 should have a weighting of 0.4. Chromosomes can be mixed through a crossover process (e.g., chromosomes 1 and 2 are mixed to produce chromosomes 3 and 4 by exchanging some genes). The new chromosomes are evaluated based on the MAE (i.e., fitness function). Sometimes, they may be better than their parents in terms of the fitness function. Occasionally (i.e., with a certain probability), a chromosome can be changed through a mutation process to introduce abrupt change. This seeks to

Algorithm 1 Model Aggregation—GA

Input: Estimated distances by the machine learning models: RF, kNN, SVM, NN
Output: Best weighting w for each model
 1: C = Randomly initialize a two-dimensional chromosome pool
 2: **while** iteration times **do**
 3:     P = Randomly assign the chromosomes into pairs
 4:     **for** (c_i, c_j) in P **do**
 5:         c_n, c_m = Crossover(c_i, c_j) // for all six sub-chromosomes (e.g., cross-over of c_{i1} with c_{j1})
 6:         Mutation(c_n, c_m) with a probability of 0.1
 7:         Put (c_n, c_m) into C
 8:     **end for**
 9:     C = Best 100 two-dimensional chromosomes based on the MAE (fitness function) as computed by the actual distances, estimated distances, and model weightings
10: **end while**
11: **return** Best weighting w based on C

Fig. 6. Hybrid learning model.

facilitate the search for a global minimum rather than a local minimum for MAE. In our case, with a certain small probability (e.g., 0.1%), some genes/models are randomly changed to another gene/model. For example, chromosome 3 and chromosome 4 are changed to chromosome 5 and chromosome 6, respectively, through a mutation process as shown by the red genes. Starting from a random population of chromosomes and after a certain number of generations of crossover and mutation, the best chromosome(s) and hence the best weighting for the models can then be determined.

To further enhance the scheme, we propose an innovative 2-D GA scheme (Algorithm 1). Basically, each chromosome can now be 2-D instead of 1-D. By default, a general chromosome can also be defined. If certain conditions are met, the corresponding chromosomes will then be used. For example, as shown in Fig. 5, if the estimated distances found by model 1 and model 2 are the closest (e.g., as represented by the bit vector), the chromosome 1.1 should be used for the weighting. Similarly, if the estimated distances found by model 1 and model 3 are the closest, chromosome 1.2 should be used. The aforementioned crossover and mutation process can be extended to the 2-D chromosomes (e.g., chromosome 1.1 and chromosome 2.1 can be mixed to produce chromosome 3.1 and so forth). The rationale of the aforementioned scheme is that if two independent models find similar estimated distances (i.e., a particular condition), the two models may receive a higher weighting. Note that other conditions can also be defined for the 2-D GA scheme, making it a general scheme. Similar to the 1-D GA process, the best weightings can be found by the 2-D GA process.

Last but not least, we also propose a flagging mechanism/scheme. If the estimated distances of the models highly deviate (i.e., as reflected by the variance of the distances estimated by the models), it indicates that the estimated distance (i.e., by the aggregated models) may not be reliable (e.g., due to the data quality or noise). In this case, it is better not to accept the result or to alert the user of its possible inaccuracy (e.g., flag the result or alert the user to take another measurement). In the later experiments, we shall evaluate this flagging mechanism as well.
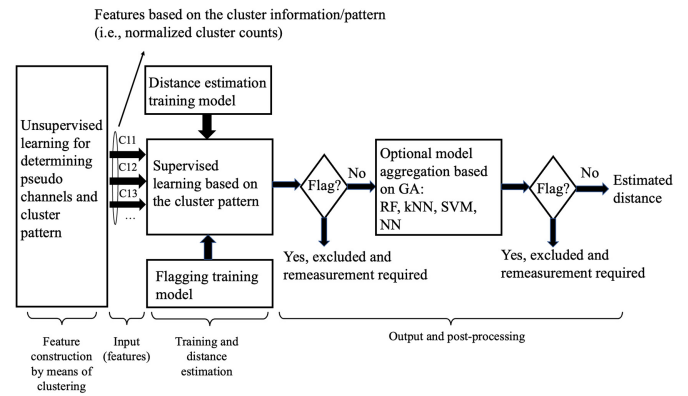
In summary, Fig. 6 shows the hybrid learning model. Unsupervised learning is used to discover pseudochannels for enhanced RSSI data processing. Note that this is unlike conventional RSSI-based machine learning methods (i.e., this is a new approach). The result of the unsupervised learning, i.e., the clustering information, is used as the input for the supervised learning model. More specifically, normalized cluster counts as discussed above are used as the training features. If there are s sampling distances, there will be $3s$ features for the training process due to the three pseudo channels associated with the three clusters. That means, the number of features depends on the required sampling distances. Note that the features seek to capture the channel or cluster pattern. Supervised learning is employed for distance estimation based on the cluster pattern. Flagging is used to exclude uncertain estimated distances (i.e., if an estimated distance is less certain, it can be excluded). Model aggregation with further flagging is employed to combine the results of different machine learning models with a 2-D GA scheme. This seeks to further enhance the hybrid learning method by integrating various machine learning models. Table V shows an example of the prediction records with the distance estimation, error, and flagging result.

E. Mobile Ad-Hoc Positioning

In this section, we discuss the MAP scenario. The basic question is that after estimating the distances between the phone pairs using hybrid learning, how can the phone positions be estimated? Note that this is a new interesting positioning problem that has seldom been studied in literature, as the conventional trilateration method (e.g., by means of three beacons) cannot be used due to its ad-hoc nature. Suppose there are n users/phones in the MAD/MAP system with an area of $w_1 \times w_2$ (all units in m) and they can share the estimated distances between one another through the ad-hoc network. The coordinate of the phone i is (x_i, y_i). The actual distance between phones i and j is d_{ij}. Based on the hybrid learning method, the measured distance of phone j from phone i is estimated to be m_{ij}. Note that m_{ij} and m_{ji} may not be the same due to different algorithm estimations, but the difference in general should be small.

The objective of MAP is to estimate the coordinate of each phone, i.e., the estimated coordinate of phone i: a_i and b_i.

TABLE V
FLAGGING EXAMPLE

Distance	C11 Normalized Count	C12 Normalized Count	C13 Normalized Count	C21 Normalized Count	C22 Normalized Count	C23 Normalized Count	Estimated Distance	Absolute Error	Flag
1	0	0.2	0.6	0.04	0.16	0	1.3	0.3	No
1	0.2	0.2	0.3	0.15	0.05	0	3.2	2.2	Yes

Algorithm 2 MAP Iterative Algorithm

Input: Number of phones N, Array of pair-distance of each phone pair A, Width of the area: w_1, Length of the area w_2
Output: Array of estimated positions
1: initialize estimated positions P
2: $P[0...n-1] = (w_1/2, w_2/2)$
3: $i = 0$
4: **while** $i < 10000$ **do**
5:   **for** all pair-distance d in A **do**
6:     $d_{max} = \text{Max}(A)$
7:     $B = \text{GetPhoneFromLargestPairedDistance}(d_{max})$
8:     **if** $B[0]$ is AnchorPhone **then**
9:       $PhoneToUpdate = B[1]$
10:     **else**
11:       $PhoneToUpdate = B[0]$
12:     **end if**
13:     UpdatePosition($PhoneToUpdate$)
14:     $Cost_i = Cost(P)$
15:     **if** $Cost_i < Cost_{i-1}$ **then**
16:       Update position in P
17:     **else**
18:       Ignore position
19:     **end if**
20:   **end for**
21:   $i++$
22: **end while**

Fig. 7. Experimental setup.

For coordinate set up, we assume that there are three anchor points. To facilitate the discussion, they can be represented as three fixed phones (i.e., anchor points/phones).

Initially all users/phones are assumed to be in the middle of the area, i.e., $a_i = w_1/2$ and $b_i = w_2/2$ for all values of i. Based on the current estimated coordinates, the estimated distance l_{ij} can be computed. The difference of the estimated distance with the measured distance can then be determined, i.e., $l_{ij} - m_{ij}$. The objective is to minimize the cost, i.e., the sum of the squared differences, as shown in

$$\text{Cost} = \frac{1}{c}\sum_{k=1}^{c}(l_k - m_k)^2 \qquad (1)$$

where c is the number of combinations of paired distances between two phones (i.e., $c = {}_N C_2$ where N is the number of users), l_k is the estimated distance of the paired distance of two phones i and j (i.e., $l_{i,j}$) and m_k is the corresponding measured distance.

An iterative algorithm (see Algorithm 2), inspired by the gradient descent algorithm, is used to determine the coordinates. In each iteration, the largest value of $(l_{ij} - m_{ij})^2$ is found in order to minimize the objective function (e.g., suppose that phones i and j are found). At first, the initial step size and direction are decided, such as $-0.1w$ or $0.1w$. After moving one of the phones, the cost is calculated. If the new cost is larger than the previous cost, the step is discarded and in the next iteration, a new step and direction are tried. If the cost is lower, the change is adopted. The coordinates and distances for the whole system are then updated accordingly. Upon updating, the iteration will then repeat for the next largest $(l_{ij} - m_{ij})^2$. The iterations will stop when a certain number of iterations is reached or the target sum of the square differences is fulfilled.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Experimental Setting

To evaluate the proposed hybrid learning method, we have conducted extensive experiments in an indoor environment (i.e., a room). The experimental setting for the base experiments is outlined in Fig. 7. The room is a computer laboratory about 7×12 m^2 in size, with computers placed at the edge of the room, and chairs and desks placed around the room. The experiments were carried out using a variety of mobile phones, as shown in Fig. 7. These four types of Android mobile phones (A, B, C, and D) cover different hardware configurations. Each mobile phone can be a broadcaster and a scanner, so there are 12 combinations (i.e., phone pairs). To advertise and scan
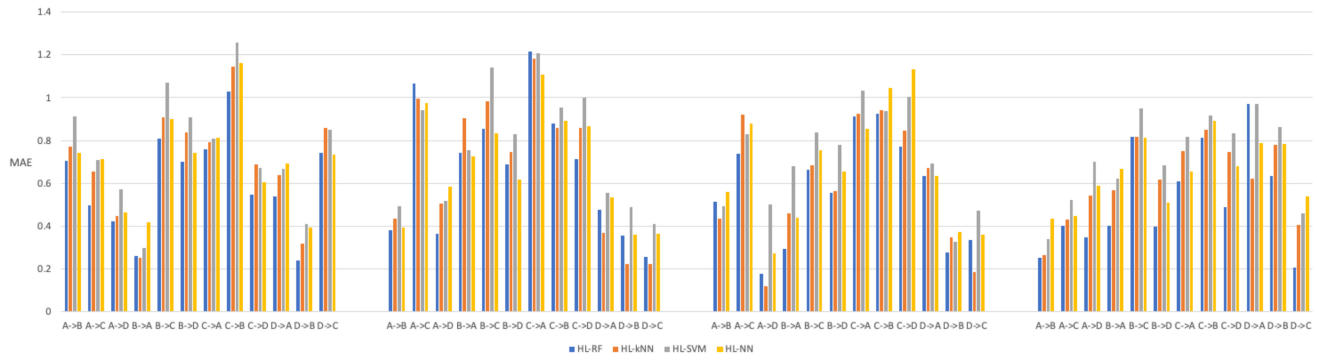
Fig. 8. MAE for different phone pairs for the daily experiments (Four result groups: day 1, day 2, day 3, and day 4).

Bluetooth signals, we used the Android built-in BLE library (namely, BluetoothLeScanner).

We have run the same experiment for four days (e.g., to check for data consistency) and have computed the overall average results. The objective of the experiments is to evaluate the different methods under the same conditions. On each day, RSSI data for each phone pair were collected at sampling distances of 1, 2, 3, 4, 5, and 6 m with two tripods (see Fig. 7). At each sample distance and for each phone pair combination, about 5 min of RSSI data/records were collected. Twenty RSSI records were grouped into one RSSI record group. With experiments taking place over four days, with 12 phone pair combinations and six sampling distances, more than 200 000 RSSI records were processed. The collected data were processed based on the hybrid learning method, basic learning method (i.e., conventional machine learning based on mean RSSI), and baseline method. Orange machine learning tool [52] and programs written in Python were used for data processing and machine learning purposes. Unless otherwise specified, 70% and 30% of the RSSI data (i.e., per phone pair) were used for training and testing, respectively. The machine learning baselines (i.e., basic machine learning methods) are based on the commonly used machine learning algorithms (i.e., similar to the related papers). The baseline method is based on the Android Beacon Model (also known as AltBeacon) [53] commonly used by the industry for distance estimation. The model uses the following formula to convert RSSI to distance:

$$d = A\left(\frac{r}{t}\right)^B + C \qquad (2)$$

where d is estimated distance, r is the measured RSSI, t is the reference RSSI measured at 1 m, and A, B, and C are constants. The default constant values provided by the model were used. The aim of the following experiments is to evaluate the performance of hybrid learning with these baselines.

B. Distance Estimation

Fig. 8 shows the MAE for the hybrid learning method using various machine learning models (RF, kNN, SVM, and NN) for the 12 phone pair combinations for the four-day experiments (i.e., four result sets from day 1, day 2, day 3, and day 4). It can be seen that while the trend in general is similar, the results depend on the phone pairs (i.e., the configurations
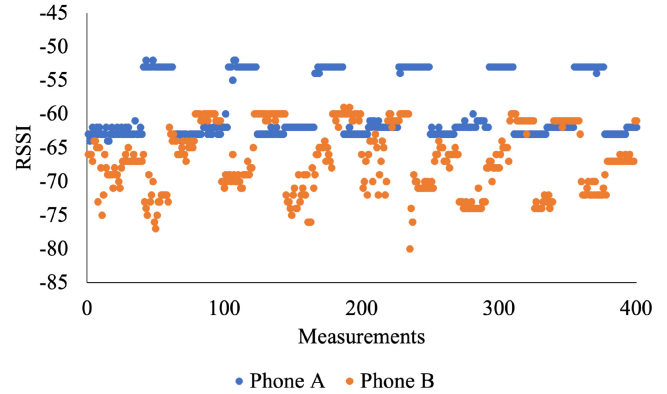

Fig. 9. Measurements from a broadcaster at the same distance by two different mobile phones.

of both the transmitting and receiving phones). Note that even for the same transmitting phone, the RSSIs measured by the receiving phones can be different. For example, Fig. 9 shows the distribution of measurements from a broadcaster taken by two different devices at the same distance. We can see that the measurements can vary, which is also explained in our previous findings [30], due to the difference in the hardware (e.g., Bluetooth chip, antenna, etc.). That means, device heterogeneity is a key issue. To take the important issue of device heterogeneity into consideration, data training should be conducted on a phone pair basis in practice. In fact, we used this training approach in our experiments. To cope with the vast number of phone models, a crowd-sourcing approach can be used in practice. For example, there can be a cloud database that stores the training data of different pairs of transmitting and receiving mobile phones (i.e., for different types of mobile phones, taking into account device heterogeneity). The data can be contributed through crowd sourcing and other suitable means.

Fig. 10 shows the overall average results for the experiments. The baseline method achieves an MAE of 1.53 m. Using basic machine learning (i.e., BL), the MAE can be enhanced to around 1 m. For example, with basic machine learning using RF (i.e., BL-RF), the MAE is about 1.07 m. With the hybrid learning method (i.e., HL in the figure), the MAE can be further enhanced to around 0.6–0.7 m. RF (i.e., HL-RF) provides better performance in general. By means of
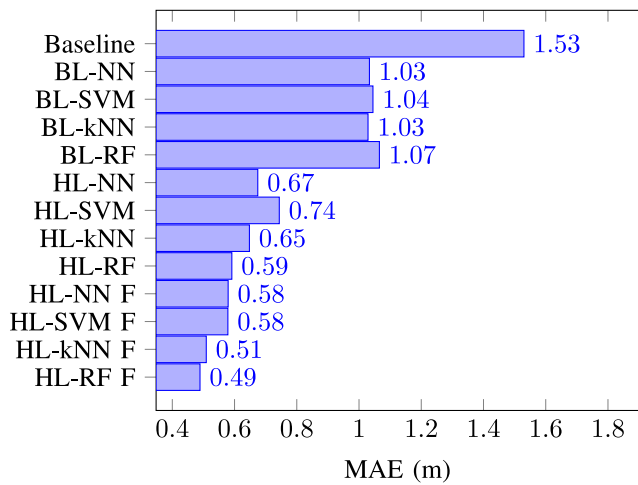
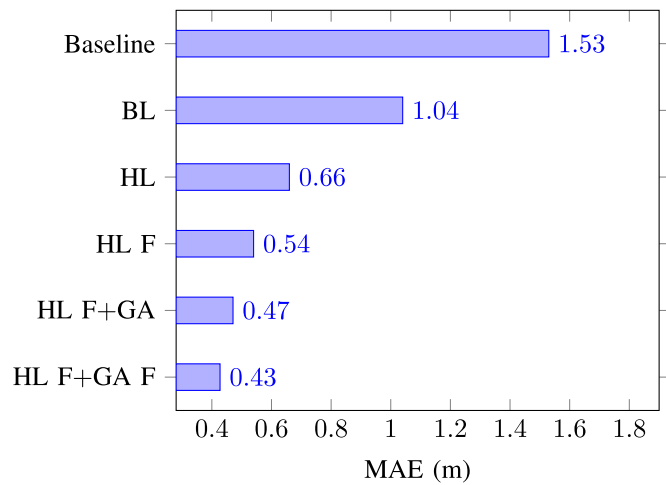Fig. 10.   Average MAE for different methods.

Fig. 12.   MAE and improvement of various methods.
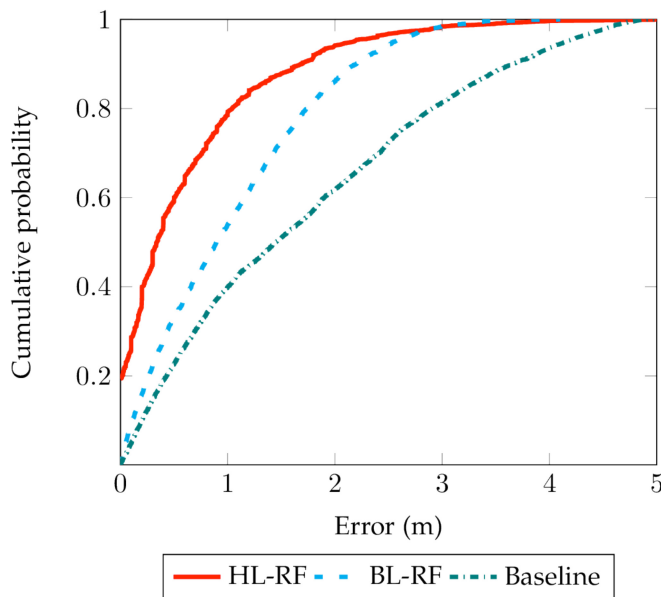

Fig. 11.   Cumulative distribution function of absolute error distance.
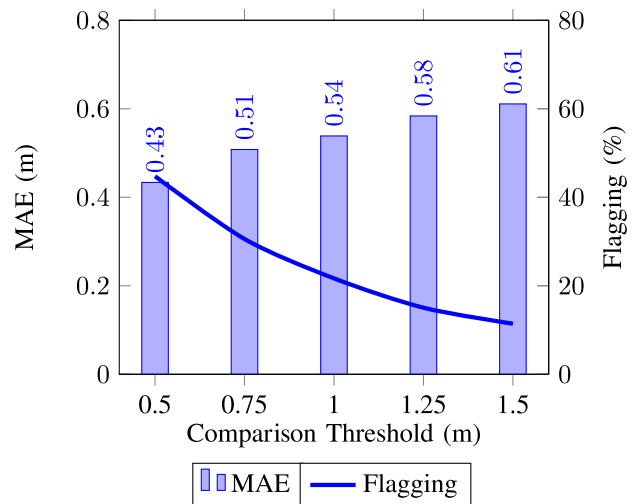

Fig. 13.   Tradeoff of flagging.

flagging (i.e., with F in the figure), the MAE can be slightly further enhanced, to around 0.5–0.6 m with hybrid learning using RF (i.e., HL-RF F), providing the best performance. Note that in the case of flagging, flagged results are not included in the calculation because the estimated distance may be uncertain.

To further evaluate the distance estimation results, Fig. 11 shows the cumulative distribution function of the absolute error. For the hybrid learning method, RF (i.e., the best one) is used (i.e., HL-RF). The graph shows that HL-RF outperforms the other two methods. Around 20% of the estimations have zero errors, and 80% of the estimated distances have an error within 1 m (i.e., without flagging).

Fig. 12 shows the average results of the four machine learning models (RF, SVM, kNN, and NN) as well as the model aggregation results (i.e., by means of 2-D GA). For the model aggregation, the first and last two days of distance estimation results are used for training and testing, respectively. With

basic machine learning (i.e., BL in the figure), the average MAE is 1.04 m, representing a 32% improvement over the baseline method. Using hybrid learning, the average MAE is 0.66 m, representing a 57% improvement over the baseline method or a further 25% improvement over the basic machine learning method. If flagging is used for hybrid learning, the average MAE can be further improved to 0.54 m or a 65% improvement over the baseline method. In other words, there can be a further 8% improvement compared to the nonflagging case. With flagging (i.e., using a comparison threshold of 1 m), 22% of the estimated distances are flagged or excluded. By means of model aggregation, a further 4% improvement can be realized. Finally, if further flagging is used by the model aggregation, the average MAE can be enhanced to 0.43 m, representing a 72% improvement over the baseline method. This is achieved at the expense of excluding an additional 11% of estimated distances. It is assumed that the excluded measurements can be retaken later.

Fig. 13 studies the tradeoff between flagging and MAE based on the average results of the four machine learning models. For example, by using a comparison threshold of 1 m, the
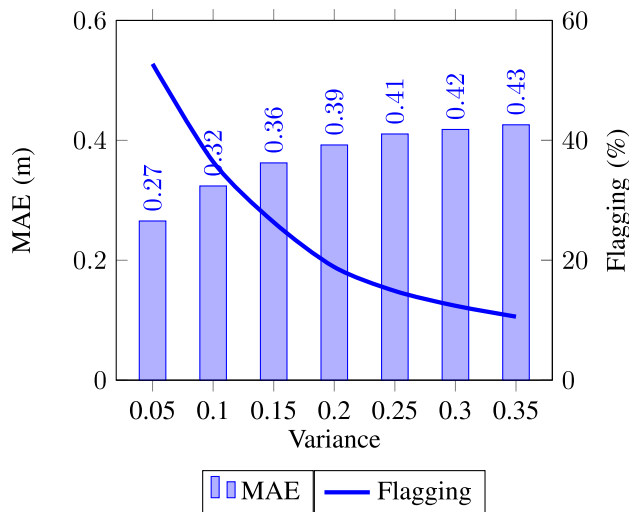
Fig. 14.   Tradeoff of flagging for model aggregation/GA.

Fig. 15.   Cluster patterns.

flagging percentage is about 22% (i.e., about 22% of the RSSI records are excluded) and an MAE of 0.54 m can be achieved. If the comparison threshold is lowered to 0.75 m, the flagging percentage becomes about 30%. In this case, the MAE can be further enhanced to 0.51 m. A reasonable flagging percentage should be acceptable. Note that for each flagged estimated distance, the measurement may need to be retaken. Furthermore, when an estimated distance is flagged, the estimation is likely to be inaccurate, although it may also occasionally be accurate. If the flagging percentage is too high, many measurements may become invalid. On the other hand, as discussed earlier, there may occasionally be confusing RSSIs. In these cases, it would be better to notify users that the measurements may not be accurate (e.g., due to uncertainty in the RSSI measurements) so they can be retaken later, rather than providing a potentially inaccurate estimated distance. Fig. 14 shows the tradeoff between flagging and MAE for the model aggregation scheme. In this case, when the variance of the estimated distance among the four models is greater than a certain threshold, the corresponding estimated distance is flagged and excluded from the calculation of the MAE. For example, if the variance is set to 0.35, the flagging percentage is about 11% and the achievable MAE is 0.43 m. While it is possible to achieve a lower MAE by setting a lower variance, the flagging percentage will rise (i.e., many estimated distances become invalid), which may not be acceptable.

Fig. 15 shows the sample cluster patterns (i.e., heatmap) for 1 m (first row), 3 m (second row), and 5 m (third row). The color intensity of each cluster is based on the normalized cluster count. The cluster patterns (i.e., based on the normalized cluster counts) provide the underlying mechanism for the machine learning process. It is expected that the cluster patterns for the same distance should be similar, thus facilitating the distance estimation. Note that for each sample distance, there may be multiple similar patterns and the ones in Fig. 15 are just examples. In general, this pattern-inspired approach complements the traditional approach. In the examples, it can be seen that the cluster patterns for the first and second columns look similar. However, the cluster patterns for
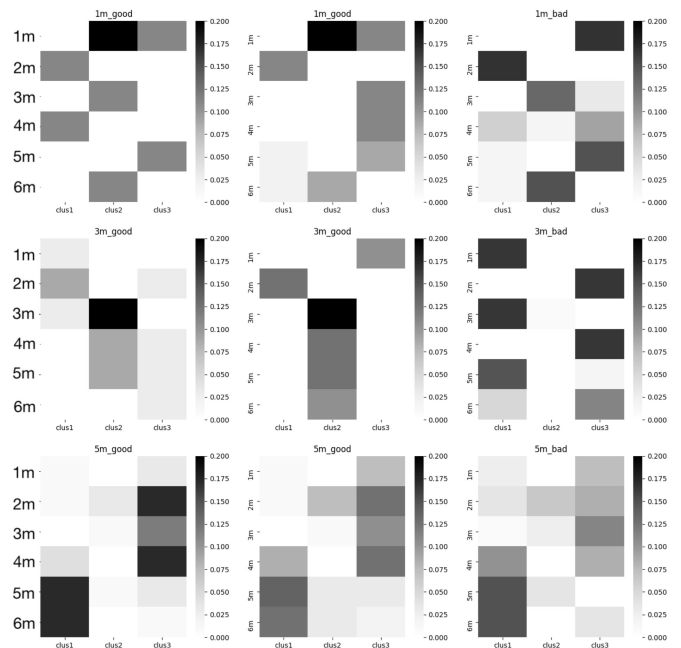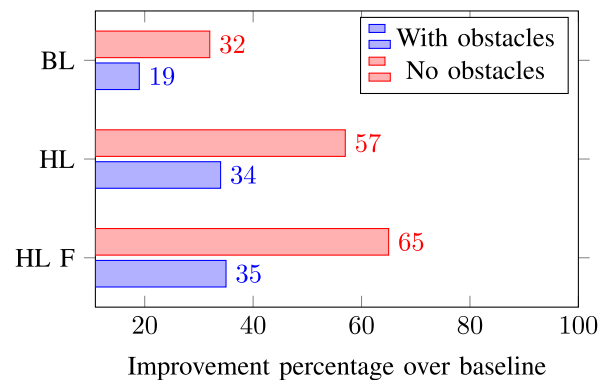

Fig. 16.   Distance estimation accuracy—improvement percentage over baseline.

the third column look different from the other two. Thus, the cluster patterns of the last column are flagged. To the best of our knowledge, this pattern-inspired approach for distance estimation is new and has not been studied in the literature. It has the potential to open a new area of research.

In general, it is well known that obstacles and environmental conditions can affect distance estimation accuracy of RSSI-based positioning methods. In other words, this is a common problem. In the base experiments, we assume that there was a clear environment with a reasonable line of sight so that the MAE of different methods can be compared under the same general experimental conditions. To investigate the basic effect of obstacles, we have also repeated an additional set of experiments on another day. In this case, two chairs, as obstacles, were put between a pair of phones. As discussed above, in terms of MAE, the basic learning method and hybrid learning method can achieve an improvement percentage of about 32% and 57%, respectively, over the baseline method in the base experiments. The objective of the additional experiments is to evaluate how the improvement percentage changes in
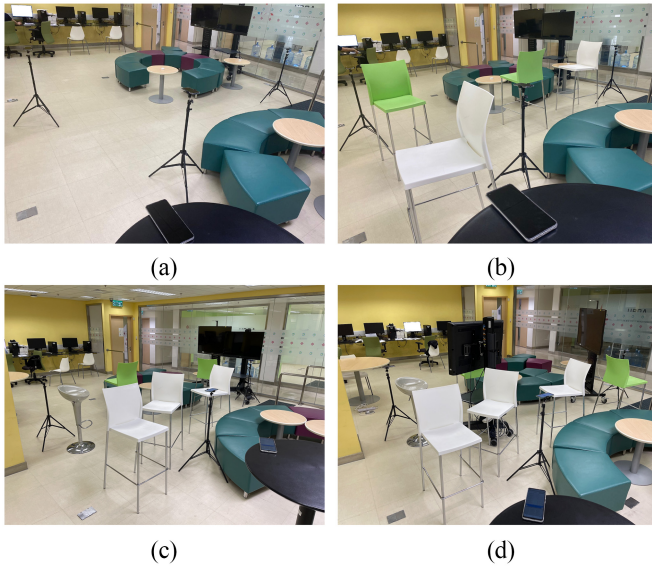
Fig. 17. Experimental setups for mobile ad-hoc positioning. (a) No obstacles. (b) Four chairs. (c) Six chairs. (d) Six chairs and two monitors.

the presence of obstacles. Fig. 16 shows that as expected, the improvement percentage drops with the presence of obstacles. For basic learning, hybrid learning, and hybrid learning with flagging, the improvement percentages become 19%, 34%, and 35%, respectively, with chair obstacles. That means, hybrid learning can still significantly outperform basic learning in the presence of obstacles. Note that this is an illustrative experiment to investigate the basic effect of obstacles. More studies can be conducted in future work.

C. Mobile Ad-Hoc Positioning

We have also conducted experiments to verify the feasibility of using MAD to support MAP based on the iterative algorithm (i.e., Algorithm 2) and to study the effect of obstacles. As mentioned above, this is not a conventional positioning problem because of the mobile ad-hoc nature. Our work should provide insights for future work in this relatively new area. Four sets of experiments were conducted, as shown in Fig. 17: one set without obstacles, one set with four chairs as obstacles, one set with six chairs as obstacles, and one set with six chairs and two monitors as obstacles. In each experiment, four mobile phones (A, B, C, and D) were placed evenly in the experimental area. For comparison purposes, the phones were placed at the same coordinates in all the experiments. For setting up the coordinates accordingly, three anchor points were used. The aforementioned hybrid learning method was used for distance estimation and the MAP algorithm (i.e., Algorithm 2) was employed for position estimation (i.e., to determine the estimated coordinates). As an example, hybrid learning RF (HF-RF) was used as the machine learning model. For the evaluation, RMSE between the actual coordinate and estimated coordinate was computed.

Fig. 18 shows the mean RMSE of four phones against the number of iterations. It can be seen that for both cases (i.e., without and with obstacles) the RMSEs converge to a steady state value after a certain number of iterations. This verifies
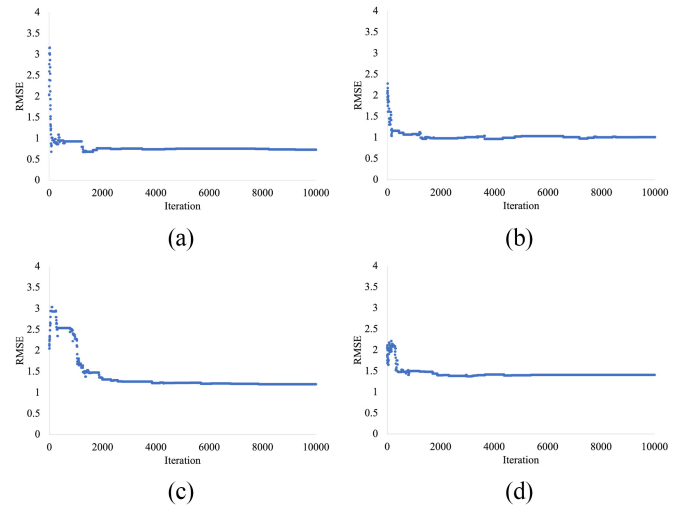

Fig. 18. Positioning error (in meter) of the MAP experiment over different iterations. (a) No obstacles. (b) Four chairs. (c) Six chairs. (d) Six chairs and two monitors.

TABLE VI
POSITIONING ERROR (IN METERS) FOR DIFFERENT
ITERATIONS—WITHOUT OBSTACLES

Iteration	A's RMSE	B's RMSE	C's RMSE	D's RMSE	Overall RMSE
0	2.50	2.06	2.50	1.12	2.04
50	1.13	0.71	1.19	3.05	1.52
100	0.83	0.71	1.19	1.33	1.02
500	1.03	0.71	0.92	1.14	0.95
1000	0.94	0.71	0.92	1.14	0.93
5000	0.86	0.20	0.92	1.05	0.76
10000	0.81	0.20	0.92	1.00	0.73

TABLE VII
POSITIONING ERROR (IN METERS) FOR DIFFERENT ITERATIONS—WITH
FOUR CHAIRS AS OBSTACLES

Iteration	A's RMSE	B's RMSE	C's RMSE	D's RMSE	Overall RMSE
0	2.50	2.06	2.50	1.12	2.04
50	2.02	1.06	0.72	3.61	1.85
100	1.70	1.33	0.98	1.74	1.44
500	1.76	1.33	0.98	0.38	1.11
1000	1.76	1.33	1.03	0.21	1.08
5000	1.68	1.25	0.75	0.46	1.03
10000	1.70	1.11	0.68	0.57	1.01

TABLE VIII
POSITIONING ERROR (IN METERS) FOR DIFFERENT ITERATIONS—WITH
SIX CHAIRS AS OBSTACLES

Iteration	A's RMSE	B's RMSE	C's RMSE	D's RMSE	Overall RMSE
0	2.50	2.06	2.50	1.12	2.04
50	2.50	1.81	4.14	3.33	2.94
100	2.50	1.81	4.14	3.70	3.04
500	1.80	1.90	3.65	2.82	2.54
1000	1.58	1.56	3.33	2.65	2.28
5000	1.68	0.59	0.91	1.74	1.23
10000	1.60	0.60	0.91	1.68	1.20

the feasibility of using the MAP algorithm for positioning estimation. Tables VI–IX show the RMSE of four phones as well as the overall RMSE. In the case of no obstacles, the overall RMSEs are about 0.73 m. In the case of four chairs,

TABLE IX
POSITIONING ERROR (IN METERS) FOR DIFFERENT ITERATIONS—WITH
SIX CHAIRS AND TWO MONITORS AS OBSTACLES

Iteration	A's RMSE	B's RMSE	C's RMSE	D's RMSE	Overall RMSE
0	2.50	2.06	2.50	1.12	2.04
50	1.12	0.50	1.47	3.95	1.76
100	1.97	0.16	1.76	3.95	1.96
500	1.81	0.49	1.95	1.68	1.48
1000	2.14	0.48	1.72	1.68	1.50
5000	2.07	0.46	1.57	1.50	1.40
10000	2.07	0.46	1.61	1.50	1.41

six chairs, and six chairs and two monitors, the RMSEs are about 1.01, 1.20, and 1.41 m, respectively. It means that the MAP algorithm can still operate in an environment with obstacles although the positioning accuracy is worse. In general, the effect of obstacles is a complex issue and further investigation can be carried out in future work.

V. CONCLUSION AND FUTURE WORK

In conclusion, we have presented a novel hybrid learning method for BLE-based distancing/positioning. As an innovative approach, it combines unsupervised learning, supervised learning, and GAs to enhance distance estimation accuracy. In addition to using data analytics, the underlying mechanism is based on a pattern-inspired approach to enhance the machine learning process. A flagging mechanism and a model aggregation scheme are also presented to achieve further enhancement. We have also presented a new MAP scenario with an iterative algorithm to estimate mobile user positions in an ad-hoc environment. Experimental results (based on the processing of more than 200 000 RSSI records for 12 phone pair combinations) show the hybrid learning method can achieve significant improvements over conventional methods. It is expected that the hybrid learning approach can also be applied to previous work for improving performance and provide new insights for further research. Future research work includes a further study on the pattern-inspired approach (e.g., exploring image recognition techniques), the MAP algorithm (e.g., developing advanced algorithms) and further investigation on the effect of obstacles.

REFERENCES

[1] "Number of smartphone users worldwide from 2016 to 2021." 2022. [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

[2] Y.-C. Pu and P.-C. You, "Indoor positioning system based on BLE location fingerprinting with classification approach," *Appl. Math. Model.*, vol. 62, pp. 654–663, Oct. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0307904X18302841

[3] H. Cho, D. Ippolito, and Y. W. Yu, "Contact tracing mobile apps for COVID-19: Privacy considerations and related trade-offs," 2020, *arXiv:2003.11511.*

[4] N. Ahmed et al., "A survey of COVID-19 contact tracing apps," *IEEE Access*, vol. 8, pp. 134577–134601, 2020.

[5] K. D. Pandl, S. Thiebes, M. Schmidt-Kraepelin, and A. Sunyaev, "How detection ranges and usage stops impact digital contact tracing effectiveness for COVID-19," *Sci. Rep.*, vol. 11, no. 1, p. 9414, 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-88768-6

[6] R. Natarajan, P. Zand, and M. Nabi, "Analysis of coexistence between IEEE 802.15.4, BLE and IEEE 802.11 in the 2.4 GHz ISM band," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc.*, 2016, pp. 6025–6032.

[7] D. Giovanelli and E. Farella, "RSSI or time-of-flight for Bluetooth low energy based localization? An experimental evaluation," in *Proc. 11th IFIP Wireless Mobile Netw. Conf. (WMNC)*, 2018, pp. 1–8.

[8] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019.

[9] F. Yin, Y. Zhao, F. Gunnarsson, and F. Gustafsson, "Received-signal-strength threshold optimization using Gaussian processes," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 2164–2177, Apr. 2017.

[10] Y. Zhao, C. Fritsche, F. Yin, F. Gunnarsson, and F. Gustafsson, "Sequential Monte Carlo methods and theoretical bounds for proximity report based indoor positioning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5372–5386, Jun. 2018.

[11] P. C. Ng, J. She, and S. Park, "High resolution beacon-based proximity detection for dense deployment," *IEEE Trans. Mobile Comput.*, vol. 17, no. 6, pp. 1369–1382, Jun. 2018.

[12] S. Sadowski and P. Spachos, "RSSI-based indoor localization with the Internet of Things," *IEEE Access*, vol. 6, pp. 30149–30161, 2018.

[13] Z. Jianyong, L. Haiyong, C. Zili, and L. Zhaohui, "RSSI based Bluetooth low energy indoor positioning," in *Proc. Int. Conf. Indoor Position. Indoor Navig. (IPIN)*, Oct. 2014, pp. 526–533.

[14] S. Subedi, G. R. Kwon, S. Shin, S. S. Hwang, and J.-Y. Pyun, "Beacon based indoor positioning system using weighted centroid localization approach," in *Proc. 8th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2016, pp. 1016–1019.

[15] X. Hou, T. Arslan, and J. Gu, "Indoor localization for Bluetooth low energy using wavelet and smoothing filter," in *Proc. Int. Conf. Localization GNSS (ICL-GNSS)*, Jun. 2017, pp. 1–6.

[16] V. C. Paterna, A. C. Augé, J. P. Aspas, and M. A. P. Bullones, "A Bluetooth low energy indoor positioning system with channel diversity, weighted trilateration and Kalman filtering," *Sensors*, vol. 17, no. 12, p. 2927, 2017.

[17] A. Thaljaoui, T. Val, N. Nasri, and D. Brulin, "BLE localization using RSSI measurements and iRingLA," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, 2015, pp. 2178–2183.

[18] J. Röbesaat, P. Zhang, M. Abdelaal, and O. Theel, "An improved BLE indoor localization with Kalman-based fusion: An experimental study," *Sensors*, vol. 17, no. 5, p. 951, Apr. 2017. [Online]. Available: http://dx.doi.org/10.3390/s17050951

[19] R. Faragher and R. Harle, "Location fingerprinting with Bluetooth low energy beacons," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2418–2428, Nov. 2015.

[20] G. D. Blasio, A. Quesada-Arencibia, C. R. García, J. C. Rodríguez-Rodríguez, and R. Moreno-Díaz, "A protocol-channel-based indoor positioning performance study for Bluetooth low energy," *IEEE Access*, vol. 6, pp. 33440–33450, 2018.

[21] C. Xiao, D. Yang, Z. Chen, and G. Tan, "3-D BLE indoor localization based on denoising autoencoder," *IEEE Access*, vol. 5, pp. 12751–12760, 2017.

[22] M. Li, L. Zhao, D. Tan, and X. Tong, "BLE fingerprint indoor localization algorithm based on eight-neighborhood template matching," *Sensors*, vol. 19, no. 22, p. 4859, 2019.

[23] D. Ordóñez-Camacho and E. Cabrera-Goyes, "An adaptive-bounds band-pass moving-average filter to increase precision on distance estimation from Bluetooth RSSI," in *Proc. Int. Conf. Inf. Technol. Syst.*, 2018, pp. 823–832.

[24] C. Zhou, J. Yuan, H. Liu, and J. Qiu, "Bluetooth indoor positioning based on RSSI and Kalman filter," *Wireless Pers. Commun.*, vol. 96, no. 3, pp. 4115–4130, 2017. [Online]. Available: https://doi.org/10.1007/s11277-017-4371-4

[25] M. Moneer, M. M. Aljawarneh, L. D. Dhomeja, G. Laghari, and B. R. Memon, "An indoor tracking system using iBeacon and Android," *Sukkur IBA J. Comput. Math. Sci.*, vol. 4, no. 2, pp. 61–68, 2020.

[26] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Trans. Veh. Technol.*, vol. 29, no. 3, pp. 317–325, Aug. 1980.

[27] C.-K. Ke, M. Wu, Y. Chan, and K. Lu, "Developing a BLE beacon-based location system using location fingerprint positioning for smart home power management," *Energies*, vol. 11, no. 12, p. 3464, Dec. 2018.

[28] R. K. Yadav, B. Bhattarai, H. Gang, and J. Pyun, "Trusted *K* nearest Bayesian estimation for indoor positioning system," *IEEE Access*, vol. 7, pp. 51484–51498, 2019.

[29] Z. Ma, S. Poslad, J. Bigham, X. Zhang, and L. Men, "A BLE RSSI ranking based indoor positioning system for generic smartphones," in *Proc. Wireless Telecommun. Symp. (WTS)*, 2017, pp. 1–8.

[30] Y. H. Ho and H. C. B. Chan, "Decentralized adaptive indoor positioning protocol using Bluetooth low energy," *Comput. Commun.*, vol. 159, pp. 231–244, Jun. 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366419309831

[31] D. Hollander. "How AoA & AoD changed the direction of Bluetooth location services." 2019. [Online]. Available: https://www.bluetooth.com/blog/new-aoa-aod-bluetooth-capabilities

[32] Q. Wang, Z. Duan, and X. R. Li, "Source localization with AOA-only and hybrid RSS/AOA measurements via semidefinite programming," in *Proc. IEEE 23rd Int. Conf. Inf. Fusion (FUSION)*, 2020, pp. 1–8.

[33] X. Wei, N. Palleit, and T. Weber, "AOD/AOA/TOA-based 3D positioning in NLOS multipath environments," in *Proc. IEEE 22nd Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2011, pp. 1289–1293.

[34] P. Davidson and R. Piché, "A survey of selected indoor positioning methods for smartphones," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1347–1370, 2nd Quart., 2017.

[35] K. Huang, K. He, and X. Du, "A hybrid method to improve the BLE-based indoor positioning in a dense Bluetooth environment," *Sensors*, vol. 19, no. 2, p. 424, Jan. 2019.

[36] B. Huang, J. Liu, W. Sun, and F. Yang, "A robust indoor positioning method based on Bluetooth low energy with separate channel information," *Sensors*, vol. 19, no. 16, p. 3487, Aug. 2019.

[37] F. Ye, R. Chen, G. Guo, X. Peng, Z. Liu, and L. Huang, "A low-cost single-anchor solution for indoor positioning using BLE and inertial sensor data," *IEEE Access*, vol. 7, pp. 162439–162453, 2019.

[38] Z. Li, K. Xu, H. Wang, Y. Zhao, X. Wang, and M. Shen, "Machine-learning-based positioning: A survey and future directions," *IEEE Netw.*, vol. 33, no. 3, pp. 96–101, May/Jun. 2019.

[39] T. Koike-Akino, P. Wang, M. Pajovic, H. Sun, and P. V. Orlik, "Fingerprinting-based indoor localization with commercial MMWave WiFi: A deep learning approach," *IEEE Access*, vol. 8, pp. 84879–84892, 2020.

[40] S. Bozkurt, G. Elibol, S. Gunal, and U. Yayan, "A comparative study on machine learning algorithms for indoor positioning," in *Proc. Int. Symp. Innov. Intell. Syst. Appl. (INISTA)*, 2015, pp. 1–8.

[41] K. N. R. S. V. Prasad, E. Hossain, and V. K. Bhargava, "Machine learning methods for RSS-based user positioning in distributed massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 8402–8417, Dec. 2018.

[42] E. Homayounvala, M. Nabati, R. Shahbazian, S. A. Ghorashi, and V. Moghtadaiee, "A novel smartphone application for indoor positioning of users based on machine learning," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. ACM Int. Symp. Wearable Comput.*, 2019, pp. 430–437. [Online]. Available: https://doi.org/10.1145/3341162.3349300

[43] X. Lu, H. Zou, H. Zhou, L. Xie, and G.-B. Huang, "Robust extreme learning machine with its application to indoor positioning," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 194–205, Jan. 2016.

[44] J. Zhao and J. Wang, "WiFi indoor positioning algorithm based on machine learning," in *Proc. 7th IEEE Int. Conf. Electron. Inf. Emerg. Commun. (ICEIEC)*, 2017, pp. 279–283.

[45] H. Q. Tran and C. Ha, "Improved visible light-based indoor positioning system using machine learning classification and regression," *Appl. Sci.*, vol. 9, no. 6, p. 1048, Mar. 2019. [Online]. Available: http://dx.doi.org/10.3390/app9061048

[46] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[47] H. Zou, H. Jiang, X. Lu, and L. Xie, "An online sequential extreme learning machine approach to WiFi based indoor positioning," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, 2014, pp. 111–116.

[48] S. BelMannoubi and H. Touati, "Deep neural networks for indoor localization using WiFi fingerprints," in *Mobile, Secure, and Programmable Networking (MSPN)* (Lecture Notes in Computer Science, 11557). Cham, Switzerland: Springer Int. Publ., 2019, pp. 247–258.

[49] Z. Chen, H. Zou, H. Jiang, J. Yang, and L. Xie, "WiFi fingerprinting indoor localization using local feature-based deep LSTM," *IEEE Syst. J.*, vol. 14, no. 2, pp. 3001–3010, Jun. 2020.

[50] S. Cheng, S. Wang, W. Guan, H. Xu, and P. Li, "3DLRA: An RFID 3D indoor localization method based on deep learning," *Sensors*, vol. 20, no. 9, p. 2731, May 2020.

[51] S. Naghdi and K. O'Keefe, "Trilateration with BLE RSSI accounting for pathloss due to human obstacles," in *Proc. Int. Conf. Indoor Positioning Indoor Navig. (IPIN)*, 2019, pp. 1–8.

[52] J. Demšar et al., "Orange: Data mining toolbox in Python," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2349–2353, 2013. [Online]. Available: http://jmlr.org/papers/v14/demsar13a.html

[53] M. Al Qathrady and A. Helmy, "Improving BLE distance estimation and classification using TX power and machine learning: A comparative analysis," in *Proc. 20th ACM Int. Conf. Model., Anal. Simulat. Wireless Mobile Syst.*, 2017, pp. 79–83. [Online]. Available: https://doi.org/10.1145/3127540.3127577

Yik Him Ho received the B.A. degree in computing and the Ph.D. degree from The Hong Kong Polytechnic University (PolyU), Hong Kong, in 2014 and 2021, respectively.

He is currently a Postdoctoral Fellow with the Department of Computing, PolyU. His research interests include Bluetooth low energy and cloud computing.
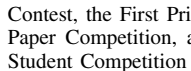
Dr. Ho has also received several local and regional IEEE awards, including Third Prize in the IEEE Hong Kong Section 2014 (UG) Student Paper Contest, the First Prize in the IEEE 2015 Region 10 Undergraduate Student Paper Competition, and the Honorary Mention in the 2016 IEEE ComSoc Student Competition Communications Technology Changing the World.

Yunfei Liu is currently pursuing the undergraduate degree with the Department of Computing, The Hong Kong Polytechnic University, PolyU, Hong Kong.

Mr. Liu received several scholarships, such as The Hong Kong Polytechnic University Scholarship 2021/2022 and 2022/2023. He also received the First Prize in the 36th Chinese Physics Olympiad in 2019.

Caiqi Zhang received the B.Sc. degree in computing from The Hong Kong Polytechnic University (PolyU), Hong Kong, in 2022. He is currently pursuing the M.Phil. degree with the University of Cambridge, Cambridge, U.K.

His research interests include Bluetooth low energy and indoor positioning machine learning algorithms.

Mr. Zhang was awarded the Outstanding Student Award of the Faculty of Engineering 2021 and the Outstanding Student Award of the Department of Computing 2021 at PolyU.

Yerkezhan Sartayeva received the bachelor's degree from The Hong Kong Polytechnic University, Hong Kong, in 2020, where she is currently pursuing the Ph.D. degree.

Previously, she worked as a Technical Development Lead with Tengizchevroil LLP, Satpayev, Kazakhstan. Her research focuses on the use of ultra-wideband technologies for indoor positioning.

Ms. Sartayeva is a recipient of the Hong Kong Ph.D. Fellowship.

Henry C. B. Chan (Member, IEEE) received the B.A. and M.A. degrees from the University of Cambridge, Cambridge, U.K., and the Ph.D. degree from the University of British Columbia, Vancouver, BC, Canada.

He is currently an Associate Professor and an Associate Head of the Department of Computing, The Hong Kong Polytechnic University (PolyU), Hong Kong. He has conducted various research projects and coauthored research papers published in a variety of journals. His research interests include networking/communications, Internet technologies, and computing education.

Dr. Chan was the recipient of the 2022 IEEE Education Society William E. Sayle II Award for Achievement in Education and the 2015 IEEE Computer Society Computer Science and Engineering Undergraduate Teaching Award. At PolyU, he has received four President's Awards and five Faculty Awards. Under his supervision/guidance, his students have received many awards. He was the Chair of the IEEE Hong Kong Section in 2012 and the Chair of the IEEE Hong Kong Section Computer Society Chapter from 2008 to 2009.