BlueTrk+ for Tracking Presence and Position

Yik Him Ho, Yun Fei Liu, Caiqi Zhang and Henry C. B. Chan

Abstract—**This article gives an overview of Bluetooth and Bluetooth Low Energy. Furthermore, it presents BlueTrk+, a lightweight application for tracking presence and position securely and flexibly for military-related or Disconnected, Intermittent and Limited (DIL) environments. For tracking presence, four schemes with varying security levels are presented for broadcasting identities. For tracking position, a polar coordinate-based approach is used. It involves using a radar-inspired method for estimating direction/angle and a machine learning method for estimating distance. Experimental results are presented to evaluate the positioning methods.**

Index Terms—**Military, BLE, DIL networks, Positioning, Tracking**

I. Introduction

Bluetooth aims to support short-range communications, typically covering a small, or personal area. As an extension of classic Bluetooth, Bluetooth Low Energy (BLE) provides a low energy version suitable for supporting Internet of Things (IoT) (e.g., sensor-based) applications in particular. While Bluetooth and BLE are designed for commercial applications in general, they can be extended or enhanced for military-related purposes. Indeed, Bluetooth in general and BLE in particular are well-suited for supporting various military-related applications.

In general, BLE can be used to support a mobile ad-hoc network for scenarios as defined by the NATO IST-147 group (i.e., scenarios where people are involved in military-related operations in a smart city) [1]. Some key services include information collection, group communications, people management and navigation control. One popular military-related application is to control robots through a mobile terminal. A commonly used Bluetooth module for research and development purposes is the HC-05 module, which facilitates communications with Arduino and other microcontrollers. With the HC-5 module and other microcontrollers, a tank-based military robot [2] or a small unarmed ground vehicle [3] can be built to support tracking and object detection (e.g., in a battlefield).

The IoT, particularly different types of sensors, plays an important role in unmanned vehicles for military operations. These sensors include pyroelectric, temperature, metal, gas, infrared and ultrasonic sensors for detecting/sensing heat radiation, temperature, metals/bombs, toxic gases and depth/distance, respectively [4]. Moreover, there are also wearable biosensors that monitor the physiological, cognitive and emotional condition of a person (e.g., a soldier) [5]. To facilitate sensor management, an open and integrated framework is required [5]. For all of the aforementioned sensors, BLE provides an effective, energy-efficient and flexible architecture/platform supporting development and operation.

Complementing the aforementioned Bluetooth/BLE applications, another potential military-related application using BLE tracks both presence and position. As conventional BLE is not primarily designed for military applications, enhancements are required. As military-related applications are more demanding and sensitive, security should be strengthened. Apart from security, existing BLE systems or applications are often infrastructure-based. Military-related environments are often ad-hoc in nature, which are so-called Disconnected, Intermittent and Limited (DIL) environments. That means, conventional infrastructure-based solutions (e.g., for positioning purposes) cannot easily be deployed in these environments. Hence, more flexible and adaptive solutions are required for DIL environments.

This article presents BlueTrk+ to tackle the aforementioned technical issues. The aim is to design a BLE-oriented lightweight application for tracking presence and position securely, flexibly and adaptively for military-related or DIL environments.

II. bluetooth and ble Overview

Bluetooth is designed to support short-range communications, typically covering a personal area network. A good overview of Bluetooth can be found in [6]. The protocol model has two major components: transport protocols and middleware protocols. The transport protocols have four core layers: radio layer, baseband layer, link manager layer and Logical Link Control and Adaptation Protocol (L2CAP) layer. Specified by the radio layer, Bluetooth operates over the 2.4 GHz band based on frequency hopping spread spectrum techniques. Based on the radio layer, the baseband layer specifies the communication mechanism for Bluetooth devices, including the network architecture and packet format. Bluetooth devices communicate with one another by forming an ad-hoc network called a piconet using a master-slave communication mechanism. The link manager protocol is for managing the link between two Bluetooth devices, such as authenticating with one another through a challenge-and response protocol and setting different operational modes. L2CAP provides an interface between the upper layer and lower layer protocols, allowing the multiplexing of logical channels. There are three types of logical channels: signalling channel, connectionless channel and connection-oriented channel. The middle layer provides various protocols to support specific functions, such as a service discovery protocol, for conveying service information. Based on the underlying middleware and transport protocols, various Bluetooth applications can be developed.

As an extension to classic Bluetooth, BLE has been included in the core Bluetooth standard (i.e., Bluetooth specification 4.0). However, while it is based on a similar/related framework, BLE is not compatible with classic Bluetooth. A good overview of BLE can be found in [7]. The BLE protocol model has three basic layers: physical layer, link layer and L2CAP layer, as well as three core elements: Attribute Protocol (ATT), Generic Attribute Profile (GATT) and Security Manager Protocol (SMP). At the physical layer, BLE devices operate on the 2.4 GHz band. There are 40 channels, including three that function as advertising channels. A BLE device can advertise (broadcast) data through these advertising channels. Indeed, the advertising process can be used to develop tracking and positioning applications. The advertising process also facilitates the discovery and connection of BLE devices. Once connected, two BLE devices can communicate at the link layer through a master-slave communication mechanism. The L2CAP layer in the BLE protocol model can be viewed as a simplified version of the L2CAP layer in the classic Bluetooth model. Its main purpose is to multiplex data from the upper layers. ATT defines a server/client-based protocol with various attributes stored in a server. Based on these attributes, various services can be defined and provided through GATT. SMP handles the security protocols. The focus is on security for connected BLE devices. A good overview of BLE security and privacy can be found in [8]. In the latest specification, to enhance BLE security/privacy, a Resolvable Privacy Address (RPA) is included based on the hash value of a random number and an Identity Resolving Key (IRK).

III. BlueTrk+ Overview

In this section, we provide an overview of BlueTrk+ for tracking presence and position in particular. The aim is to design a lightweight solution suitable for a DIL environment, focusing on broadcasting identities (IDs) securely and estimating positions in an ad-hoc manner (i.e., not infrastructure-based). Compared to other location tracking or positioning technologies such as WiFi, LoRa and RFID [9][10], Bluetrk+ has several advantages or new contributions in terms of effectiveness and/or performance. BLE is more energy efficient than traditional Bluetooth making it more suitable for tracking and positioning purposes. While WiFi is effective for communication purposes, it is more heavy weight and infrastructure-oriented for tracking or positioning purposes (i.e., less suitable for the DIL environment). For instance, a network of WiFi access points (i.e., an infrastructure) should be set up and the communication protocols are more complex. In contrast, Bluetrk+ provides a more lightweight tracking/positioning solution for an ad-hoc environment in particular (i.e., no infrastructure or pre-installed beacons/devices is/are required). Although LoRa can provide a similar lightweight solution, BLE provides a better peer-to-peer solution. Furthermore, a Bluetooth-oriented is beneficial because Bluetooth can also provide other functionalities such as cable replacement. While RFID (e.g., passive RFID) provides a cost-effective tracking and identification solution, BlueTrk+ support better security functions, which are important for the military-related applications. In particular, there are different levels of security for broadcasting IDs.

Imagine that there are objects (people and things) to be tracked in an open area (e.g., in a military-related environment). Like standard BLE, each object can broadcast an ID (i.e., hardware and/or logical ID). Based on the received signals, the received signal strength indicator (RSSI) can also be measured, depending on the distance between the sender and receiver. For conventional BLE, the broadcast ID can be read by anyone. Furthermore, as an object can be identified based on the same ID, there can be security concerns, especially for sensitive or military-related applications. Furthermore, most BLE-based solutions are designed for an infrastructure-oriented environment. In many military-related environments, ad-hoc solutions are often required (i.e., an infrastructure cannot easily be set up in a DIL environment). BlueTrk+ seeks to address these issues. In particular, it facilitates tracking presence and positions securely and adaptively (i.e., in an ad-hoc manner).

Let us first consider tracking presence based on an identity (ID). We consider a general approach based on hardware IDs (e.g., hardware addresses) and/or logical IDs (e.g., configurable object identifiers). To enhance security for broadcasting IDs, Fig. 1 shows four general schemes. In the first scheme, instead of broadcasting a real ID, the sender broadcasts the corresponding hashed ID. Upon receiving the hashed ID, the receiver can compare it with a list of hashed IDs to identify the original ID. As hashing is simple to implement, this scheme is energy efficient. However, a predefined list of IDs and the corresponding hashes is required. In other words, it may not provide a general solution (e.g., for detecting general or unknown objects). In the second scheme, the sender broadcasts an encrypted ID, such as the Advanced Encryption Standard (AES). We assume that all devices or objects share a group secret key. In this case, a predefined list of IDs is not required, but a group secret key should be maintained. Furthermore, as encryption is more processing intensive, this scheme is less energy efficient. For both the first and second schemes, although the original ID can be hidden, it is still vulnerable to replay attack. For example, after capturing messages with hashed or encrypted IDs, a fake object can broadcast the hashed or encrypted IDs (i.e., pretending to be the real object). One simple way to tackle this issue is to set up a number of virtual IDs linking to the real ID (i.e., the messages are broadcast based on the virtual IDs representing the same object). In the third scheme, a key is linked with the actual ID. The sender broadcasts a nonce together with a hash of the key combined

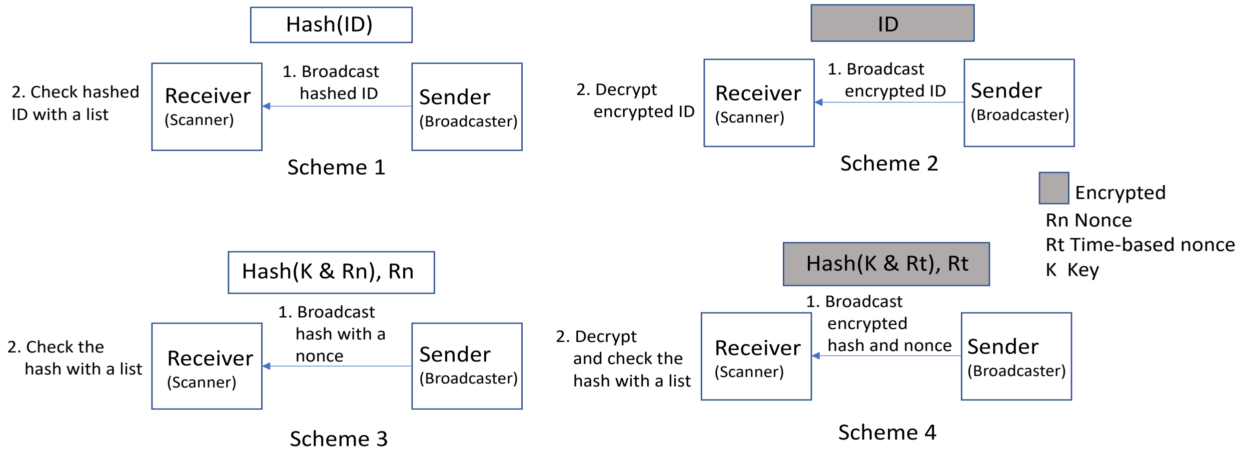


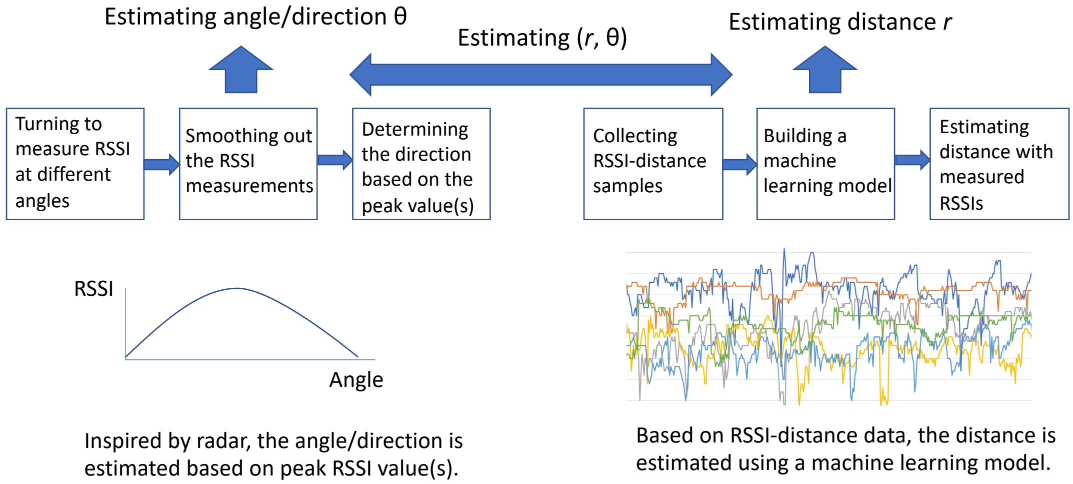
Fig. 1. Different schemes for securely broadcasting IDs for tracking presence

with the nonce. In this case, as a different nonce is broadcast each time, the hash will also be different. To check for the original ID, the receiver needs to compute the hash of the corresponding key of the ID together with the received nonce (i.e., by comparing the computed hash and the received hash). Note that this scheme is based on the security privacy mechanism as defined in the latest Bluetooth specification. In this case, the RPA and IRK (i.e., the one linking to the Bluetooth address) can be the hash and key, respectively. This scheme makes replay attack more difficult. For example, the receiver can keep a list of the received hashes to eliminate duplicated messages. The fourth scheme is an extension of the third scheme. Basically, the broadcast message is encrypted with a group key and a time-based nonce is used (e.g., including a time-stamp or time-related information in the nonce). This provides two levels of security. To check for the ID, the message should be decrypted with a group key first and then the hash should be checked with another key (e.g., IRK). Furthermore, as a time-based nonce is used, replay attack is extremely difficult because outdated messages can be eliminated. However, more processing is required (i.e., it is less energy efficient) and more information should be transmitted.

In summary, there are three considerations: security, processing and ease of implementation. For scheme 1, it can fulfill basic security by hiding the IDs. As hashing is used, it is efficient in terms of processing. For scheme 2, the security is stronger as encryption is used to ensure confidentiality. However, more processing and implementation complexity is required. In terms of security, scheme 3 can perform better than scheme 2 and scheme 3. In particular, replay attack can be tackled. Scheme 4 provides the highest level of security at the cost of more processing and high implementation complexity.

In a typical BLE-based positioning system, an infrastructure-based approach is often used. Basically, fixed beacons are used to estimate the positions (i.e., based on (x, y) coordinates) of an object based on the triangulation method in particular. However, this approach cannot easily be applied in a DIL environment. In other words, an ad-hoc and flexible approach is required. Inspired by radar, we consider using a polar coordinate-based approach (i.e., using polar (r, θ) coordinates instead of (x, y) coordinates). To track positions based on polar (r, θ) coordinates, there are two sub-problems: estimating direction θ using a radar-inspired mechanism and estimating distance r using machine learning. To estimate direction θ, a person first tries to find the detected object's strongest signal by circling (i.e., turning around with the mobile terminal to collect the signals and hence measuring the RSSI). Using the compass function (i.e., based on the magnetic field), the mean RSSI for various directions can be measured. To smooth out the signals or collected data for better detection of the peak signal, two lightweight methods are used. Experimental results will be presented in the next section. Inspired by the kNN algorithm, the first method computes the mean angle of the k (e.g., 3) samples with the top three RSSI. For example, suppose that we have the following measurements expressed in descending order of RSSI (35, -60), (32, -63), (33, -65), (40, -66), (42, -68), where the first and second numbers inside a bracket denote the direction/angle and RSSI, respectively. In this case, the estimated direction is (35+32+33)/3 = 33 degrees. The second method is inspired by finding moving averages in stock price charts. In this method, the n-point moving averages for both the RSSI values and angles is computed to determine the peak and hence the estimated direction/angle. For example, suppose we consider a three-point moving average (i.e., $n = 3$) and have the following measurements sorted in ascending order of angles: (26, -70), (29, -71), (32, -63), (33, -65), (35, -60). The three-point moving averages can be computed as: (29, -68), (31, -66), (33, -63). Note that for example, for the first one, the moving averages of angle and RSSI are computed as (26+29+32)/3 and the (-70-71-63)/3, respectively. Based on the largest moving average of RSSI (i.e., -63), the estimated angle is 33 degrees.

For distance estimation (i.e., to estimate distance r), machine learning (i.e., supervised learning) is used based on training RSSI, as shown in Fig. 2. According to Fig. 2, the machine learning process has three main processes: collecting RSSIs and distances for training purposes, building the machine learning model(s) and estimating distances based on the measured RSSIs and trained models. Pre-training can be conducted based on


Estimating angle/direction θ
Estimating (r, θ)
Estimating distance r
Turning to measure RSSI at different angles
Smoothing out the RSSI measurements
Determining the direction based on the peak value(s)
Collecting RSSI-distance samples
Building a machine learning model
Estimating distance with measured RSSIs
RSSI
Angle

Inspired by radar, the angle/direction is estimated based on peak RSSI value(s).

Based on RSSI-distance data, the distance is estimated using a machine learning model.

Fig. 2. Estimating angle/direction and distance

different environments so that users can choose the most similar or suitable environments. The training part is more processing intensive so it can be conducted by more powerful servers (e.g., in the cloud). Once the models are built, they can be deployed to the mobile terminals for implementation (i.e., estimating the distance based on measured RSSIs). This part is less complex and the processing is less intensive so it can be handled by mobile terminals. Re-training may also be conducted whenever and wherever required. In this case, data samples can be collected in the field and transmitted to a cloud for re-training. Then the trained model(s) can be downloaded to mobile terminals for implementation. In summary, the combination of cloud computing and mobile computing can facilitate effective implementation in terms of low complexity and efficient processing.

For the training, three approaches are studied. Various machine learning (regression) models such as Artificial Neural Network (ANN), kNN, Random Forest (RF) and Support Vector Machine (SVM) can be used for training and distance estimation. In the first approach, training is conducted based on mean RSSI (i.e., using the mean RSSI as the feature and the distance between the sender and receiver as the label). In the second approach, training is conducted based on peak RSSI values. The third approach is similar to the second approach, except that the minimum RSSI values are used for training purposes. Let us explain the approaches with a simple example. Suppose that we have two sets of time series data (i.e., measured RSSI values for 1 m and 2 m). For the first dataset for 1 m, the measured RSSI values are -60, -61, -62, -65 and -66, so the mean RSSI is -62.8. Similarly, for the second dataset for 2 m, the measured RSSI values are -63, -62, -65, -64 and -67, so the mean RSSI is -64.2. Hence, for the first approach, the feature values for the labels 1 m and 2 m are -62.8 and -64.2, respectively. For the second approach, to smooth out the data, we use n-point moving average to obtain the m highest RSSI values for the supervised learning process. For illustration purposes, we simply use a three-point moving average to obtain the highest value in the previous example (i.e., n = 3, m = 1). The moving averages of RSSI values for 1 m (starting from the third sample) are -61.0, -62.7 and -64.3 with the largest value of -61. Similarly, for 2 m, the moving averages of RSSI values are -63.3, -63.7 and -65.3 with the largest value of -63.3. Hence, in the second approach, the feature values for the labels 1 m and 2 m are -61 and -63.3, respectively. Again, for illustration purposes, let us use a three-point moving average (i.e., for smoothing out the data) to obtain the lowest RSSI value for the third approach. It can be found that the lowest values are -64.3 and -65.3 for 1 m and 2 m, respectively. Hence, in the third approach, the feature values for the labels 1 m and 2 m are -64.3 and -65.3, respectively.

In summary, BlueTrk+ seeks to track positions based on polar coordinates. First it detects the direction of the objects based on the peak RSSI. Then it estimates the distance based on machine learning using RSSI-based features. Experimental results will be presented in the next section to evaluate the proposed methods.

IV. EXPERIMENTAL RESULTS

To evaluate the proposed methods, we have developed a BlueTrk+ prototype (mobile app) based on BLE for conducting the later experiments. It can be used to receive BLE packets, detect RSSI and determine the corresponding direction based on the compass function of a mobile terminal. To facilitate data processing, the mobile app can also store the detected RSSI values and the corresponding directions in a CSV file. With the aid of the BlueTrk+ prototype, the following experiments and evaluations were conducted.

The first experiment aims to evaluate the direction estimation methods. As an illustrative example, Fig. 3 shows how real RSSI values vary with the angle/direction when a mobile terminal (i.e., using the BlueTrk+ mobile app) turns around. The actual angle/direction of the tracked object should be 207°, as shown in the figure. As the angle of the mobile terminal approaches 207°, it can be seen that the peak trend becomes apparent (i.e., as seen from the raw RSSI). However, for the raw RSSI, the highest RSSI value occurs at about 280° (point A), not close to the actual direction (i.e., there is another peak due to RSSI fluctuation). By considering a number of top RSSI values (points B, C and D), better angle/direction estimation can be made based on their mean values (i.e., closer to the actual angle). Furthermore, by using the five-point moving average method to smooth out RSSI values, the estimation can be improved based on the top RSSI values (see points E, F and G).

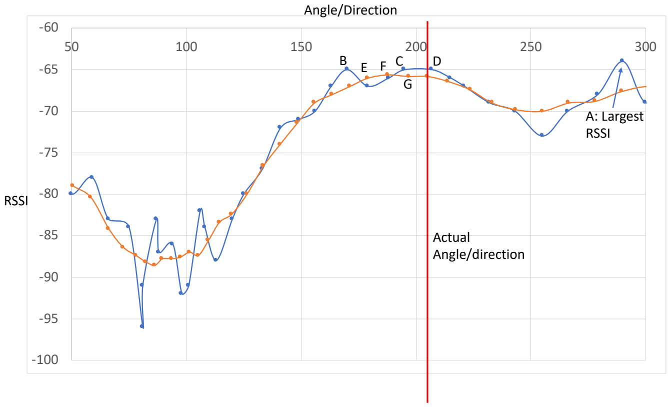


Fig. 3: Example for estimating angle/direction

Experiments were conducted to evaluate the direction/angle estimation methods. In an experiment, the direction of a tracked object was estimated at the sampling points of 1 m, 2 m, 3 m, 4 m and 5 m using the following methods (i.e., by means of circling, as mentioned previously):

- Raw-Max1/3/5: Raw RSSI data with mean of top 1/3/5 value(s)
- MA5-Max1/3/5: five-point moving average of RSSI data with mean of top 1/3/5 value(s)
- MA-10-Max1/3/5: ten-point moving average of RSSI data with mean of top 1/3/5 value(s)

At each sample point, 20 tests were conducted to collect the data in an outdoor environment. The aforementioned experiment was repeated twice at different time periods (e.g., to check for data consistency). The aggregated results were evaluated based on absolute angle error. Overall, more than 10,000 RSSI measurements were processed. Note that performance depends on a variety of factors (e.g., mobile terminal used, environmental factors). The aim of the experiments is to compare the aforementioned methods under the same conditions. Fig. 4 shows mean absolute angle error and standard deviation. It can be seen that estimation can be improved by considering several peak RSSI values (i.e., finding their mean values). Furthermore, the moving averaging method can further improve results. Among the methods, MA10-Max5 provides the best performance in terms of mean absolute error and standard deviation. It was also found that how a mobile terminal was held affected performance, due to the antenna position. In other words, it is important in experiments to determine the best way to hold the mobile terminal and use it consistently.

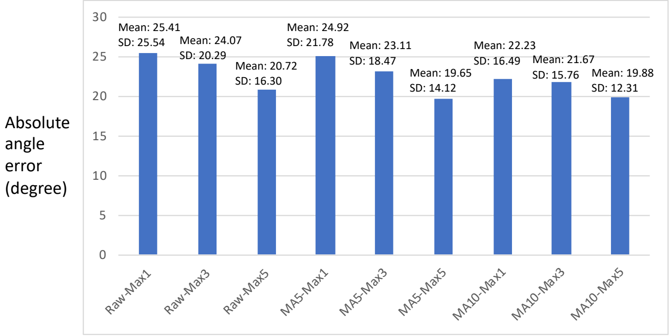


Fig. 4. Absolute angle error

To evaluate the distance estimation method based on machine learning, the following experiments were conducted. In an experiment, a scanner mobile terminal with the BlueTrk+ mobile app scanned the BLE packets from a broadcaster mobile terminal at different sampling distances: 1 m, 2 m, 3 m, 4 m, and 5 m. At each sampling distance, measurements were taken for a certain period of time (approximately two minutes). With the broadcasting interval setting to 100 ms, there were around 900 RSSI measurements received at each sampling distance. The collected data were then grouped for every 30 RSSI measurements. In other words, one set of training data was prepared by 30 RSSI measurements for machine learning. The aforementioned experiment was repeated twice (e.g., to check for data consistency) and the aggregated data were used for machine learning. Overall, the training is based on close to 10,000 RSSI measurements. As mentioned in the previous section, the following training features were used:

- Mean RSSI value (Mean)
- Top 5 RSSI values of five-point moving average (5MA-T5),
- Bottom 5 RSSI values of five-point moving average (5MA-B5
- Top 5 RSSI values of ten-point moving average (10MA-T5),
- Bottom 5 RSSI values of ten-point moving average (10MA-B5)
- Raw RSSI value (i.e., each RSSI measurement) (Raw)

As an illustrative example, Fig. 5 shows real RSSI samples/measurements for 1 m, 3 m and 5 m (i.e., the detected object was 1 m, 3 m and 5 m from the receiver). It can be seen that in some cases the RSSI samples for 1 m, 3 m and 5 m overlap significantly (see R1), making distance estimation difficult. For 1 m and 3 m, while the mean RSSI values are close, the peak values show a wider difference. For example, for 1 m, the RSSI can reach certain peak values (see R2) (i.e., not reachable by the RSSI samples for 3 m). Similarly, for 5 m, the RSSI can fall to very low values more consistently (see R3). Overall, the use of moving averages can smooth out the RSSI values (see the dotted lines).

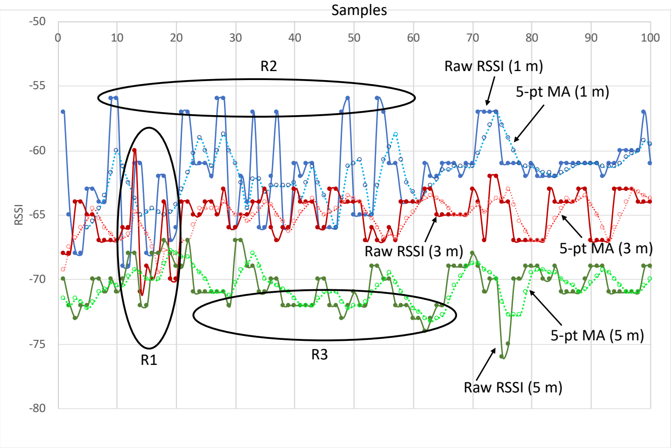


Fig. 5. RSSI samples/measurements

For evaluation, the aforementioned experiment was then repeated three times to collect the testing data. The Orange data analytics tool was used for data processing and different machine learning models were employed and compared, namely ANN (three hidden layers with 50 neurons), kNN, RF and SVM. The distance estimation accuracy was evaluated based on root mean squared error (RMSE).

Fig. 6 shows the comparison results. The experiments confirm the practicality of the proposed methods. Note that the performance depends on various factors (e.g., environmental factors). Our focus is to compare the aforementioned methods under the same conditions. Overall, it can be seen that 10MA-T5 provides the best performance in terms of RMSE. That means, by smoothing out the RSSI values with a moving average and using the top RSSI values as features, better distance estimation accuracy can be achieved.

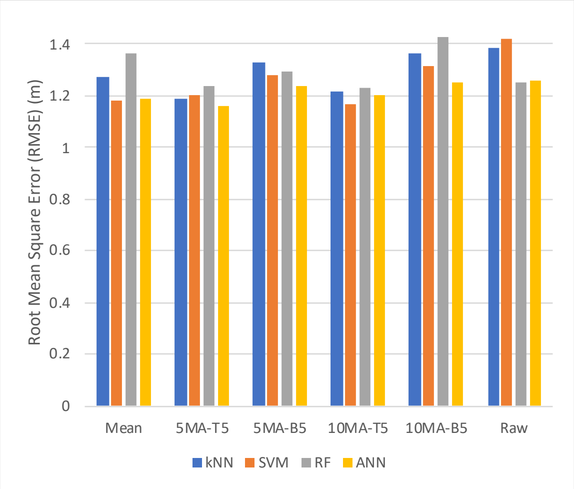


Fig. 6. Comparison of RMSE

V. Conclusion

In conclusion, Bluetooth and BLE can be extended or enhanced to support military-related or DIL environments. BlueTrk+ is a lightweight BLE-oriented application for tracking presence and position securely, flexibly and adaptively. Different levels of security can be provided to support the tracking of presence (i.e., based on the hardware of logical IDs). Inspired by radar, a polar coordinate-based approach can be used to track position in an ad-hoc manner. Machine learning can be employed to facilitate distance estimation. The experimental results should provide valuable insights into the development of BlueTrk+ or related tracking applications.

Acknowledgment

This work was supported by the Department of Computing, The Hong Kong Polytechnic University under account ZVQ5.

References

[1] A. Sikora, M. Krzysztoń and M. Marks, "Application of Bluetooth low energy protocol for communication in mobile networks," *2018 International Conference on Military Communications and Information Systems (ICMCIS)*, Warsaw, Poland, 2018, pp. 1-6.

[2] W. Budiharto, A. A. S. Gunawan, E. Irwansyah and J. S. Suroso, "Android-based wireless controller for military robot using Bluetooth technology," *2019 2nd World Symposium on Communication Engineering (WSCE)*, Nagoya, Japan, 2019, pp. 215-219.

[3] B. P. Aniruddha Prabhu and S. Hebbal, "Small unarmed robot for defense and security: a cost-effective approach using Arduino uno," *2017 2nd International Conference On Emerging Computation and Information Technologies (ICECIT)*, Tumakuru, 2017, pp. 1-6.

[4] R. Abhishek, S. Caroline and A. D. Jose Raju, "IoT driven defence vehicle system," *2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC)*, Nagercoil, India, 2019, pp. 1-4.

[5] N. K. Singh and D. O. Ricke, "Towards an open data framework for body sensor networks supporting Bluetooth Low Energy," *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, San Francisco, CA, USA, 2016, pp. 396-401.

[6] C. Bisdikian, "An overview of the Bluetooth wireless technology," *IEEE Communications Magazine*, vol. 39, no. 12, pp. 86-94, Dec. 2001.

[7] C. Gomez, J. Oller, J. Paradells, "Overview and evaluation of Bluetooth Low Energy: An emerging low-power wireless technology," *Sensors*, vol. 12, pp. 11734–11753, 2012.

[8] Zhang Y., Weng J., Dey R., Fu X, "Bluetooth Low Energy (BLE) security and privacy," *Encyclopedia of Wireless Networks*, Springer 2019.

[9] Y. Gu, A. Lo and I. Niemegeers, “A survey of indoor positioning systems for wireless personal networks,” *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 13-32, First Quarter 2009.

[10] K. Lam, C. Cheung and W. Lee, “RSSI-based LoRa localization systems for large-scale indoor and outdoor environments,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 11778-11791, Dec. 2019.

Yik Him Ho received his Bachelor of Arts in Computing (First Class Honors) from The Hong Kong Polytechnic University in 2014. He is currently a PhD candidate in the Department of Computing, The Hong Kong Polytechnic University. His research interests include Bluetooth Low Energy and cloud computing. He has also received several local and regional IEEE awards, including third prize in the IEEE Hong Kong Section 2014 (UG) Student Paper Contest, first prize in the IEEE 2015 Region 10 Undergraduate Student Paper Competition, and honorary mention in the 2016 IEEE ComSoc Student Competition “Communications Technology Changing the World”.

Yun Fei Liu is an undergraduate student in the Department of Computing at The Hong Kong Polytechnic University. He received first prize in the 36th Chinese Physics Olympiad in 2019.

Caiqi Zhang is an undergraduate student in the Department of Computing at The Hong Kong Polytechnic University. He has received a number of scholarships, such as HKSAR Government Scholarship 2020/21, The Hong Kong Polytechnic University Scholarship 2019/20 and Tellhow Group Scholarship 2018/19. He was also a recipient of the 2020 Institution of Engineering and Technology (IET) Prize (Hong Kong) for outstanding academic performance.

Henry C. B. Chan received his B.A. and M.A. degrees from the University of Cambridge, and his Ph.D. from the University of British Columbia. He is currently an associate professor and associate head of the Department of Computing, The Hong Kong Polytechnic University (PolyU). His research interests include networking/communications, Internet technologies, and computing education. He has conducted various research projects and co-authored research papers published in a variety of journals. He was the Chair (2012) of the IEEE Hong Kong Section and the Chair (2008-2009) of the IEEE Hong Kong Section Computer Society Chapter. He was the recipient of 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.