Sensors and Materials, Vol. 38, No. 9 (2026) 5091–5120
MYU Tokyo

S & M 4612

Origin–Destination Estimation for Fixed-route Buses via MAC Address Carry-over in Randomized Bluetooth Low Energy Advertising Packets

Taito Yoshimura,[1] Kotaro Hayashi,[1] Ismail Arai,[2] and Yuki Matsuda[2,3*]

[1]Graduate School of Environmental, Life, Natural Science and Technology,
Okayama University, Okayama 700-0082, Japan
[2]Graduate School of Science and Technology, Nara Institute of Science and Technology, Nara 630-0192, Japan
[3]Faculty of Environmental, Life, Natural Science and Technology, Okayama University, Okayama 700-0082, Japan

(Received March 31, 2026; accepted June 29, 2026)

Keywords: origin-destination estimation, people flow estimation, Bluetooth Low Energy (BLE), public transportation

In recent years, regional bus services have been facing serious challenges such as a decline in ridership and a shortage of drivers. To address these issues and develop efficient operation plans, the collection of origin–destination (OD) data is essential. However, conventional OD data collection methods suffer from several problems, including labor and time constraints, missing data for flat-rate and cash users when using IC card records, and privacy concerns associated with face recognition cameras. To overcome these limitations, this study focuses on the use of Bluetooth Low Energy (BLE) advertising packets emitted by passengers' mobile devices. A major challenge, however, is that modern mobile devices frequently and randomly change their Media Access Control (MAC) addresses, making continuous device tracking throughout the entire ride difficult. To address this, we propose an OD estimation method that enables continuous device tracking by applying a MAC address carry-over technique, which leverages the asynchronous timing of MAC address changes across nearby devices and information contained in BLE packets. To evaluate the effectiveness of the proposed method, field experiments were conducted on two bus routes with different characteristics. The results show that the proposed method achieved an average precision of 0.458 and an average recall of 0.407. Compared with our previous work, the average recall increased from 0.347 to 0.407, whereas the average precision decreased from 0.702 to 0.458 and the F1-score decreased from 0.465 to 0.431. These results indicate that the proposed method can detect OD records that were missed by the prior method, but it also increases false positives, revealing a trade-off between recall and precision.

[*]Corresponding author: e-mail: yukimat@okayama-u.ac.jp
https://doi.org/10.18494/SAM6358

ISSN 0914-4935

1. Introduction

According to the document of the Ministry of Land, Infrastructure, Transport and Tourism (MLIT), Current Status and Issues Surrounding Regional Public Transportation and Perspectives for Consideration,(1) regional bus services face problems such as declining ridership and driver shortages. Given these circumstances, bus operators need to achieve efficient operations with limited labor and vehicle resources. To realize efficient operations, it is essential to understand the actual usage patterns of buses. Origin-destination (OD) data, which aggregates information on boarding bus stops (origin) and alighting bus stops (destination) for each passenger, is effective for understanding usage patterns. By utilizing OD data, it becomes possible to increase service frequency on sections with high ridership and reduce frequency on sections with low ridership. Additionally, for sections with many transfer passengers, convenience and profitability can be improved by adding direct routes that eliminate the need for transfers.

Research on bus passenger behavior analysis includes various approaches in addition to OD data estimation, for example, passenger counting estimation using image analysis,(2) passenger counting estimation using infrared sensors,(3) and crowding estimation using Bluetooth Low Energy (BLE) advertising packets.(4,5) However, while these methods can capture passenger conditions at specific locations and times, they have the limitation that they cannot obtain complete passenger movement patterns.

Traditionally, OD data collection methods have primarily included card distribution to passengers by surveyors, visual recording by drivers or onboard surveyors, and methods using IC cards.(6–9) However, data collection through card distribution or visual observation has the challenge that the feasible number of days is limited by labor cost constraints, resulting in only limited-period data. When collecting data over long periods and across multiple routes, human costs become a significant problem. Additionally, while IC card-based methods can capture OD data from usage history, they have the constraint that passengers using flat-rate sections or cash payments are not included in the data. Camera-based methods also exist, but they face challenges such as installation location constraints and social acceptability concerns because they create a sense of surveillance, making them difficult to implement easily. Given these challenges, there is a need to establish methods of obtaining OD data at low cost while considering privacy concerns.

Therefore, the aim of this research is to establish a method for estimating OD data for fixed-route buses by scanning BLE advertising packets emitted from electronic devices, such as smartphones that people carry, and using the information obtained from these packets. We propose a method for estimating OD data for fixed-route buses using BLE advertising packets. Specifically, BLE sensing devices are installed inside buses to collect BLE advertising packets from devices carried by passengers and estimate OD data by tracking the Media Access Control (MAC) addresses of the packets. However, in recent years, a challenge that MAC addresses are changed to irregular and random values for privacy protection has arisen. To address this issue, we propose an OD data estimation method that focuses on the asynchronous nature of MAC address changes, which is the fact that nearby devices update their randomized MAC addresses independently at different times rather than simultaneously, and on packet payloads to link randomized MAC addresses as belonging to the same device. Since simultaneous address changes by multiple nearby devices are unlikely within a short time window, this property can be used to narrow down candidate MAC addresses after an address change. This method enables continuous tracking even when MAC addresses change during bus rides. In our prior work,[10] we proposed a BLE-based OD data estimation method for fixed-route buses. The method collected BLE advertising packets inside a bus, applied address carry-over to continuously track randomized MAC addresses, and estimated boarding and alighting stops from the first and last observation times of each tracked MAC address sequence. However, the OD estimation process in the prior work used a fixed time-window threshold for boarding and alighting detection. Specifically, a MAC address sequence was treated as a boarding or alighting event only when its first or last observation time fell within a ±30 s time window around the recorded arrival or departure time of a bus stop. Therefore, when a MAC address changed immediately before alighting, or when the observation time corresponding to the actual boarding or alighting event fell outside the predefined time window, MAC address sequences likely corresponding to actual passenger devices could be excluded from the estimation target, leading to reduced recall. To address this limitation, this study revises the OD estimation logic after address carry-over and eliminates the ±30 s time-window-based boarding and alighting detection used in the prior work. In the proposed method, MAC address sequences remaining after initial filtering and address carry-over are first regarded as passenger device candidate sequences. For each candidate sequence, the stops closest in time to its first and last observation times are estimated as the boarding and alighting stops, respectively. If the estimated boarding and alighting stops are identical, the sequence is regarded as a non-passenger device sequence and excluded, because it is considered to span less than one stop-to-stop section. By estimating boarding and alighting stops on the basis of temporal proximity, the proposed method reduces the possibility that passenger device sequences are excluded solely because their observation times fall outside a fixed time window, thereby increasing the possibility of detecting OD records that were missed by the prior method. The aim of this study is therefore to analyze the causes of failure in our prior work, clarify the relationship between the threshold-based OD estimation logic and recall reduction, and evaluate how the revised OD estimation method affects estimation accuracy. The experimental results show that the proposed method increased the average recall from 0.347 to 0.407 compared with our prior work, whereas the average precision decreased from 0.702 to 0.458 and the F1-score decreased from 0.465 to 0.431. These results indicate a trade-off between increasing recall and maintaining precision.

2. Related Work

In this section, we describe related research on OD data estimation for fixed-route buses and the tracking of devices with randomized MAC addresses.

2.1 Camera-based OD data estimation methods

Camera-based methods exist for OD data estimation.[11,12] Takao *et al.* proposed an OD data estimation system for bus users using intelligent video analysis and applied the estimated information to Synerex, a demand–supply information exchange platform.[11] Their system analyzes the video captured inside a bus to detect passengers' boarding and alighting stops and estimate bus congestion and OD data. Zhang *et al.* estimated OD data by installing cameras at entry and exit doors to identify boarding and alighting passengers and match boarding passengers with alighting passengers.[12] However, camera-based approaches face social acceptability concerns, such as creating a sense of surveillance, making them difficult to implement. Additionally, the accuracy of person detection decreases owing to installation location constraints and high passenger volumes, resulting in the reduced accuracy of OD data acquisition.

2.2 IC card-based OD data estimation methods

IC card-based OD data analysis is effective for railways because passengers need to tap their cards upon both entry and exit. However, for transportation systems like buses that operate with flat-rate fares (such as tapping only when boarding or only when alighting), it is difficult to apply similar methods.[13] Although methods to address this problem have been studied,[7–9] their general applicability has not been fully validated.

2.3 OD data estimation methods using packets transmitted by smartphones

OD data can be estimated by other methods using Wi-Fi and/or BLE packets transmitted by smartphones.[14–19] Ryu *et al.* and Nitti *et al.* estimated OD data using only Wi-Fi packets.[14,18] Kawashima *et al.* estimated OD data using BLE advertising packets.[16] Pu *et al.* and Dunlap *et al.* estimated OD data using Wi-Fi and Bluetooth packets.[15,19] In those studies, sensing devices were installed inside buses to obtain MAC addresses from Wi-Fi or Bluetooth packets transmitted by smartphones. MAC addresses associated with non-passengers were filtered out on the basis of their occurrence counts and average received signal strength indicator (RSSI). Finally, the appearance and disappearance times of MAC addresses were identified and OD data were estimated by matching those times with bus stop departure and arrival schedules. Xu *et al.* estimated OD data using BLE advertising packets and GPS data from fixed-route buses.[17] However, current smartphones usually adopt algorithms for randomizing MAC addresses for privacy protection. Since MAC addresses are occasionally changed at random times, these methods have become difficult to implement by simply tracking MAC addresses. Additionally, the frequency of transmitting Wi-Fi probe request packets is significantly lower than that of BLE advertising packets. Paradeda *et al.* experimentally demonstrated that the first detection of 86% of nearby devices took up to 40 s, with an average of 80 s required for a second detection.[20] This detection latency is a factor that makes OD data estimation using Wi-Fi packets challenging. Furthermore, although MAC address randomization is also applied to Wi-Fi probe requests, randomization in Wi-Fi probe requests is more difficult to address than that in BLE advertising packets, as Wi-Fi probe requests lack payload identifiers that can be used for device tracking.

2.4 Methods of tracking devices with randomized MAC addresses

To overcome problems caused by the randomization of MAC addresses, methods for associating these addresses in order to track unique devices have been proposed in several studies. Kawashima *et al.* tracked devices with randomized MAC addresses and estimated OD data using BLE advertising packets from Exposure Notification, a feature developed by Google and Apple.(16) Exposure Notification is a technology that was developed to mitigate the spread of COVID-19, in which Bluetooth was used to anonymously track contact with other app users. To track devices with randomized MAC addresses, they used Rolling Proximity Identifiers and RSSI contained in Exposure Notification to track BLE devices of bus passengers and estimate OD data. This method requires bus passengers to have installed applications that utilize Exposure Notification, such as COVID-19 Contact-Confirming Application (COCOA),(21) but COCOA is currently unavailable since the functionality was discontinued in November 2022 owing to changes in COVID-19 handling policies. Nitti *et al.* proposed iABACUS, which tracks MAC addresses using invariant fields of Wi-Fi probe requests and reception times to estimate bus boarding and alighting OD data.(18) The evaluation experiments were conducted only in an environment that simulated bus routes using regular cars, and the effectiveness in real-world environments remains unverified. Becker *et al.* proposed an address carry-over algorithm for BLE advertising packets.(22) Their method exploits intra-device asynchrony between random MAC address updates and identifying-token updates in the payload. Here, an identifying token refers to a bit sequence in the payload that is sufficiently device-specific and remains stable for a certain period. If the same identifying token remains after a MAC address change, an observer can link the old and new MAC addresses as belonging to the same device. Therefore, Becker *et al.*'s method can be regarded as a tracking attack that depends on the persistence of device-specific payload tokens across MAC address changes. However, in the devices and OS versions examined in this study, such device-specific payload tokens were not persistently reused across MAC address changes. Therefore, for the devices examined in this study, deterministic tracking based on the same token matching used by Becker *et al.* is difficult.

2.5 Position of this study

Table 1 shows the characteristics of existing OD data collection methods discussed above. As shown in the table, each OD data collection method has distinct limitations. Manual surveys are costly and constrained by labor availability. Camera-based methods raise social acceptability concerns due to privacy issues. IC card-based methods cannot cover passengers who pay flat-rate fares or pay with cash. Wi-Fi and BLE-based methods that transmit smartphone packets offer a low-cost and privacy-conscious alternative, but many do not address MAC address randomization, which has become a standard feature in current smartphones. The only BLE-based method that addressed MAC address randomization relied on COCOA utilizing Exposure Notification; however, COCOA was discontinued in November 2022 and is no longer available. Table 1 lists the authors' prior work and this work as separate entries, and the existing OD data collection methods. Both methods are low-cost, app-free, privacy-conscious, and are designed to

Table 1
OD data collection methods.

	Low cost	App-free	MAC randomization support	Privacy
Manual survey	✗	✓	N/A	✓
IC card	Partial	✓	N/A	✓
Camera[11,12]	✗	✓	N/A	✗
Wi-Fi/Wi-Fi + BLE[14,15,18,19]	✓	✓	✗	✓
BLE (no MAC Rand. support)[17]	✓	✓	✗	✓
BLE (Exposure Notification)[16]	✓	✗	✓	✓
BLE (our prior work)[10]	✓	✓	✓	✓
BLE (this work)	✓	✓	✓	✓

handle MAC address randomization. However, they differ in the OD estimation logic applied after address carry-over. In the authors' prior work,[10] boarding and alighting events were detected using a ±30 s time window based on the recorded arrival or departure time of each bus stop. In contrast, this work eliminates the fixed time-window-based boarding and alighting detection. Instead, it estimates the bus stops temporally closest to the first and last observation times of each tracked device sequence as the boarding and alighting stops, respectively.

Therefore, in this study, the authors aim to establish a BLE-based OD estimation method that handles MAC address randomization without requiring passengers to install a specific application or rely on an external service. Unlike Becker *et al.*, the proposed method does not use persistent device-specific payload tokens. Instead, it uses target packet labels observed in BLE advertising packets, namely, 0x004c10 for Apple devices and 0xfef3 for Android devices, to restrict the analysis to comparable BLE packets. These labels indicate packet categories and are not unique identifiers for individual devices. The proposed method exploits the fact that MAC address changes are not synchronized across nearby devices. When a MAC address disappears, the method searches for a newly appearing MAC address with the same packet label and similar RSSI within a short time window, and treats it as a candidate for the same device. Thus, the proposed carry-over process is not a tracking attack based on persistent device-specific tokens, but a short-term association method for OD estimation in the limited environment of a bus. We build upon our prior work,[10] which faced limitations in low recall rates and failed to detect some passenger-related OD records. We present further analysis of the OD data estimation logic, propose a revised method, and evaluate it in detail.

3. Preliminary Investigation

In this section, BLE advertising packets for OD data estimation using BLE devices with randomized MAC addresses are investigated. First, we analyze the information contained in the payload of advertising packets and examine identifiers that can be used for device tracking. Next, we conduct an investigation using an RF-shielded box to determine which types of devices are transmitting advertising packets and whether the identifiers examined above are included in the advertising packets.

3.1 BLE advertising packet specifications

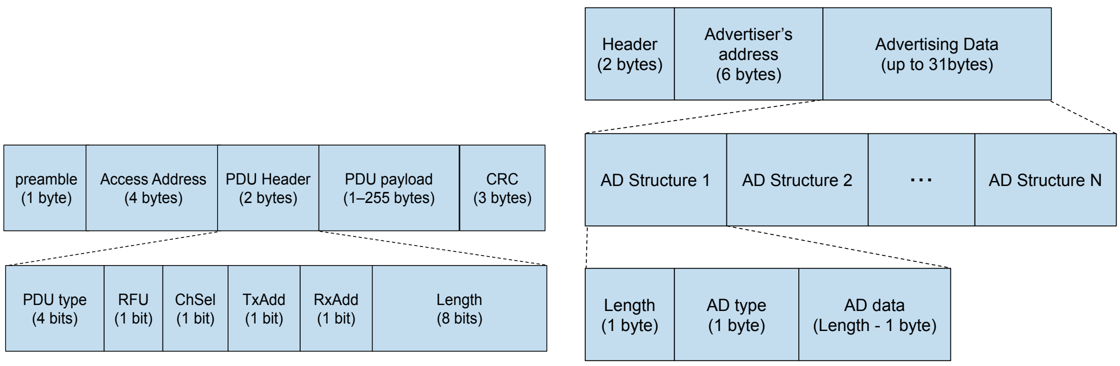
We explain the structure and detailed content of BLE advertising packets based on Bluetooth Core Specification.(23) First, the format of advertising packets is shown in Fig. 1(a). It includes a preamble, access address, PDU header, PDU payload, and CRC.

There are two types of advertising: advertising directed to specific devices and advertising directed to unspecified multiple devices. These two types have different PDU payload formats.

The PDU payload of advertising packets directed to specific devices includes an Advertiser's address and Target Address. The format of the PDU payload for advertising packets directed to unspecified multiple devices is shown in Fig. 1(b). As shown in the figure, it includes an Advertiser's address and Advertising Data (AD). In this paper, AD refers to the entire Advertising Data field, which contains one or more AD Structures.

Finally, we explain the structure of AD. As shown in Fig. 1(b), each AD Structure consists of a Length field, an AD type field, and an AD data field. The AD type indicates the type of information contained in the AD data field, and the interpretation of the AD data field depends on the AD type. Table 2 shows examples of major AD types.

As shown in Table 2, the AD data contained in BLE advertising packets includes identifiers that could potentially serve as effective identifiers for tracking BLE devices. We actually collected BLE advertising packets to investigate what types of devices are transmitting BLE advertising packets, examine the duration of MAC addresses, and verify whether AD data can serve as effective identifiers for tracking MAC addresses.



(a) (b)

Fig. 1. (Color online) Structure of BLE advertising packets: (a) BLE advertising packet format and (b) PDU payload of advertising packets to an unspecified number of devices.(23)

Table 2
Examples of major AD types.

AD type	Name	Content
0x16	Service Data – 16-bit UUID	The first value contains a 16-bit service UUID. The remainder contains additional service data.
0xff	Manufacturer Specific Data (MSD)	The first 2 bytes contain a Manufacturer ID issued by Bluetooth SIG to companies followed by arbitrary-length binary data.

3.2 Investigation of BLE advertising behavior using RF-shielded box

To investigate what types of devices are transmitting BLE advertising packets and examine the duration of MAC addresses, we conducted a survey by placing several types of devices in an RF-shielded box.

3.2.1 Experimental device setup

As the RF-shielded box, we used MICRONIX MY1510,[24] which provides approximately 70 dB of shielding performance in the 2.4 GHz band and is equipped with RF absorbers (MYA-75) providing more than 20 dB of attenuation.

As the BLE advertising packet collection device (hereafter referred to as the BLE scanner), we used Raspberry Pi 4 Model B[25] equipped with a Bluetooth 4.0+EDR/LE Class1 compatible USB adapter (BUFFALO BSBT4D100BK),[26] as shown in Fig. 2. The LTE module shown in Fig. 2 was used for time synchronization during field experiments and was not attached during preliminary experiments. Although Raspberry Pi 4 Model B has a built-in Bluetooth communication module, we used an external Bluetooth communication module to ensure consistent specifications.

For advertising packet collection, we used the Python library bluepy.[27] BLE scanning is the process by which a BLE scanner detects nearby BLE devices by receiving advertising packets transmitted by them. There are two types of scanning methods: passive scanning and active scanning. In passive scanning, the BLE scanner is set to receive advertising packets and collects them without additional interaction. In active scanning, the BLE scanner, after receiving advertising packets, sends scan requests to the advertising devices to obtain additional information.

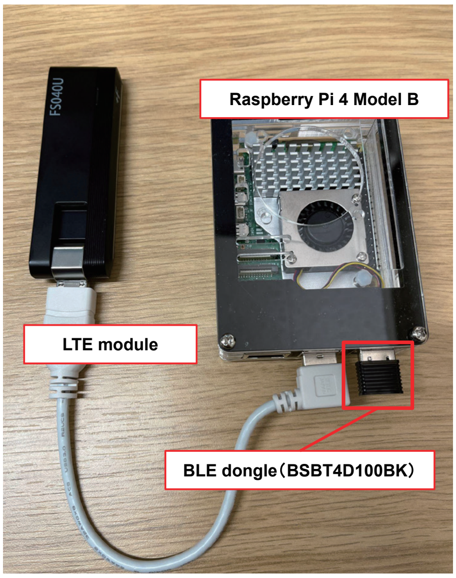


Fig. 2. (Color online) BLE sensing device.

For the bluepy used in this research, both methods are feasible (although this is not explicitly stated in the official documentation, the scanning mode can be selected by setting the passive argument of the scan() function to True for passive scanning or False for active scanning, with False being the default), but we adopted the active scanning method for the experiments in this study. We chose this scanning method because it allows the reception of scan responses, which increases the number of observable events and reduces the impact of packet misses.

3.2.2 Analysis of devices transmitting BLE advertising packets

We investigated what types of devices transmit BLE advertising packets. We selected iPhones, Apple Watches, a MacBook, and Android smartphones as investigation targets, as shown in Table 3, and the details of each device are described below.

The iPhones investigated are the five devices shown as A in Table 3. All devices had Bluetooth enabled and were placed in the RF-shielded box with the BLE scanner running and all devices in sleep mode. Among them, iPhone A-2, which was brand new and had only been signed in to an Apple ID with no user-installed apps, did not emit any BLE advertising packets during the observation period. The other investigated iPhones transmitted one to three types of advertising packets. We confirmed that iPhones include Manufacturer Specific Data (MSD) in the AD data field. The first 2 bytes contain a company identifier called the manufacturer ID, which was 0x004c in the acquired data. This was registered as Apple in the Bluetooth SIG Assigned Numbers document.[28] According to reverse engineering studies of the BLE advertising packets of Apple products,[29,30] the MSD format of Apple products is as shown in Fig. 3, and we confirmed that the data we actually acquired followed the same format.

We conducted similar investigations for the two Apple Watches and one MacBook shown as B and C in Table 3. The investigated MacBook did not transmit advertising packets, while the two Apple Watches transmitted advertising packets with MSD beginning with 0x004c10.

Table 3
Devices used in preliminary investigation.

	(A) iPhone	(B) Apple Watch	(C) MacBook	(D) Android Smartphone
1	13, iOS 16.2	Series 5, watchOS 10.3	Pro, macOS 14.5	Google Pixel 3a, OS 10
2	13 mini, iOS 15.3.1	Series 9, watchOS 10.3		Google Pixel 4, OS 10
3	15 Pro Max, iOS 17.0			Google Pixel 4a, OS 13
4	SE2, iOS 16.1.1			Google Pixel 6 Pro, OS 12
5	XR, iOS 17.5.1			Google Pixel 6 Pro, OS 14
6				AQUOS sense3, OS 11
7				Oppo A55s 5G, OS 11
8				ROG Phone 5s, OS 13

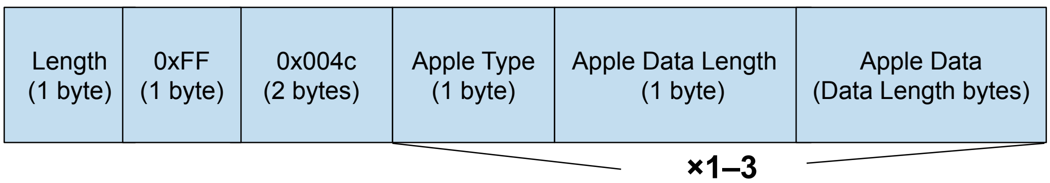


Fig. 3. (Color online) Format of iPhone Manufacturer Specific Data (MSD).

We conducted similar investigations for the eight Android smartphones shown as D in Table 3. All devices had Bluetooth enabled and were placed in the RF-shielded box with the BLE scanner running and all devices in sleep mode. The investigation revealed that four out of eight Android smartphones (D-1, D-2, D-4, D-7) did not transmit advertising packets. The conditions for packet transmission are currently unknown. The advertising packets transmitted by the remaining four Android smartphones included Service Data – 16-bit UUID in the AD data field, with the first 2 bytes being 0xfef3. This was registered as a Google LLC service in the Bluetooth SIG Assigned Numbers document.(28)

3.2.3 Analysis of MAC address randomization timing

For the analysis of MAC address duration, the experimental target device was Apple iPhone 15 Pro Max with iOS 17.0. The experiment was conducted in a laboratory setting, and to block radio waves from devices other than the target smartphone, we placed one target smartphone and one BLE scanner in an RF-shielded box. We collected advertising packets for approximately 20 h in the RF-shielded box. The iPhone used as the experimental target transmitted two types of advertising packets with MSD beginning with 0x004c10 and 0x004c12. The results of examining the address duration for each packet are shown in Fig. 4. The blue graph represents MAC addresses with MSD beginning with 0x004c10, and the orange graph represents those with 0x004c12. The statistical values for MAC address duration are shown in Table 4.

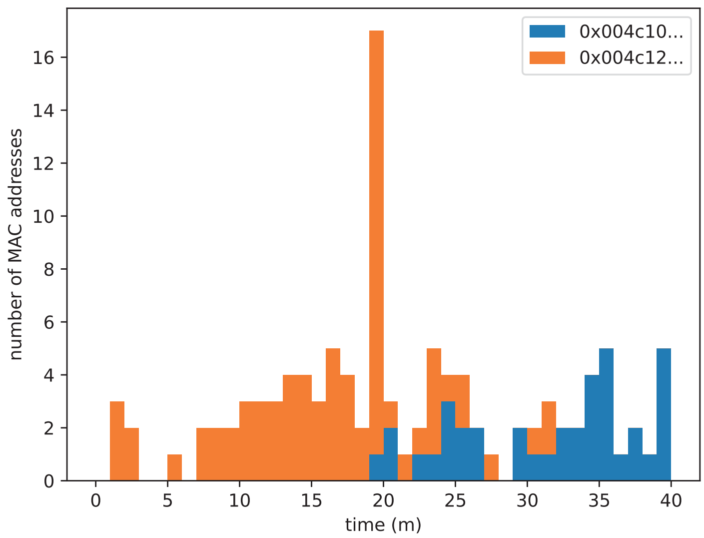


Fig. 4. (Color online) Histogram of MAC address durations.

Table 4
MAC address duration statistics.

MSD	AVG.	MAX	MIN
0x004c10	31 min 37 s	39 min 32 s	19 min 58 s
0x004c12	16 min 12 s	31 min 51 s	1 min 6 s

The experimental results revealed that among the packets transmitted by iPhones and Apple Watches, packets with MSD beginning with 0x004c10 have a longer average MAC address duration than do packets beginning with 0x004c12. Additionally, we found that, for both iPhones and Android smartphones, the AD data fields associated with MSD and Service Data – 16-bit UUID were updated simultaneously with MAC address changes.

Furthermore, we confirmed that these 0x004c10 packets are continuously advertised by the examined Apple devices, except for the new device A-2, unless Bluetooth is manually disabled. From these findings, we conclude that the continuous tracking of Apple devices is possible by targeting the MAC addresses of these specific 0x004c10 packets. For Android smartphones, we conclude that device tracking is possible by using MAC addresses from packets whose Service Data – 16-bit UUID field begins with 0xfef3.

4. Proposed Method

According to the preliminary investigation results in Sect. 3 and the authors' prior work,[10] we propose an OD data estimation method utilizing BLE advertising packets. To address the limitations observed in the prior work, we made the following modifications to the method: Steps 1–3 are identical to those in the prior work and remain unchanged. Step 4 (Secondary Filtering) is newly introduced in this paper. This step was not present in the prior work. Its purpose is to reduce false detections of external devices that persist even after address carry-over. The details are described in Step 4 below. In Step 5 (OD Estimation), an approach different from the prior work is adopted. In the prior work, a time-window threshold (±30 s from the recorded arrival and departure times at each bus stop) was used to determine whether a MAC address corresponded to a boarding or alighting event. Events falling within this window were classified as boarding or alighting events. In the proposed method, this threshold is eliminated in favor of a different approach. The details are described below in Step 5. An overview of the proposed method is shown in Fig. 5, and the details of each step are described below.

(1) Packet Collection: We placed two BLE scanners, identical to the devices used in the preliminary experiments, at the front and rear of the bus (one at each position) to capture BLE advertising packets. During the scanning process, we recorded the scan time, RSSI, random MAC addresses, and AD data. Since the recorded MAC addresses are random addresses, they are unlikely to lead to permanent personal identification, and the privacy invasion risk is considered relatively low. However, even with random addresses, there is a risk that they could lead to indirect personal identification when combined with other information. Privacy considerations are discussed in detail in Sect. 6.7.

(2) Initial Filtering: First, since we use packets with MSD beginning with 0x004c10 for tracking iPhone devices and packets whose Service Data – 16-bit UUID field begins with 0xfef3 for Android devices, we extract only packets matching these criteria. Next, assuming that passenger devices are observed by both front and rear BLE scanners, we retain only the MAC addresses detected by both BLE scanners. Finally, since external devices are also detected during travel, we set an occurrence count threshold and exclude MAC addresses with total occurrence counts below the threshold from both BLE scanners as external devices.

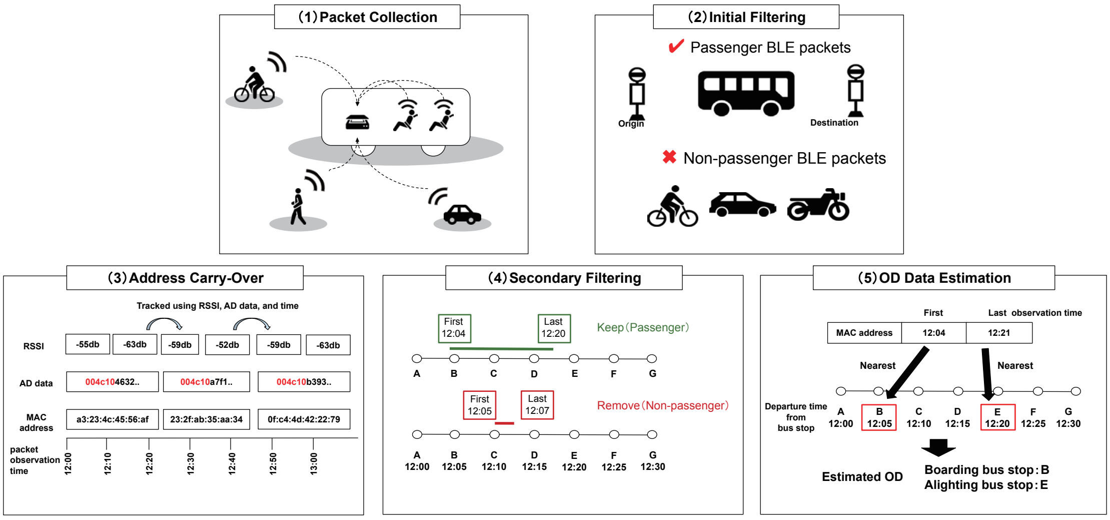


Fig. 5. (Color online) Overview of the proposed method.

(3) Address Carry-over: Figure 6 shows an overview of candidate extraction in the address carry-over process. Address carry-over links different randomized MAC addresses that are presumed to belong to the same device. When a tracked MAC address is no longer observed, the system calculates the time difference (Δt) between its last observation time (t_{last_old}) and the first observation time of each newly appearing MAC address (t_{first_new}), and compares the difference (Δt) with the address carry-over threshold (T_{co}). If the time difference is less than or equal to the threshold ($\Delta t \leq T_{co}$) and the newly appearing MAC address is associated with the same target packet label, the MAC address is regarded as a candidate for the changed address. Here, a candidate refers to a MAC address newly observed within the threshold after the disappearance of the tracked MAC address. Specifically, candidates are selected from packets with MSD beginning with 0x004c10 for iPhones and from packets whose Service Data – 16-bit UUID field begins with 0xfef3 for Android smartphones. If no candidate is found, the system terminates the tracking of the current MAC address sequence. If multiple candidates are found, the system compares the average RSSI of the tracked MAC address sequence with that of each candidate sequence. The candidate with the smallest difference in average RSSI is then linked to the tracked sequence as the changed MAC address of the same device. This process enables the continuous tracking of a device even when its randomized MAC address changes during the bus ride.

(4) Secondary Filtering: The MAC address sequences obtained after carry-over may include those from devices outside the bus. Because the BLE scanners are installed on the bus, the appearance and disappearance times of a MAC address sequence are associated with the bus position along the route. In this study, this bus position is approximated using the recorded arrival and departure times at bus stops. For each MAC address sequence, we identify the bus stop whose recorded arrival or departure time is closest to its appearance time and the one closest to its disappearance time. Here, the closest bus stop is determined on the basis of the

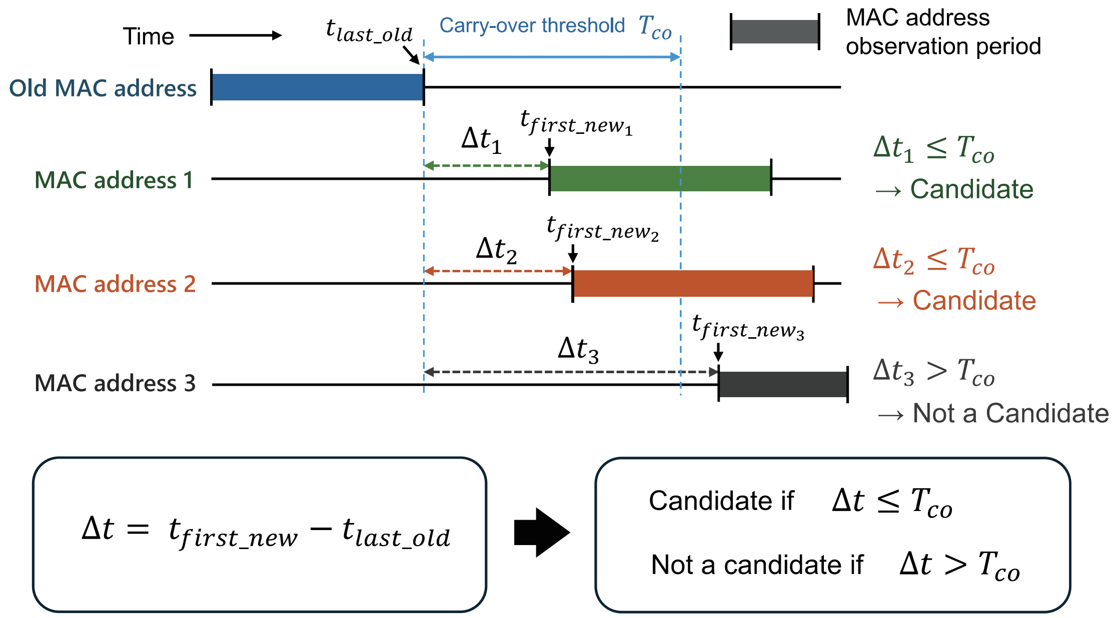


Fig. 6. (Color online) Candidate extraction based on the address carry-over threshold.

recorded arrival and departure times of the bus, rather than by directly measuring the geographical distance between the bus position and each bus stop. If these two bus stops are the same, i.e., the sequence spans less than one stop-to-stop section, we consider the sequence to originate from a non-passenger device and exclude it.

(5) OD Estimation: Using the results from Step 4, which provide the appearance and disappearance times for MAC addresses presumed to belong to the same device, we proceed to estimate the boarding and alighting stops. For each device, the boarding stop is inferred as the bus stop temporally closest to its first detection, and the alighting stop is inferred as the one closest to its last detection.

5. Field Experiments and Evaluation

5.1 Data collection routes

We selected two fixed-route buses operating in Okayama City as data collection targets. The selected route map is shown in Fig. 7 with the red line (National Hospital Line) and blue line (Industrial Technology Center Line) indicating the bus routes.

These routes were selected for two reasons. First, both routes have relatively long scheduled travel times of over 30 min and a large number of bus stops. The proposed OD estimation method using BLE advertising packets requires detecting changes in the randomized MAC addresses transmitted by passengers' devices. Routes with sufficiently long travel durations tend to result in longer passenger ride times, increasing the likelihood of MAC address changes occurring during a single trip, making them suitable for validating the proposed method.

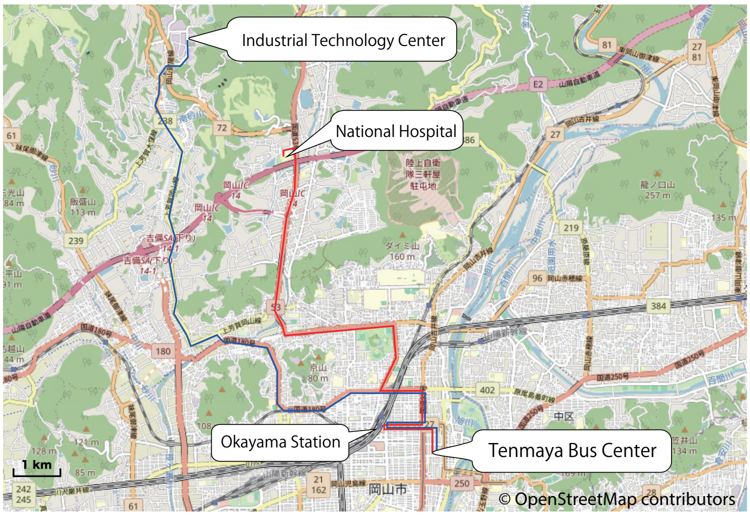


Fig. 7. (Color online) Route map of the experimental area.

Furthermore, the high number of stops is expected to yield diverse OD patterns, which allows for a more comprehensive evaluation of the method's performance across various scenarios.

Second, we aim to evaluate the proposed method under different traffic conditions. The National Hospital Line passes through central Okayama City with heavy traffic, while the Industrial Technology Center Line includes suburban residential areas and industrial parks with lighter traffic in some sections. These considerations led to our decision that these two lines are appropriate for validating the effectiveness of the proposed method.

The experiments were conducted with the cooperation of Chutetsu Bus Co., Ltd., on October 15 and December 18, 2024. This study was approved by the Ethical Review Committee for Research Involving Human Subjects at Nara Institute of Science and Technology (Approval No.: 2020-I16). The list below shows the actual datasets collected during the evaluation.

- National Hospital Line
  First stop: Tenmaya Bus Center/Final stop: National Hospital
  (1) October 15, 8:25–9:08 (43 min)
  (2) October 15, 14:30–15:11 (41 min)
  (3) October 15, 17:37–18:31 (54 min)
- Industrial Technology Center Line (outward trip)
  First stop: Tenmaya Bus Center/Final stop: Industrial Technology Center
  (1) December 18, 7:33–8:29 (56 min)
  (2) December 18, 11:10–11:51 (41 min) (Final stop: Sayama Danchi)
  (3) December 18, 14:30–15:18 (48 min)
  (4) December 18, 17:20–18:11 (51 min) (Final stop: Sayama Danchi)
- Industrial Technology Center Line (return trip)
  First stop: Industrial Technology Center/Final stop: Tenmaya Bus Center
  (1) December 18, 8:46–9:38 (52 min)
  (2) December 18, 11:58–12:36 (38 min) (First stop: Sayama Danchi)
  (3) December 18, 15:31–16:04 (33 min) (Final stop: Okayama Station)

In this study, two BLE scanners were installed inside the bus, one near the front and the other near the rear of the bus, referring to previous BLE sensing studies conducted in real bus environments. Kanamitsu *et al.* placed a BLE sensing device at the rear of a bus and estimated bus congestion using BLE signals.(5) In contrast, Ikenaga *et al.* used an existing bus location system installed near the driver's seat as a BLE scanner and collected BLE signals at the front of the bus.(4) These studies indicate that BLE signals can be collected in real bus environments using scanners placed at either the front or rear of the bus. However, when a scanner is installed at a single position, detecting passenger devices located far from the scanner may be difficult. Therefore, in this study, two BLE scanners were installed near the front and rear of the bus to observe passenger devices throughout the passenger cabin more reliably. Specifically, whenever possible, the BLE scanners were placed on window-side seats at the frontmost and rearmost parts of the passenger cabin, at approximately seat height. However, because the experiments were conducted during regular bus operations with general passengers on board, the same seats could not always be secured in every run. Therefore, although the approximate front–rear placement was maintained, the exact scanner positions differed slightly across runs. Each BLE scanner consisted of a Raspberry Pi with a USB BLE dongle directly attached. No external or directional antenna was used, and the antenna orientation was not intentionally controlled during the field experiments. The scan interval was set to 1 s, and we recorded the collection time, random MAC addresses, RSSI, and AD data of advertising packets. Additionally, at each bus stop, we visually confirmed and recorded the arrival time (when the doors opened) and the departure time (when the doors closed). To monitor BLE scanner operation and ensure time synchronization, we integrated LTE communication modules. Specifically, we connected LTE-compatible USB dongles (PIXELA PIX-MT110 or Fujisoft +F FS040U) to the BLE scanners described in Sect. 3.

5.2 Ground-truth OD data collection method

To create ground-truth OD data for evaluation, we manually recorded the boarding and alighting stops of each passenger. Since we recorded data from general passengers, we could not determine whether their smartphones had Bluetooth turned on.

An example of a handwritten annotation is shown in Fig. 8. First, a seating chart corresponding to each bus stop was prepared. When a passenger boarded at a stop, a circle (○) was drawn at their seat; when a passenger alighted, a cross (×) was drawn. For passengers who remained on board, a hyphen or similar mark was used. In rare cases, especially during crowded times, passengers moved to different seats. These movements were indicated by arrows between seats. If standing passengers were observed owing to congestion, their approximate locations and OD information were written in the margin of the seating chart. By tracking these seat-level annotations, from circles to crosses, accurate OD data were compiled, including both seated and standing passengers. Since passengers typically move only at stops, they remain stationary while the bus is in transit. During travel between stops, the annotations on the seating chart were cross-checked against the actual positions of passengers, ensuring the accuracy of the recorded data.

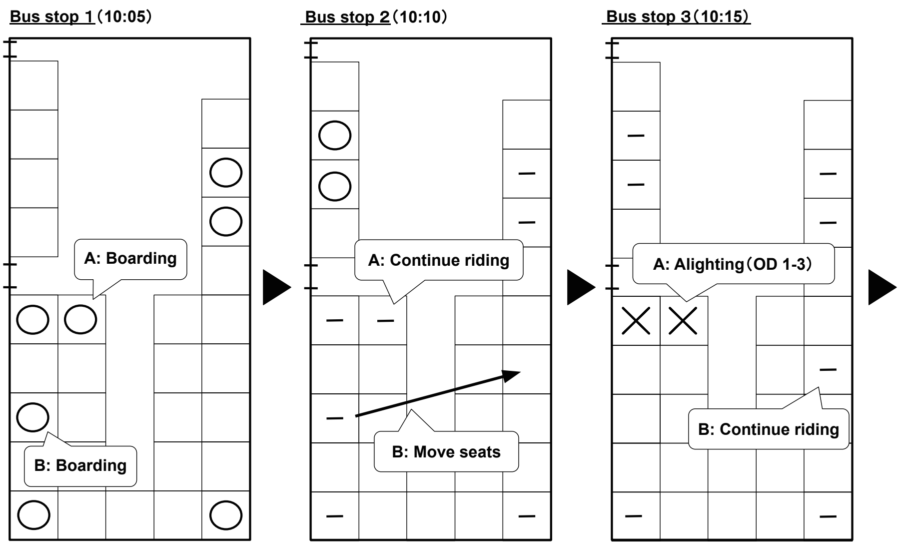


Fig. 8. Example of annotation memo.

5.3 Evaluation method

To evaluate the proposed method, we compared the estimated OD data with the ground-truth OD data. For each bus operation, bus stops that were passed without stopping were excluded from the analysis. The evaluation was based on three standard metrics: recall, precision, and F1-score. Recall represents the proportion of correctly estimated OD data among the ground-truth data [Eq. (1)]. Precision represents the proportion of correct estimations among all OD data estimated by the proposed method [Eq. (2)].

A predicted OD is considered incorrect if either the boarding or alighting stop does not match the ground truth, or if there is no corresponding ground-truth OD entry. In cases where the predicted number of OD matches exceeds the actual number (e.g., if three OD records are estimated for a route segment where only one person actually traveled), only the number matching the ground truth is counted as correct; the rest are considered false positives.

$$Recall = \frac{Number\ of\ Correct\ OD\ Estimations}{Number\ of\ Ground\ Truth\ ODs} \tag{1}$$

$$Precision = \frac{Number\ of\ Correct\ OD\ Estimations}{Number\ of\ Estimated\ ODs} \tag{2}$$

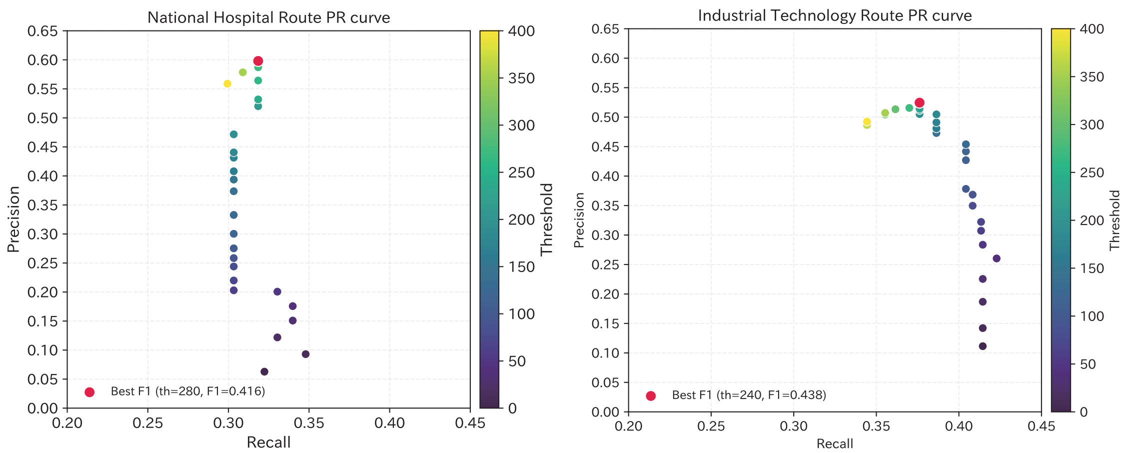
5.4 Data analysis and parameter setting

In data analysis, we assume that devices inside the bus are observable by both front and rear BLE scanners, so we extract MAC addresses detected by both BLE scanners. For iPhone tracking, we extract MAC addresses with MSD beginning with 0x004c10. For Android smartphone tracking, we extract MAC addresses where the Service Data – 16-bit UUID begins with 0xfef3.

Next, we describe the two parameter settings required for OD data estimation using the proposed method presented in Sect. 4. iPhones and Android smartphones differ in BLE advertising packet transmission intervals and characteristics. Therefore, we configured the parameters appropriately for each device type, and the setting criteria, as described below.

iPhone Packets: iPhone packets can be collected at every scan when the transmission interval is stable at 1 s or less. Additionally, considering the 1 s scan interval and the possibility of missed packets, we set the address carry-over threshold to 2 s. To determine the occurrence count threshold, we used the BLE advertising packets collected from three runs of the National Hospital Line and seven runs of the Industrial Technology Center Line. We evaluated the average recall and precision for each route by varying the occurrence count threshold from 0 to 400. The results for the National Hospital Line are shown in Fig. 9(a). Since the average F1-score was highest at a threshold of 280, we set the occurrence count threshold to 280 for this route. Next, the results for the Industrial Technology Center Line are shown in Fig. 9(b). Since the average F1-score was highest at a threshold of 240, we set the occurrence count threshold to 240 for this route.

Android Smartphone Packets: Android smartphone packets have a lower average RSSI than iPhone packets and are more frequently missed; therefore, we set the address carry-over threshold to 5 or 10 s. Additionally, since Android smartphone packets exhibited very little noise, we set the occurrence count threshold to 20 or 30 counts. The low noise level is likely attributable to the low RSSI of Android smartphone packets, which may prevent the BLE scanners from detecting external devices. However, many aspects of Android smartphone packet behavior remain unclear, and the appropriate parameter values varied across different runs, making it impossible to determine universally optimal settings.



(a) (b)

Fig. 9. (Color online) Occurrence count thresholds for the (a) National Hospital Line and (b) Industrial Technology Center Line.

6. Discussion

6.1 Evaluation results

Table 5 shows the evaluation results for the prior research method,[10] and Table 6 shows the results for the proposed method. Asterisks indicate improvements over the prior research method. The prior research method achieved an average precision of 0.702 (maximum 1.000, minimum 0.533) and an average recall of 0.347 (maximum 0.591, minimum 0.158) when evaluated on two different bus routes in Okayama City. The proposed method achieved an average precision of 0.458 (maximum 0.591, minimum 0.364) and an average recall of 0.407 (maximum 0.591, minimum 0.158). In terms of the F1-score, the proposed method achieved an average F1-score of 0.431, which was lower than the 0.465 achieved by the prior method.

The recall for iPhone and Android was calculated on the basis of all passengers, rather than only those known to carry each respective device type. This is because it was unknown whether individual passengers carried an iPhone or an Android smartphone. Consequently, neither value

Table 5
Evaluation results for each run (prior research method).

	Operating time	Total no. of passengers	iPhone		Android		iPhone and Android	
			Precision	Recall	Precision	Recall	Precision	Recall
National Hospital Line	8:25–9:08 (43 m)	22	0.900	0.455	0.750	0.136	0.867	0.591
	14:30–15:11 (41 m)	19	0.755	0.158	–	–	0.755	0.158
	17:37–18:31 (54 m)	35	0.818	0.257	1.000	0.057	0.846	0.314
Industrial Technology Center Line	7:33–8:29 (56 m)	35	0.625	0.286	0.500	0.029	0.611	0.314
	8:46–9:38 (52 m)	23	0.538	0.304	1.000	0.087	0.533	0.348
	11:10–11:51 (41 m)	13	1.000	0.231	–	–	1.000	0.231
	11:58–12:36 (38 m)	17	0.556	0.294	0.500	0.118	0.538	0.412
	14:30–15:18 (48 m)	14	0.667	0.286	1.000	0.071	0.571	0.286
	15:31–16:04 (33 m)	8	0.667	0.500	–	–	0.667	0.500
	17:20–18:11 (51 m)	28	0.667	0.286	1.000	0.107	0.643	0.321
Average	–	–	0.719	0.305	0.821	0.086	0.702	0.347

Table 6
Evaluation results for each run (proposed method).

	Operating time	Total no. of passengers	iPhone		Android		iPhone and Android	
			Precision	Recall	Precision	Recall	Precision	Recall
National Hospital Line	8:25–9:08 (43 m)	22	0.588	0.455	0.600	0.136	0.591	0.591
	14:30–15:11 (41 m)	19	0.500	0.158	–	–	0.500	0.158
	17:37–18:31 (54 m)	35	0.706	0.343*	0.333	0.086*	0.577	0.429*
Industrial Technology Center Line	7:33–8:29 (56 m)	35	0.483	0.400*	0.333	0.086*	0.421	0.457*
	8:46–9:38 (52 m)	23	0.500	0.348*	0.400	0.087	0.429	0.391*
	11:10–11:51 (41 m)	13	0.571	0.308*	0.000	0.000	0.400	0.308*
	11:58–12:36 (38 m)	17	0.471	0.471*	0.500	0.176*	0.391	0.529*
	14:30–15:18 (48 m)	14	0.600	0.214	0.500	0.071	0.429	0.214
	15:31–16:04 (33 m)	8	0.400	0.500	0.000	0.000	0.364	0.500
	17:20–18:11 (51 m)	28	0.647	0.393*	0.333	0.143*	0.483	0.500*
Average	–	–	0.546	0.358*	0.333	0.087*	0.458	0.407*

*Asterisks indicate improvements over the results of the prior research method shown in Table 5.

is expected to reach 1.000. For example, the iPhone recall would reach 1.000 only if all passengers carried iPhones. In that case, the Android recall would be expected to be 0.

Table 6 also shows that recall varied depending on the operating time. In particular, the runs departing around 14:30 showed low recall on both routes. For the iPhone and Android combined results, recall values were 0.158 for the National Hospital Line departing at 14:30 and 0.214 for the Industrial Technology Center Line departing at 14:30. In contrast, the corresponding precision values were 0.500 and 0.429, respectively, which were comparable to those of the other runs. One possible reason for this tendency is a difference in the proportion of passengers carrying BLE-detectable devices. Recall was calculated using all passengers in the ground truth as the denominator. Therefore, if many passengers did not carry smartphones, had Bluetooth disabled, or did not transmit the target BLE advertising packets, the number of detectable passenger devices would decrease, leading to lower recall. In contrast, precision was calculated only from the OD records estimated by the proposed method; thus, passengers without BLE-detectable devices would not directly decrease precision. During manual annotation, the runs departing around 14:30 appeared to include more elderly passengers than the morning commuting and evening return periods, suggesting that the proportion of passengers carrying BLE-detectable devices may have been relatively low during these runs. However, because passengers' smartphone ownership, Bluetooth settings, and age groups were not directly recorded, this interpretation remains a possible explanation, and further investigation with a larger number of runs is necessary to verify this tendency.

A preliminary experiment and field observations suggest several possible factors contributing to the lower recall for Android smartphones. First, the RSSI of Android BLE advertising packets has been reported to be lower than that of iPhones (approximately −40 dBm for iPhone vs −60 dBm for Android),[31] increasing the likelihood of packet capture failures due to insufficient signal strength. Second, Android devices have longer BLE advertising intervals than iPhones,[31] further reducing the number of packets. Third, not all Android devices continuously broadcast 0xfef3 packets; in the preliminary experiment, approximately half of the observed Android devices did not transmit 0xfef3 packets at all. Fourth, there is a possibility that fewer passengers on the bus carried Android smartphones to begin with. Verifying whether Android smartphone users were underrepresented among bus passengers would require a follow-up study on the smartphone ownership distribution among bus users.

In addition to the overall comparison with the prior method, we evaluated the effect of the newly introduced Step 4 by comparing the proposed method with and without secondary filtering, as shown in Tables 6 and 7. Table 7 shows the results obtained without Step 4. Compared with the results with Step 4 in Table 6, the average precision for iPhone and Android combined decreased from 0.458 to 0.400 when Step 4 was not applied, whereas the average recall remained unchanged at 0.407. Similarly, the average recall values for iPhone and Android individually were unchanged, while precision decreased, especially for Android devices. In this analysis, additional false negatives caused by Step 4 were defined as cases in which true positive OD estimates obtained without Step 4 would be removed by Step 4. We confirmed that no such cases occurred, indicating that Step 4 reduced false positives without introducing additional false negatives in this evaluation.

Table 7
Evaluation results for each run (proposed method without Step 4).

	Operating time	Total no. of passengers	iPhone		Android		iPhone and Android	
			Precision	Recall	Precision	Recall	Precision	Recall
National Hospital Line	8:25–9:08 (43 m)	22	0.588	0.455	0.375*	0.136	0.520*	0.591
	14:30–15:11 (41 m)	19	0.500	0.158	–	–	0.500	0.158
	17:37–18:31 (54 m)	35	0.706	0.343	0.300*	0.086	0.556*	0.429
Industrial Technology Center Line	7:33–8:29 (56 m)	35	0.452*	0.400	0.231*	0.086	0.364*	0.457
	8:46–9:38 (52 m)	23	0.500	0.348	0.286*	0.087	0.391*	0.391
	11:10–11:51 (41 m)	13	0.571	0.308	0.000	0.000	0.364*	0.308
	11:58–12:36 (38 m)	17	0.471	0.471	0.250*	0.176	0.310*	0.529
	14:30–15:18 (48 m)	14	0.600	0.214	0.167*	0.071	0.273*	0.214
	15:31–16:04 (33 m)	8	0.400	0.500	0.000	0.000	0.333*	0.500
	17:20–18:11 (51 m)	28	0.647	0.393	0.211*	0.143	0.389*	0.500
Average	–	–	0.543*	0.358	0.202*	0.087	0.400*	0.407

*Asterisks indicate decreases compared with Table 6.

The MAC address sequences removed by Step 4 were those for which the bus stops closest to the appearance and disappearance times were the same, meaning that the sequences spanned less than one stop-to-stop section. Such short-duration sequences are likely to correspond to devices outside the bus, such as pedestrians near bus stops or passengers in nearby vehicles. However, passenger devices may also be observed for only a short duration when packet losses occur or when scanner placement is not optimal. Therefore, further evaluation under different route environments and scanner placements is necessary to verify the robustness of Step 4.

6.2 Discussion of prior research method

On the basis of the results shown in Sect. 6.1, we conducted a detailed analysis of two runs: the 17:37 departure on the National Hospital Line (hereafter Run A) and the 7:33 departure on the Industrial Technology Center Line (hereafter Run B). These runs had the highest passenger counts on their respective routes and yielded intermediate values for both precision and recall, making them suitable for examining both the effectiveness and limitations of the prior research method. For Run A, the prior research method estimated a total of 13 passengers, of which 11 were correctly estimated. For Run B, the prior research method estimated a total of 18 passengers, of which 11 were correctly estimated.

We visualized the data for iPhone devices after applying the address carry-over process. Figure 10 shows the results for Run A and Fig. 11 shows the results for Run B. The x-axis represents time, while the tick labels indicate the corresponding bus stop numbers. The y-axis shows different devices. Horizontally aligned entries are MAC addresses presumed to belong to the same device. Different line colors indicate different MAC addresses. Red dashed lines represent bus stop arrival times and blue dashed lines represent departure times. By analyzing these figures, we explored potential improvements in recall and precision. Labels indicating successful estimations or failure causes are placed to the left of each trace in Figs. 10 and 11. Below, we detail each label and its implications.

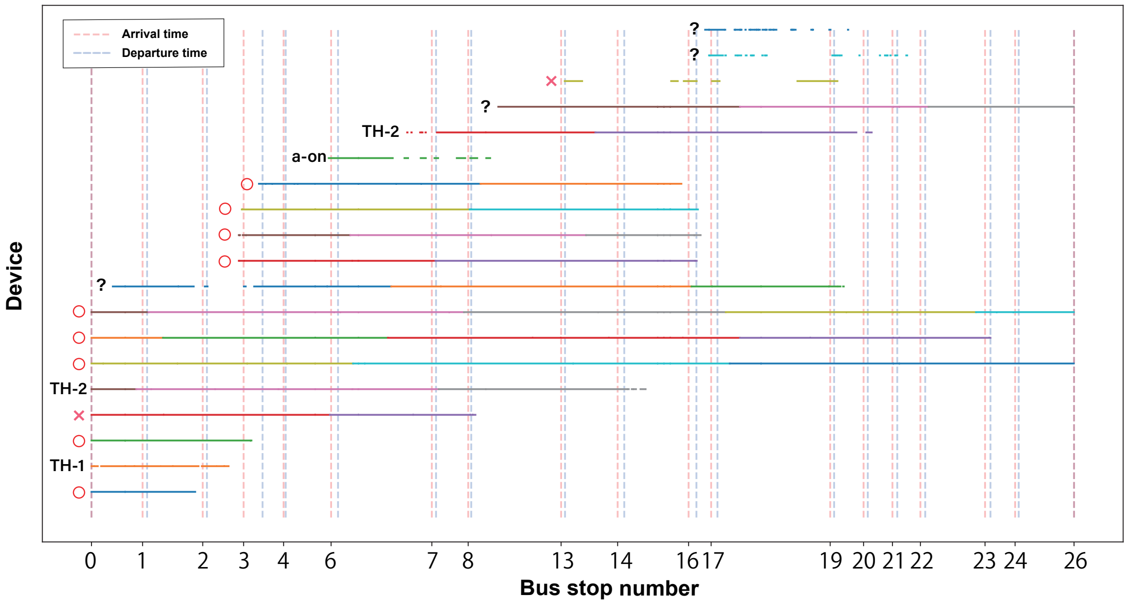


Fig. 10. (Color online) Graph after address carry-over (iPhone) (National Hospital Line, departing at 17:37, 54 min).

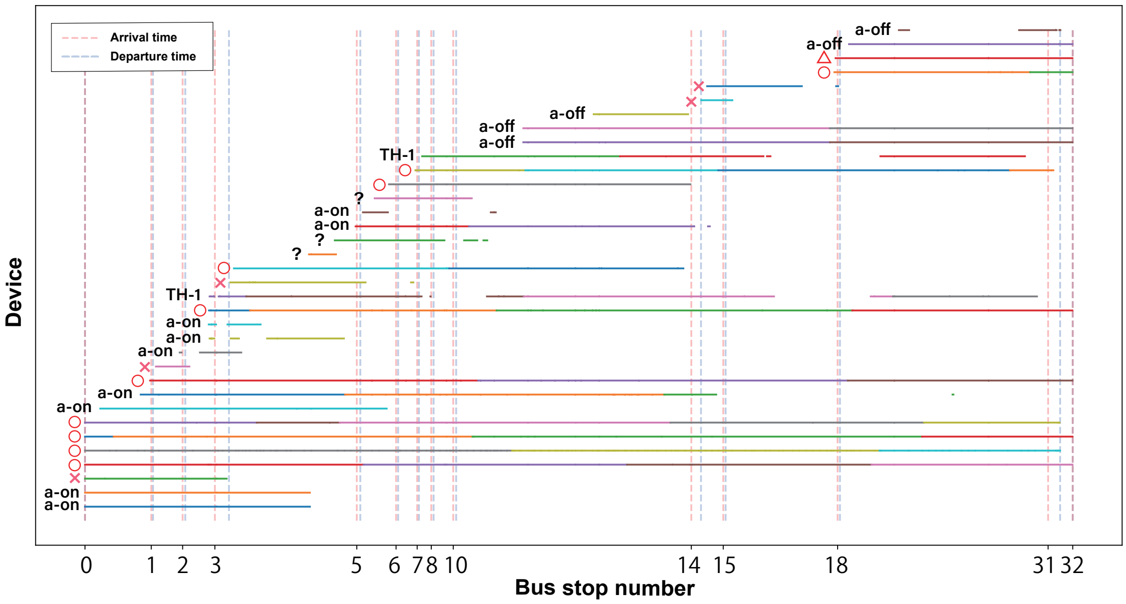


Fig. 11. (Color online) Graph after address carry-over (iPhone) (Industrial Technology Center Line, departing at 7:33, 56 min).

- Red circles (○) represent correctly estimated OD pairs.
- Crosses (×) represent incorrect estimations.
- Triangles (△) indicate overestimations due to users carrying multiple devices.
- "TH-1" represents failures due to MAC address changes right before alighting. The new MAC address appears only briefly and is filtered out by the occurrence count threshold. If the threshold is very low, noise increases (lower precision); if very high, valid but short-lived MAC addresses are missed (lower recall).
- "TH-2" indicates cases where MAC addresses are filtered out because their appearance or disappearance falls outside the 30 s boarding or alighting detection window. Expanding the window increases recall but risks false positives, lowering precision. Shrinking the window has the opposite effect: precision improves, but recall may drop, especially when users remain near the bus after alighting.
- "a-on" indicates that the MAC address appeared near a boarding stop where a known passenger boarded, but no corresponding alighting passenger was observed around the time it disappeared.
- "a-off" denotes the converse case of a-on: the MAC address disappears near a valid alighting stop, but no boarding passenger is matched to its appearance time.
- "?" is data that appears to be from a passenger but cannot be confirmed, as no boarding or alighting passengers were observed near its appearance and disappearance times.

Also, the failure cases labeled "a-on" and "a-off" (one-sided matches) suggest that the observation periods of many MAC address sequences partially overlap the actual riding periods of passengers but lack full OD correspondence. This may be due to noise or incomplete tracking due to address randomization.

6.3 Comparison with prior research method

The prior research method used threshold-based boarding and alighting detection.(10) However, as shown by cases TH-1 and TH-2 in Figs. 10 and 11, this approach sometimes excluded devices that likely belonged to actual passengers, resulting in reduced recall. To improve recall, in this paper, we propose a different approach that does not use thresholds for boarding and alighting detection. Instead, all MAC address sequences remaining after the filtering process are treated as passenger-device candidates. The boarding stop is estimated as the stop closest to the first observation time of the MAC address, and the alighting stop as the stop closest to the last observation time.

The ground-truth and estimated OD data for Run A are shown in Fig. 12(a). The proposed method estimated a total of 26 passengers, of which 15 were correctly estimated. The ground-truth and estimated OD data for Run B are shown in Fig. 12(b). The proposed method estimated a total of 38 passengers, of which 16 were correctly estimated. The results are shown in Table 6. The proposed method achieved an average precision of 0.458 and an average recall of 0.407. Compared with the prior method,(10) the average recall increased from 0.347 to 0.407, whereas the average precision decreased from 0.702 to 0.458, a decrease of approximately 0.244. Consequently, the F1-score of the proposed method was 0.431, which was lower than that of the

Alighting bus stop / Boarding bus stop	2		3		6		7		8		13		14		16		17		19		20		21		22		23		24		26	
	T	E	T	E	T	E	T	E	T	E	T	E	T	E	T	E	T	E	T	E	T	E	T	E	T	E	T	E	T	E	T	E
0 Bus stop A	1	1	3	2	0	0	1	0	0	1	1	0	1	1	0	0	2	0	0	1	1	0	0	0	1	1	1	1	1	0	2	2
1 Bus stop B	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2 Bus stop C			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3 Bus stop D					0	2	0	0	1	0	0	0	0	0	5	4	1	0	2	1	0	0	1	0	3	0	0	0	0	0	0	0
6 Bus stop E							0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
7 Bus stop F									0	0	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
8 Bus stop G											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
13 Bus stop H													0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
16 Bus stop I																	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

(a)

Alighting bus stop / Boarding bus stop	3		5		6		7		10		14		15		18		31		32	
	T	E	T	E	T	E	T	E	T	E	T	E	T	E	T	E	T	E	T	E
0 Bus stop A	0	1	0	3	0	1	0	0	0	0	1	0	0	0	0	0	3	2	5	2
1 Bus stop B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
2 Bus stop C	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
3 Bus stop D			0	1	0	0	0	1	0	0	7	1	1	0	0	0	2	1	2	1
5 Bus stop E					0	1	0	0	0	2	2	1	0	0	0	0	1	1	0	0
6 Bus stop F							0	0	0	0	1	1	0	0	0	0	1	0	0	0
7 Bus stop G									0	0	0	0	0	0	0	0	1	2	0	0
8 Bus stop H									0	0	0	1	0	0	0	0	1	1	1	0
10 Bus stop I											0	1	0	0	0	0	0	0	1	2
14 Bus stop J													0	0	0	2	0	0	0	0
15 Bus stop K															0	0	0	0	0	0
18 Bus stop L																	1	2	1	3
31 Bus stop M																			0	1
32 Bus stop N																				

(b)

Fig. 12. (Color online) (a) Proposed method: OD estimation results (T: Ground-truth OD, E: Estimated OD) (Run A: National Hospital Line, departing at 17:37, 54 min). (b) Proposed method: OD estimation results (T: Ground-truth OD, E: Estimated OD) (Run B: Industrial Technology Center Line, departing at 7:33, 56 min).

prior method (0.465). These findings suggest that the prior research method maintains relatively high precision at the expense of recall,[10] indicating a trade-off between recall and precision. While the proposed method may be effective in environments with good packet observation conditions, it faces challenges in real-world settings owing to the high risk of noise contamination from over-detection.

Future work should focus on improving recall while maintaining the high precision of the prior research method. This can be achieved by implementing mechanisms to dynamically adjust thresholds on the basis of observation conditions and bus operating circumstances, as well as incorporating RSSI-based boarding and alighting detection methods.

6.4 Route-section coverage analysis

In the recall and precision metrics based on exact OD matching, a predicted OD is considered incorrect if either its boarding or alighting stop does not match the ground truth. However, even a predicted OD that is not an exact match may still capture part of the passenger's actual travel section. Such partial matches can provide useful information for understanding section-level travel demand. Therefore, we conducted a supplementary analysis using route-section coverage to examine what proportion of the ground-truth travel sections was covered by the estimated OD data.

Each OD pair was represented as a set of consecutive stop-to-stop sections between the boarding and alighting stops. Let g be a ground-truth OD pair and e be an estimated OD pair, and let S_g and S_e be their corresponding sets of stop-to-stop sections. Here, one section denotes the link between two adjacent bus stops; thus, route-section coverage is based on the number of overlapping stop-to-stop sections, not on physical distance or travel time.

In this analysis, the estimated and ground-truth OD pairs that exactly matched in both boarding and alighting stops were matched first. For the remaining OD pairs, one-to-one matching was then performed to maximize the total number of overlapping sections. The ground-truth OD pairs with no corresponding estimated OD pair were assigned zero overlapping sections. The estimated OD pairs not used in the one-to-one matching were excluded from the numerator of route-section coverage. The final route-section coverage was calculated by dividing the total number of overlapping sections in the matched OD pairs by the total number of sections in all ground-truth OD pairs, as follows:

$$\textit{Route-section coverage} = \frac{\sum_{(g,e)\in M} \left| S_g \cap S_e \right|}{\sum_{g\in G} \left| S_g \right|}, \tag{3}$$

where G is the set of all ground-truth OD pairs, M is the set of matched pairs between ground-truth and estimated OD pairs, and S_g and S_e denote the sets of stop-to-stop sections corresponding to a ground-truth OD pair g and an estimated OD pair e, respectively.

Table 8 shows the route-section coverage for each run. Across all 10 runs, 2676 of the 3669 stop-to-stop sections counted over the ground-truth OD pairs overlapped with the estimated OD data, yielding a route-section coverage of 0.729. Thus, when the OD pairs not counted as correct under exact matching were also considered, approximately 73% of the actual travel sections were covered by the estimated OD data. Although the proposed method does not always identify every passenger's exact boarding and alighting stops, it may still be useful for understanding passenger flows at the section level.

Coverage also varied across runs. It was high for the Industrial Technology Center Line from 11:58 to 12:36, at 0.946, but low for the National Hospital Line from 14:30 to 15:11, at 0.356. The latter run also had low recall under exact OD matching, which is consistent with the existing evaluation results. As discussed in Sect. 6.1, this low value may be related to differences in the

Table 8
Route-section coverage for each run.

Route	Operating time	Route-section coverage	Overlapping/ total no. of ground-truth sections
National Hospital Line	8:25–9:08 (43 m)	0.762	250 / 328
	14:30–15:11 (41 m)	0.356	99 / 278
	17:37–18:31 (54 m)	0.620	298 / 481
Industrial Technology Center Line	7:33–8:29 (56 m)	0.705	534 / 757
	8:46–9:38 (52 m)	0.883	371 / 420
	11:10–11:51 (41 m)	0.647	132 / 204
	11:58–12:36 (38 m)	0.946	296 / 313
	14:30–15:18 (48 m)	0.657	161 / 245
	15:31–16:04 (33 m)	0.935	172 / 184
	17:20–18:11 (51 m)	0.791	363 / 459
Total	-	0.729	2676 / 3669

proportion of passengers carrying BLE-detectable devices during the 14:30 time period. However, because this study did not directly record each passenger's smartphone ownership, Bluetooth settings, or age group, further investigation is needed to verify this interpretation.

Route-section coverage should be interpreted as a supplementary metric based on post hoc one-to-one matching between the estimated and ground-truth OD data; it does not directly verify correspondences between passenger devices and individual passengers. Moreover, the metric can be high even when an estimated OD pair covers a longer range than the corresponding ground-truth OD pair. Therefore, it should not be interpreted as a strict measure of OD estimation accuracy. Rather, it is an exploratory analysis that supplements exact-match-based evaluation. Future work should examine the validity of section-level coverage as an evaluation metric through controlled riding experiments in which participants' actual OD pairs and device traces can be directly matched.

6.5 Differences between the two routes

As described in Sect. 5.4, for iPhone packets, we set the MAC address occurrence count threshold to 280 for the National Hospital Line and 240 for the Industrial Technology Center Line. These different values reflect the distinct environmental conditions of each route, which affect the number of scanned packets and noise levels. Specifically, the National Hospital Line operates through central urban areas with heavy traffic, resulting in the more frequent scanning of external devices and higher noise levels. In contrast, the Industrial Technology Center Line passes through low-traffic residential areas, resulting in lower noise levels and thus a lower optimal threshold. These findings indicate that the proposed method requires route-specific parameter adjustment.

The evaluation results showed a minimal difference in average recall between the two routes (approximately 0.022), but a notable difference of approximately 0.139 in average precision, with the National Hospital Line achieving higher precision. This route operates through urban areas with heavy traffic, where buses frequently slow down or stop owing to congestion and traffic signals. Under these conditions, passengers' BLE devices are scanned for longer periods inside the bus, increasing MAC address occurrence counts and improving passenger device identification during filtering. Although external device noise is higher, these devices typically have low occurrence counts and low RSSI values, making them easier to exclude. This likely resulted in fewer false detections and higher precision.

As shown above, performance varied depending on route-specific environmental conditions, even across only two routes. However, the degree of variation across routes with different environments and the achievable performance levels remain unclear. In particular, the performance of the proposed method on bus runs with high passenger density has not yet been sufficiently evaluated. In such conditions, many passenger devices may simultaneously transmit BLE advertising packets, which could increase packet collisions or missed scans. In addition, passengers' bodies may attenuate or block BLE signals, making it more difficult for the scanners to continuously observe the same device. Moreover, the parameters of the proposed method must be manually tuned for each route and separately for iPhone and Android devices. For Android devices, optimal values varied across individual runs and could not be uniquely determined. Since only two routes were evaluated, sufficient data for constructing and validating an automatic parameter optimization algorithm were not available. In future work, data from diverse routes should be collected to formalize the relationship between environmental factors, such as traffic density and route length, and optimal parameter values, and the introduction of automated tuning algorithms such as Bayesian optimization should be explored.

6.6 Scanner placement

The placement of BLE scanners is an important factor affecting the reception performance of BLE advertising packets. In this study, as described in Sect. 5.1, two BLE scanners were placed near the front and rear of the passenger cabin to capture signals from passenger devices throughout the bus. However, the scanners were placed on passenger seats and were not fixed to the ceiling or other interior structures. In addition, no external or directional antenna was used. Therefore, radio propagation may have been affected by attenuation and blockage caused by passengers' bodies, seats, and interior structures of the bus. Because the scanners were placed on window-side seats whenever possible, BLE advertising packets from devices outside the bus, such as those carried by pedestrians near bus stops or passengers in nearby vehicles, may also have been received.

Ideally, BLE scanners should be installed at positions where signals from devices outside the bus are less likely to be received while signals from passenger devices inside the bus can be received stably. For example, installing one scanner near the center of the ceiling, where the effect of blockage by passengers' bodies is relatively small, or installing two scanners near the front and rear ceiling areas may be effective. However, preliminary experiments on scanner placement were not conducted in this study. Therefore, evaluating the effects of different scanner placements on the number of received packets, RSSI, the reception of signals from devices outside the bus, and OD estimation accuracy remains an important direction for future work.

6.7 Privacy considerations

The proposed method requires careful consideration of privacy issues. While linking randomized MAC addresses for device tracking poses a potential privacy risk, we consider this risk to be relatively low because tracking is only possible within the limited environment where BLE scanners are deployed. Specifically, when a device leaves the BLE scanner detection range and its random MAC address changes, the device cannot be linked to its previous address upon re-entering the BLE scanner range. Therefore, identifying the same device across multiple visits becomes difficult if its random MAC address changes outside the scanner range; however, cross-visit identification may still be possible if the address remains unchanged. Although privacy risks cannot be completely eliminated, they are considerably lower than those of camera-based methods because devices are treated as distinct entities for each bus ride. Furthermore, hashing MAC addresses during collection can further mitigate these risks. Note that hashing was not applied in the experiments of this study, as the collected data were used solely for research purposes. In this study, OD estimation uses only the header portions of MSD and service data, specifically 0x004c10 and 0xfef3, respectively, and neither the detailed payload contents nor device names are used in the estimation process. However, in practical deployment, there is a possibility that data containing personal information may be collected. For instance, some devices broadcast user-configured names in their BLE advertising packets, which could potentially identify individuals and thus pose a privacy risk. Therefore, payload data beyond the two header fields mentioned above should not be collected or retained.

7. Conclusions

We proposed and evaluated a method for estimating OD data for fixed-route buses using BLE advertising packets. Conventional methods faced challenges such as high monetary costs and tracking difficulties due to MAC address randomization. In contrast, the proposed method leverages the asynchronous timing of MAC address changes and payload information to perform MAC address carry-over, enabling continuous device tracking and OD data estimation for fixed-route buses. Compared with the prior method,[10] the proposed method increased the average recall from 0.347 to 0.407, enabling the detection of OD records missed by the prior method. However, the average precision decreased from 0.702 to 0.458, indicating an increase in false positives. Consequently, the F1-score of the proposed method was 0.431, which was lower than that of the prior method (0.465). These results indicate a trade-off between increasing recall and maintaining precision.

In future work, we aim to optimize the recall–precision trade-off by introducing dynamic threshold adjustment based on route characteristics and operating conditions, as well as RSSI-based boarding and alighting detection logic. We also plan to extend the current deterministic carry-over logic toward probabilistic models and learning-based approaches to improve same-device identification accuracy in dynamic BLE environments.

Acknowledgments

This study was supported in part by JSPS KAKENHI Grant Number JP24K20763, JST PRESTO Grant Number JPMJPR2039, and JST COI-NEXT Grant Number JPMJPF2115. We also thank Chutetsu Bus Co., Ltd. for their invaluable cooperation.

References

1 Ministry of Land, Infrastructure, Transport and Tourism (MLIT) Japan: "Current Status and Issues Surrounding Regional Public Transportation and Perspectives for Consideration" https://www.mlit.go.jp/sogoseisaku/transport/sosei_transport_tk_000211.html (accessed Mar. 26, 2026) (in Japanese).
2 C.-H. Chen, Y.-C. Chang, T.-Y. Chen, and D.-J. Wang: Proc. 8th Int. Conf. Intell. Syst. Des. Appl. **3** (2008) 565–569. https://doi.org/10.1109/ISDA.2008.335
3 A. Olivo, G. Maternini, and B. Barabino: Open Transp. J. **13** (2019) 250. https://doi.org/10.2174/1874447801913010250
4 T. Ikenaga, Y. Matsuda, I. Goto, K. Ueda, H. Suwa, and K. Yasumoto: IEEE Access **13** (2025) 67488. https://doi.org/10.1109/ACCESS.2025.3558988
5 Y. Kanamitsu, E. Taya, K. Tachibana, Y. Nakamura, Y. Matsuda, H. Suwa, and K. Yasumoto: Sensors **22** (2022) 881. https://doi.org/10.3390/s22030881
6 Ministry of Land, Infrastructure, Transport and Tourism (MLIT) Japan: "Encyclopedia of Bus Data Application," https://wwwtb.mlit.go.jp/chubu/tsukuro/library/ (accessed Mar. 26, 2026) (in Japanese).
7 W. Liu, Q. Tan, and L. Liu: Math. Probl. Eng. **2020** (2020) 8305475. https://doi.org/10.1155/2020/8305475
8 I. Lee, S.-H. Cho, K. Kim, S.-Y. Kho, and D.-K. Kim: PLoS One **17** (2022) e0270346. https://doi.org/10.1371/journal.pone.0270346
9 Y. Yuan, Z. Tan, X. Li, and Y. Liu: IEEE Trans. Intell. Transp. Syst. **26** (2025) 4808. https://doi.org/10.1109/TITS.2025.3529290
10 T. Yoshimura, K. Hayashi, I. Arai, and Y. Matsuda: Proc. 15th Int. Conf. Mobile Computing and Ubiquitous Networking (ICMU '25) (2025) 1. https://doi.org/10.23919/ICMU65253.2025.11219133
11 R. Takao, N. Ikeuchi, H. Suzuki, and Y. Matsumoto: Proc. IEEE Int. Conf. Consum. Electron. (ICCE) (2021) 1–4. https://doi.org/10.1109/ICCE50685.2021.9427742
12 C. Zhang, X. Chen, J. Zhao, Z. Jiang, and E. Chung: IEEE Trans. Intell. Transp. Syst. **26** (2025) 8353. https://doi.org/10.1109/TITS.2025.3538999
13 A. B. R. González, J. J. V. Díaz, and M. R. Wilby: IEEE Intell. Transp. Syst. Mag. **14** (2022) 141. https://doi.org/10.1109/MITS.2019.2962149
14 S. Ryu, B. B. Park, and S. El-Tawab: KSCE J. Civ. Eng. **24** (2020) 3092. https://doi.org/10.1007/s12205-020-0316-7
15 Z. Pu, M. Zhu, W. Li, Z. Cui, X. Guo, and Y. Wang: IEEE Internet Things J. **8** (2021) 474. https://doi.org/10.1109/JIOT.2020.3007373
16 M. Kawashima, I. Arai, A. Endo, M. Kakiuchi, and K. Fujikawa: Proc. IEEE Glob. Conf. Artif. Intell. Internet Things (GCAIoT) (2022) 7–12. https://doi.org/10.1109/GCAIoT57150.2022.10019176
17 J. Xu, Z. Pan, C. Zhang, and X. Yang: Sensors **25** (2025) 2351. https://doi.org/10.3390/s25082351
18 M. Nitti, F. Pinna, L. Pintor, V. Pilloni, and B. Barabino: Energies **13** (2020) 1446. https://doi.org/10.3390/en13061446
19 M. Dunlap, Z. Li, K. Henrickson, and Y. Wang: Transp. Res. Rec. **2595** (2016) 11. https://doi.org/10.3141/2595-02
20 D. B. Paradeda, W. K. Junior, and R. C. Carlson: Transportes **27** (2019) 115. https://doi.org/10.14295/transportes.v27i3.2039
21 Digital Agency: "COVID-19 Contact-Confirming Application (COCOA)" https://www.digital.go.jp/policies/cocoa/ (accessed Mar. 10, 2025) (in Japanese).
22 J. K. Becker, D. Li, and D. Starobinski: Proc. Privacy Enhancing Technologies. Issue **3** (2019) 50. https://doi.org/10.2478/popets-2019-0036
23 Bluetooth SIG: "Bluetooth Core Specification Version 5.4" https://www.bluetooth.com/specifications/specs/core-specification-5-4/ (accessed Mar. 26, 2026).

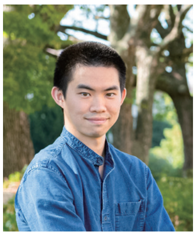
24 MICRONIX: "MY1510" https://micronix-jp.com/products/shield-box/taurus/my1510.html (accessed Sep. 1, 2026).
25 Raspberry Pi Ltd.: "Raspberry Pi 4 Model B" https://www.raspberrypi.com/products/raspberry-pi-4-model-b (accessed Sep. 1, 2026).
26 BUFFALO INC.: "BSBT4D100BK" https://www.buffalo.jp/product/detail/bsbt4d100bk.html (accessed Sep. 1, 2026).
27 I. Harvey: "bluepy" https://github.com/IanHarvey/bluepy (accessed Sep. 1, 2026).
28 Bluetooth SIG: "Assigned Numbers" https://www.bluetooth.com/specifications/assigned-numbers/ (accessed Mar. 26, 2026).
29 J. Martin, D. Alpuche, K. Bodeman, L. Brown, E. Fenske, L. Foppe, T. Mayberry, E. C. Rye, B. Sipes, and S. Teplov: Proc. Privacy Enhancing Technologies. Issue **4** (2019) 34. https://doi.org/10.2478/popets-2019-0057
30 G. Celosia and M. Cunche: Proc. Privacy Enhancing Technologies. Issue **1** (2020) 26. https://doi.org/10.2478/popets-2020-0003
31 A. Yoshihara: Master's Thesis, Graduate School of Science and Technology, Nara Institute of Science and Technology (2025) 1–105. http://hdl.handle.net/10061/0002000914

About the Authors

Taito Yoshimura received his B.E. degree in engineering from Okayama University, Japan, in 2025. He is currently pursuing an M.E. degree in engineering at Okayama University, Japan. His research interests include ubiquitous computing and human-computer interaction. He is a member of IPSJ.

Kotaro Hayashi received his B.E. degree in engineering from Okayama University, Japan, in 2025. He is currently pursuing an M.E. degree in engineering at Okayama University, Japan. His research interests include ubiquitous computing and big data analysis. He is a member of IPSJ.

Ismail Arai received his M.E. and Ph.D. degrees in engineering from the Graduate School of Information Science, Nara Institute of Science and Technology, in 2004 and 2008, respectively. After working as a postdoctoral research fellow at Ritsumeikan University and an assistant/associate professor at the National Institute of Technology, Akashi College, he has been serving as a professor at the Information Initiative Center, Nara Institute of Science and Technology, since 2016. His research interests include intelligent transportation systems, pervasive computing, and cybersecurity. He is also a member of the ACM, IPSJ, and IEICE.



Yuki Matsuda received his M.E. and Ph.D. degrees in engineering from the Graduate School of Information Science, Nara Institute of Science and Technology, in 2016 and 2019, respectively. He was an Assistant Professor at the Graduate School of Science and Technology, Nara Institute of Science and Technology. He has been a Lecturer at the Faculty of Environmental, Life, Natural Science and Technology, Okayama University, since 2024. His research interests include participatory sensing, location-based systems, and wearable computing. He is a member of IPSJ, IEICE, ACM, and IEEE.