

修士論文

バスODデータ自動取得に向けたBLEアドバタイジング・パケットのスマートフォン所持者の行動を考慮したバス車内外分類

吉原 新太

奈良先端科学技術大学院大学
先端科学技術研究科
情報理工学プログラム

主指導教員: 藤川 和利 教授
情報基盤システム学研究室 (情報科学領域)

令和6年1月31日提出

本論文は奈良先端科学技術大学院大学先端科学技術研究科に
修士(工学) 授与の要件として提出した修士論文である。

吉原 新太

審査委員：

藤川 和利 教授	(主指導教員, 情報科学領域)
安本 慶一 教授	(副指導教員, 情報科学領域)
新井 イスマイル 准教授	(副指導教員, 情報科学領域)

バス OD データ自動取得に向けた BLE アドバタイジング・パケットのスマートフォン所持者の行動を考慮したバス車内外分類*

吉原 新太

内容梗概

乗客がどこからどこまで乗車したかを記録したバス OD データの取得自動化は、効率的なバス運行サービスを実現するために重要である。バス OD データの自動取得手法としては、スマートフォンが発する BLE アドバタイジング・パケットを活用して乗客を追跡するアプローチが注目されている。しかし、バス車内には乗客の持つスマートフォンが発する BLE アドバタイジング・パケットと、バス車外のスマートフォンから発せられる BLE アドバタイジング・パケットが混在するため、それらを区別する必要がある。このアプローチでは、ランダムに変化する MAC アドレスを有するスマートフォンを追跡するために、変化後の MAC アドレスを変化前の MAC アドレスに紐付けることが必須となる。既存の BLE アドバタイジング・パケットのバス車内外分類手法としては、受信信号強度 (RSSI) の弱い BLE アドバタイジング・パケットを RSSI の閾値によってバス車外の BLE アドバタイジング・パケットと判断する手法や、送信時間が短い BLE アドバタイジング・パケットを送信時間の閾値によってバス車外の BLE アドバタイジング・パケットと判断する手法がある。しかし、RSSI を用いた手法は車内環境に合わせた閾値の調整が必要になる。本研究では、車内環境に応じた RSSI の閾値の調整をせずに BLE アドバタイジング・パケットのバス車内外分類性能を向上させるために、バス周辺のスマートフォン所持者の行動を考慮した手法の実現を目指す。そのために、バス車内外のスマートフォンの動きを再現し、バス車内外のスマートフォンから発されている BLE アドバタイジング・パケットの RSSI と BLE アドバタイジング・パケットの送信間隔を既存のバス路線で取得したデータと比較した。そこ

*奈良先端科学技術大学院大学 先端科学技術研究科 修士論文, 令和 6 年 1 月 31 日.

で明らかにした送信時間と送信時間間隔のパターンを考慮した閾値を用いた, バス車内外分類を提案する. 評価では, 実際のバスが走行する環境を模した場所でスマートフォンを所持した被験者が動き, スマートフォンが発している BLE アドバタイジング・パケットをスニファ―を使って収集したデータより作成したデータセットを用いて, バス車内外分類実験を実施した. その結果, 提案手法は送信時間を用いた従来手法と比較して車内環境に合わせた閾値の調整なしで F1 スコアの観点から最大 23.3% の分類性能の向上を示した .

キーワード

Bluetooth, MAC アドレス, OD 推定, 屋内測位, 公共交通

Classification of Bus Passengers’ Indoor/Outdoor Status Using BLE Advertising Packets for Automated Bus OD Data Collection: Accounting for Smartphone User Behavior*

Yoshihara Arata

Abstract

Automating the collection of bus origin-destination (OD) data, which records where passengers board and alight, is crucial for achieving efficient bus operation services. An approach that tracks passengers by utilizing BLE advertising packets emitted from smartphones has gained attention as a method for automatically collecting bus OD data. However, BLE advertising packets from passengers’ smartphones inside the bus are mixed with those emitted from smartphones outside the bus, necessitating a method to distinguish between them. This approach requires linking MAC addresses after they change to their previous addresses in order to track smartphones with randomly changing MAC addresses. Existing methods for classifying BLE advertising packets as inside or outside the bus include: interpreting packets with weak Received Signal Strength Indicator (RSSI) as outside the bus based on an RSSI threshold, or interpreting packets with short transmission times as outside the bus based on a transmission time threshold. However, methods using RSSI require threshold adjustments tailored to the in-vehicle environment. This research aims to improve the classification performance of BLE advertising packets without adjusting RSSI thresholds according to the in-vehicle environment by implementing a method that considers the behavior of smartphone users around buses. To achieve this, we reproduced the movement of

*Master’s Thesis, Graduate School of Science and Technology, Nara Institute of Science and Technology, January 31, 2025.

smartphones inside and outside buses and compared the RSSI and transmission intervals of BLE advertising packets from smartphones inside and outside buses with data collected from existing bus routes. We propose a bus interior/exterior classification using thresholds that consider the patterns of transmission time and transmission intervals identified from this analysis. For evaluation, we conducted classification experiments using a dataset created from BLE advertising packets collected via sniffers from smartphones carried by test subjects moving in an environment simulating actual bus operations. Results demonstrate that the proposed method improves classification performance by up to 23.3% in terms of F1 score compared to conventional methods using transmission time, without requiring threshold adjustments for the in-vehicle environment.

Keywords:

Bluetooth, MAC Address, OD estimation, indoor positioning

Contents

1. はじめに	1
2. 関連研究	5
2.1 バス OD データ自動取得手法	5
2.1.1 IC カードを利用したバス OD データ自動取得手法	5
2.1.2 カメラ画像を利用したバス OD データ自動取得手法	6
2.1.3 スマートフォンのパケットを利用したバス OD データ自動取得手法	6
2.1.4 バス OD データ自動取得手法の比較	7
2.2 アドレスキャリーオーバーアルゴリズム	8
2.3 パケット発信源のバス車内外分類手法	10
2.4 バス車内外分類手法の課題	11
3. BLE アドバタイジング・パケットに関する調査	13
3.1 想定するバス OD データ自動取得手法	13
3.1.1 Bluetooth の概要と規格	14
3.1.2 BLE アドバタイジング・パケット	14
3.1.3 OS ごとの AD Structure の技術仕様と扱うデータの詳細	17
3.1.4 iOS の AD Structure のフォーマット	17
3.1.5 Nearby パケットの技術仕様	18
3.1.6 Find My パケットの技術仕様	19
3.2 BLE アドバタイジング・パケットに対するアドレスキャリーオーバーアルゴリズムの適用に関する検証実験	20
3.2.1 実験方法	20
3.2.2 実験環境	22
3.2.3 パケットの送信条件に関する実験	22
3.2.4 BLE アドバタイジング・パケットの MAC アドレスランダム化の間隔の調査のための実験	26
3.2.5 RSSI の距離に応じた減衰に関する検討	29
3.3 既存のバス路線で取得したデータを使った分析	30
3.3.1 既存のバス路線で取得したデータの詳細	30
4. バス車外のスマートフォン所持者の行動を考慮したバス車内外分類手法	40

5. データセット	44
5.1 データセットの概要	44
5.2 データセット収集実験 A	46
5.2.1 実験 A の方法	47
5.2.2 実験 A の環境	49
5.2.3 実験 A で取得したデータの分析	51
5.3 データセット収集実験 B	59
5.3.1 データセットの概要	60
5.3.2 実験の環境	61
5.3.3 実験で発生させるイベントのスケジュール	62
5.3.4 実験で取得したデータの分析	62
6. バス車内外分類手法の評価	69
6.1 評価手法	69
6.2 バス車内外分類の正解率	71
6.3 バス OD データの推定性能の評価	73
7. 考察	78
7.1 車内外スマートフォン再現データセットに対する考察	78
7.2 車内外スマートフォンの分類結果に対する考察	78
7.2.1 車外のパターンが意図していない分類結果だったもの	79
7.2.2 先行研究と比較して有効だったもの	80
7.2.3 実験 B の対象外 MAC アドレスを含めた分類結果の詳細について	80
7.3 スニファーの位置について	83
7.4 バス車内外分類における CRC の利用可能性について	83
7.5 バス OD データ自動取得手法の実験評価に対する考察	85
7.6 今後の展望	87
8. おわりに	89
謝辞	90
参考文献	91

List of Figures

1	BLE のアドバタイズパケットフォーマット	16
2	指向性アドバタイズの PDU ペイロード	17
3	非指向性アドバタイズの PDU ペイロード	17
4	AdvData のフォーマット	17
5	Manufacturer Specific タイプのフォーマット	18
6	Nearby パケットのフォーマット	19
7	Find My パケットのフォーマット	20
8	検証対象のスマートフォンと確認用のスキャナー	22
9	Android スマートフォンから送信されていた BLE アドバタイジ ング・パケットのフォーマット	25
10	Android スマートフォンから送信されていた Service Data の構成 .	25
11	Find My パケット/0xFE3F パケットの MAC アドレスランダム化 の様子	27
12	MAC アドレス数に対する Find My パケットの MAC アドレス変化 タイミングが重複する確率に関するシミュレーション結果	28
13	iPhone・Android スマートフォンの平均 RSSI 比較	29
14	便ごとの MAC アドレスが変化していないパターン	35
15	非常に短い期間のみ出現が確認された BLE アドバタイジ ング・パケット	36
16	一定期間消失している期間がある BLE アドバタイジ ング・パケット	36
17	バス停車期間中のみ出現が確認されたアドバタイジ ング・パケット	37
18	バス停車期間中のみ出現が確認され、急激な RSSI の上昇が確認 されたアドバタイジ ング・パケット	37
19	アドレスキャリーオーバーアルゴリズムで紐付けたアドバタイ ジ ング・パケットの組み	38
20	バス停に滞在する歩行者のイベントの発生方法	45
21	実験 A の実験環境	51
22	実験 A のバス車内を模した部屋内部の俯瞰図	52
23	実験 A: 自動車の追い越しを模した Find My パケットの RSSI の時 系列変化	53
24	実験 A: 自動車の追い越しを模した Nearby パケットの RSSI の時 系列変化	54

25	実験 A: 自動車の追い越し後, 信号で再度追い付く様子を模した Find My パケットの RSSI の時系列変化	55
26	実験 A: 並走車を模した Find My パケットの RSSI の時系列変化	55
27	実験 A: バス横の歩道を歩く歩行者を模した Find My パケットの RSSI の時系列変化	56
28	実験 A: 横断歩道を歩く歩行者を模した Find My パケットの RSSI の時系列変化	56
29	実験 A: バス停で待つ人を模した Find My パケットの RSSI の時系列変化	57
30	実験 A: 乗客 1 の Find My パケットの RSSI の時系列変化	57
31	実験 A: 乗客 1 の Android の BLE アドバタイジング・パケットの RSSI の時系列変化	58
32	アドレスキャリーオーバーアルゴリズムで紐付けた Android の BLE アドバタイジング・パケットの組み	59
33	実験環境: 環境全体の俯瞰図	61
34	実験環境: バス車内の俯瞰図	62
35	実験 B: 自動車の追い越し後, 信号で再度追い付く様子を模した 16 パケットの RSSI の時系列変化	64
36	実験 B: 並走車を模した 16 パケットの RSSI の時系列変化	65
37	実験 B: 並走車を模した Android スマートフォンの BLE アドバタイジング・パケットの RSSI の時系列変化	65
38	実験 B: 乗客 2 の Find My パケットの RSSI の時系列変化	67
39	実験 B: 乗客 2 の Android スマートフォンの BLE アドバタイジング・パケットの RSSI の時系列変化	67
40	実験 B: 乗客 4 の Find My パケットの RSSI の時系列変化	68
41	実験 B: 乗客 4 の Android スマートフォンの BLE アドバタイジング・パケットの RSSI の時系列変化	68
42	提案手法で車外分類に成功した対象外スマートフォンの間欠的な動き (実験 B)	81
43	二つの BLE アドバタイジング・パケットが確認できる MAC アドレス (実験 B)	82
44	三つの BLE アドバタイジング・パケットが確認できる MAC アドレス (実験 B)	83

47	間欠的な動きをするバス OD データの不正解パターン	86
48	受信間隔が大きいバス OD データと一致しているデータパターン	87

List of Tables

1	バス OD データの例	1
2	データ収集方法の比較 ¹	7
3	使用した iPhone/Android スマートフォン	21
4	実験条件: 機能のオン・オフ	23
5	iPhone の BLE アドバタイジング・パケット送信結果	23
6	Android スマートフォンの BLE アドバタイジング・パケット送信 結果	24
7	実験条件: オンライン・オフライン環境	25
8	Find My パケットの Manufacturer Data の長さ	26
9	平均 RSSI の相関係数	30
10	8 時 25 分天満屋発のバス OD データ	31
11	14 時 30 分天満屋発のバス OD データ	32
12	17 時 37 分天満屋発のバス OD データ	33
13	データセットの形式	44
14	大学内の実験で使用した iPhone/Android スマートフォン	47
15	iPhone SE2 & Pixel 4a (スマートフォン 1) のスケジュール表 (実 験 A)	49
16	iPhone 15 Pro & Pixel 4a (スマートフォン 2) のスケジュール表 (実験 A)	50
17	iPhone 15 Pro & Pixel 4a (スマートフォン 3) のスケジュール表 (実験 A)	50
18	スマートフォンの対応	50
19	iPhone SE2 & Pixel 4a (スマートフォン 1) のスケジュール表 (実 験 B)	63
20	iPhone 15 Pro & Pixel 4a (スマートフォン 2) のスケジュール表 (実験 B)	63
21	iPhone 15 Pro & Pixel 4a (スマートフォン 3) のスケジュール表 (実験 B)	63

22	実験 A で取得したデータのパターンごとの RSSI 平均値	70
23	評価データセットの MAC アドレスの内訳	70
24	実験 A のデータセットに対する車内外分類結果	71
25	実験 B のデータセットに対する車内外分類結果	72
26	実験 B のバス車外 MAC アドレスを全て含めたデータセット分類 結果	73
27	各手法の OD 検出精度比較	77
28	スニファターの設置位置による比較	85

1. はじめに

公共交通機関は人々の日常生活における移動手段として活用されている。電車、バス問わず通学から通勤まで様々な場面で使われている。特に地方部では、中小店舗の減少、病院の統廃合等による日常的に利用する施設の減少や、高齢ドライバーの免許返納に伴う移動手段の確保のための移動手段として、公共交通機関の重要性が高まっている。

しかし、特に地方部における過疎化の影響により公共バス利用者とバスドライバーの減少が問題となっている。国土交通省の資料 [1] では、「長期的な利用者の減少、コロナの影響による急激な落ち込みもあり、公共交通事業者の経営環境は悪化。自動車（バス・タクシー）の運転業務の賃金水準は低く、人手不足が深刻化（有効求人倍率は全職業平均の2倍程度の水準で推移）」と報告されている。

こうした状況により求められる効率的なバスサービスを実現するために、バス事業者は、各乗客がどこからどこまで乗車したかを記録したバス Origin-Destination (OD) データ [2] を活用している。バス OD データは表 1 に示すように、あるバス停で乗車し、あるバス停で降車した人数を記録したデータである。なお、空欄は該当区間の乗客が 0 人であることを示す。こうしたデータを用いて、バス事業者は、必要な路線と不要な路線の見極めや、ダイヤ変更を行なっている。実際、中部運輸局が実施した調査 [3] によると、49%のバス事業者が、系統別利用実態把握のために、乗降調査員による定期的なバス OD データの取得を実施している。

バス OD データを取得するには、各乗客がどのバス停で乗車し、どのバス停で降車したかを把握する必要がある。そのため、バス OD データを取得する多くのバス事業者では、乗降調査員による目視での人物追跡が行われている。しかし、こ

表 1: バス OD データの例

乗車バス停 \ 降車バス停	バス停 1	バス停 2	バス停 3	バス停 4
バス停 1			5	3
バス停 2	3		8	8
バス停 3	3	2		8
バス停 4	6		2	

した人力でのバス OD データの取得は人件費がかかるため、頻繁に実施することは困難である。こうした背景から、人力に頼らずバス OD データを自動的に取得しようとする研究が行われている。

バス OD データを自動取得する研究としては、IC カード、カメラ、スマートフォンを用いた手法が知られている。

- IC カード情報を使った手法 [4]: 高精度なバス OD データ生成が可能であるが、バス事業者と IC カード事業者が異なるため、IC カード情報を自由に使えない場合も多い。
- 運賃箱による集計: 現金利用者のみのカウントとなり、IC カード利用者も含めた正確な集計は難しい。
- カメラを使った手法 [5]: バス車内に設置したカメラで撮影した動画に画像処理アルゴリズムを適用し、バス OD データを生成する。バス車内には監視カメラが取り付けられていることが多く、二次利用可能ならばカメラの導入費用は低い。しかし、機械学習モデルを導入する場合高いコストがかかる、カメラの画角によっては利用者同士が重なってしまい見えなくなることににより性能が低下する等の問題がある。
- スマートフォンが発するパケットを用いた手法 [6][7]: パケットに含まれる Media Access Control (MAC) アドレスを追跡することで、バス車内のスマートフォンの追跡を行う。手法は低コストかつスマートフォンを所持していれば高い精度の追跡が可能である。

コスト面・精度面から本研究ではパケットを用いた手法に着目する。

スマートフォンが発する無線 LAN パケットのアドレス情報を用いた乗客の追跡の問題としては、MAC アドレスが定期的にランダムな値に変更されることとバス車外のスマートフォンが送信するパケットが存在することという二つの問題が知られている。前者は、スマートフォンの MAC アドレスがプライバシー保護のためにランダムに変更される (MAC アドレスランダム化) ため、単純な MAC アドレスの追跡では見失ってしまうという問題である。後者は、スマートフォンから送信されたパケットがどこから送信されたものが不明であり、バス車内のスマートフォンから送信されたパケットかバス車外のスマートフォンから送信されたパケットか区別がつかないという問題である。一つ目の問題に対して、観測範囲内の同一スマートフォンから送信されたパケットに含まれるランダム化された MAC アドレス

を追跡するアドレスキャリーオーバーアルゴリズムが提案されている [7][8]. 二つ目の問題に対して, Oransirikul らはパケットに含まれる Received Signal Strength Indicator (RSSI) と MAC アドレスの出現回数からバス車内のスマートフォンとバス車外のスマートフォンにリアルタイムで分類する手法を提案し, F1 スコアで 75% のバス車内人数推定を達成している [9]. Kawashima らは MAC アドレスの出現時間が短い場合, 走行中に発生した歩行者や自動車の中にあるスマートフォンによって送信されているものとしてフィルタリングする手法を提案している [7].

しかし, バス車内外分類において, パケットの RSSI をもとに閾値で処理する場合, 想定する車内環境に適した閾値を都度用意する必要がある. また, Kawashima らは信号や渋滞時において, MAC アドレスの出現時間にもとづくバス車内外分類では, 一部のバス車外の自動車, 歩行者の行動を考慮し切れておらず, 精度が低下する場面があったと述べている.

本研究では, バス OD データの推定精度向上を目的とした, BLE アドバタイジング・パケットの送信元のバス車内外分類の性能向上を, 車内環境に依存しない閾値を使って実現することを目指す. RSSI のような車内環境に依存する値を使わないようにしつつ, 分類性能を向上させる目的で, バス車外のスマートフォン所有者の行動を考慮した閾値を設定する. 周辺の BLE デバイスと双方向の通信を確立してブロードキャストされているアドバタイジング・パケットを取得するツール (スキャナー) で BLE アドバタイジング・パケットを受信することにより, スキャナーが BLE アドバタイジング・パケットを受信した時刻 (タイムスタンプ) を取得できる. バス車外のスマートフォン所有者は自動車で追い越しをする, バス停でバスを待つ等, スマートフォン所有者の行動を考慮したとき, タイムスタンプからスマートフォン所持者の行動に合わせた複数の閾値を設定することで, 単純な閾値処理よりも分類性能を向上させることができると考えた. スマートフォン所持者の行動を考慮した BLE アドバタイジングパケットの車内外分類の可能性について検証する目的で, スマートフォンのアドバタイジング・パケットを擬似的なバス車内外環境を再現した上で収集し, 出現時間と RSSI の特徴を分析した. バス車内外再現データから明らかになった本研究における貢献を以下にまとめる.

- バス車内外のスマートフォンが発する BLE アドバタイジング・パケットの受信時刻情報からバス車内外分類可能な受信時間/出現期間のパターンが存在することを示した
- バス車内外のスマートフォンが発する BLE アドバタイジング・パケットの受信時刻情報を使った BLE アドバタイジング・パケットのバス車内外分類は,

従来の受信時刻情報を使用した手法と比較して F1 スコアが最大で 23.3% 向上し, RSSI のようなバス車内の環境に依存する値を使用せずに高い分類性能を達成できることを示した

本稿の構成は以下の通りである. 2 章では, 関連研究について述べ, 3 章では Bluetooth Low Energy (BLE) のアドバタイジング・パケットの仕様とそれに含まれる RSSI と送信時間情報について調査する. 4 章では BLE アドバタイジング・パケットの RSSI と送信時間情報を使用した 5 種類の出現パターンについて定義し, パターンに合致した BLE アドバタイジング・パケットを車内外に振り分ける提案手法について述べる. 5 章では提案手法を評価するためのデータセットについて述べる. 6 章では提案手法の評価を行う. 7 章ではデータセット, バス車内外分類性能, バス OD データの推定結果に対する考察を述べる. 8 章では, 本稿をまとめる.

2. 関連研究

本章では、2.1 節でバス OD データ自動取得手法を紹介し、その中でスマートフォンの発するパケットを利用した手法に着目する理由を説明する。2.2 節でパケットを利用したバス OD データ自動取得において必須となる、ランダムに変化する MAC アドレスを追跡し続けるための手法であるアドレスキャリーオーバーアルゴリズムの研究について紹介する。2.3 節でアドレスキャリーオーバーアルゴリズムの精度を高めるために必要な、パケットの送信源をバス車内外に分類する手法について説明する。最後に、MAC アドレスの車内外分類の課題について言及する。

2.1 バス OD データ自動取得手法

バス OD データ自動取得は、任意のデータを使用してバス OD データを推定することにより行う。バス OD データ自動取得手法としては、バス IC カードを利用した手法、カメラ画像を利用した手法、スマートフォンのパケット情報を利用した手法の三つの手法が知られている。以下で、各手法の詳細について紹介する。

2.1.1 IC カードを利用したバス OD データ自動取得手法

Huang らは IC カードデータとバス GPS データとバス停留所データを使用したバス OD データの自動生成手法を提案した [4]。なお、この手法は、運賃を前払いする方式のバスを想定している。具体的には、乗車時には支払いのために IC カードを利用するが、降車時には何もしないことが想定されている。この想定のもと、この手法は、乗客が乗車したバス停を識別するために、バスの GPS データをもとに、密度ベースのクラスタリング手法によってバスの停車地点と対応するバス停を推定する。これによりバスがある時刻にどのバス停に滞在していたかが分かるため、その時刻に使用された IC カードを所持する乗客は、そのバス停でバスに乗車したと判断できるようになる。バス停の推定結果から IC カードの取引時刻を紐付け、どのバス停で乗車したかを特定する。対して、降車駅の推定は二つの場合に分けて行う。一つは次の乗車記録が存在する場合、もう一つはその日最後の乗車など次の乗車記録がない場合である。中国・蘇州市での実地調査による評価の結果、バス OD データの推定誤差が二人以内の割合は 94.3%、一人以内の割合は 72.8% となった。

2.1.2 カメラ画像を利用したバス OD データ自動取得手法

Komatsu らはバス車内に設置された 2 台の監視カメラの動画から乗客の乗降区間を推定する手法を提案した [5]. この手法では, バスの乗客は乗車口から乗車し, 降車口から降車することを想定している. この想定下では, 乗客の乗車時・降車時の様子をそれぞれカメラで撮影し, 両者の映像間で人物追跡することで, 各乗客の乗降バス停を特定し, バス OD データを作成することができる. 具体的には, この手法は人物検出, 人物追跡, 移動方向推定, 特徴抽出, 人物対応付けの五つのフェーズによって OD データの作成を行う. 人物検出フェーズでは, 物体検出アルゴリズムである Faster R-CNN により画像内の人物を検出する. 人物追跡フェーズでは, 検出した人物を物体追跡アルゴリズムである SORT により紐付ける. 移動方向推定フェーズでは, 人物追跡から得られた移動の始点・終点座標をもとに乗客の移動を 3 クラスに分ける. 特徴抽出フェーズでは, 画像認識モデルである ResNet50 により各人物画像から特徴ベクトルを抽出し, 同一人物間で特徴ベクトルのユークリッド距離が小さくなるように, 異なる人物ならユークリッド距離が大きくなるように人物画像ごとの特徴ベクトルを出力する. 最後にハンガリアンアルゴリズムを使って乗車時と降車時の人物を一対一に対応させる. 乗車人物と降車人物の紐付けの精度において, 最高で 100% を達成している.

2.1.3 スマートフォンのパケットを利用したバス OD データ自動取得手法

Pu らは Wi-Fi のパケットデータと GPS モジュールで GPS データを収集し, 乗客の OD データを自動推定する手法を提案した [6]. この手法は, バス内に設置した Wi-Fi のセンサーデバイスを用いて, 乗客のスマートフォンから発信されるパケットをキャプチャする. このキャプチャしたパケットに記録される MAC アドレスについて, 検出回数, 検出時間, 平均 RSSI, 最大 RSSI を計算する. GPS データからは最初の検出位置, 最後の検出位置, 移動距離, 平均速度, 最高速度を計算する. これらの計算した特徴量をもとに, FCM クラスタリングによって乗客・非乗客の二つのクラスに分ける. FCM クラスタリングで推定した結果から, 乗客の MAC アドレス数を算出する. この結果と, 曜日, 時間帯, その停留所が終点かどうかを示すフラグ情報を用いて, 実際の乗客数を推定する. 提案手法は乗車人数と降車人数を推定し, 平均絶対パーセント誤差で乗車人数の予測が 14.72%, 降車人数の予測が 17.41% を達成している.

2.1.4 バス OD データ自動取得手法の比較

バス OD データを自動取得する IC カードを利用した手法、カメラ画像を利用した手法、スマートフォンのパケットを利用した手法の三つについて、導入にかかる金額（コストの大きさ）、時間帯によるバス車内の明るさの変化（時間帯による影響）、乗車人数によるバス車内にいる人の重なり度合いの変化（乗車人数による影響）、バス車外のパケットを発しているスマートフォン数（車外の環境による影響）の四点から比較した。比較結果を表 2 に示す。

IC カードを利用した手法は、バス車内外の状況に影響されずにバス OD データを取得することが可能である。しかし、バス事業者が IC カードの情報を所持している場合があり、自由に使えない場合がある、IC カード決済用の機械を新しく導入する場合高いコストがかかる等のコスト面の欠点が存在する。

カメラ画像を利用した手法は、バス車外の環境に影響されずにバス OD データを取得することが可能である。しかし、乗車人数が多いほど利用者が重なってしまうことにより性能が低下する、機械学習モデルを搭載することを考えたとき、高いコストがかかる、夜間のような暗い時間帯では人の検出性能が低下するといった欠点が存在する。

スマートフォンのパケットを利用した手法は、時間帯に影響されず、低コストで利用できる。しかし、乗車人数が多くなると、パケットのランダム化された MAC アドレスのキャリーオーバーが失敗しやすくなる（パケットの MAC アドレスランダム化は非同期的に発生するため画像処理に比べて影響は小さい）、車外のスマートフォンから送信されたパケットの MAC アドレスの影響を受けるといった欠点が存在する。

比較結果について、汎用性とコストの 2 点から考慮したとき、スマートフォンの

表 2: データ収集方法の比較¹

	IC カード	カメラ画像	スマートフォンのパケット
コストの大きさ	×	×	
時間帯による影響		×	
乗車人数による影響		×	
車外の環境による影響			×

¹IC カードは利用する場合 IC カード情報を所持している事業者から購入するとし、カメラ画像はバス車内に設置された監視カメラで撮影したものを使用するとする。

パケットデータを使用した手法が最適であると考えた。

2.2 アドレスキャリーオーバーアルゴリズム

パケットを利用するバスODデータ自動取得手法では、ランダムに変化するMACアドレスを変化前と変化後で紐付けるアドレスキャリーオーバーアルゴリズムが必要となる。iOSやAndroidを搭載する一般的なスマートフォンでは、プライバシー保護の観点から、自身のMACアドレスを定期的にランダムな別のMACアドレスに変化させる仕組みが備わっている。パケットを利用するバスODデータ自動取得手法では、MACアドレスを追跡対象のスマートフォンの識別子としているため、そのMACアドレスが変更されるとスマートフォンの追跡ができなくなる。加えて、現在Wi-FiのプロブリクエストのMACアドレスがランダム化される間隔は非常に短く、先行研究の手法の適用は困難である。対して、BLEアドバタイジング・パケットは数分から数十分の間MACアドレスが同じ値のままである。

こうした事情から、BeckerらはBLEアドバタイジング・パケットのアドレスキャリーオーバーアルゴリズムによる紐付けを提案した[8]。iOS/macOSを搭載したスマートフォンのパケットにはMACアドレスと製造者固有のデータが存在する。この製造者固有のデータの種類の中で、Nearbyと呼ばれる5バイトのデータとHandoffと呼ばれる14バイトのデータを使用する。スマートフォンのMACアドレスとNearby/Handoffのデータは変更タイミングが非同期であり、MACアドレスが変化したタイミングでNearby/Handoffのデータを確認し、同じものを紐付ける。具体的なアルゴリズムの動作フローとして、まずアドバタイジング・パケットを継続的にモニタリングする。この時、既知のアドレスとの照合を行う。アドレスが変更された場合、Nearby/Handoffのデータを基に変化先のMACアドレスを紐付ける。提案手法により、最長で53分間の追跡に成功しており、MACアドレスがランダム化されても一定期間追跡可能であることを示した。KawashimaらはBeckerらのアドレスキャリーオーバーアルゴリズムを応用し、BLEアドバタイジング・パケットに含まれるランダム化されたMACアドレスをRSSIの平均値で紐付けた。BLEアドバタイジング・パケットとして、新型コロナウイルス禍においてコロナウイルス感染者との接触検知を検知する目的でAppleとGoogleにより開発されたCOCOAと呼ばれるアプリケーションから発されるBLEアドバタイジング・パケットを利用する。COCOAのBLEアドバタイジング・パケットはおよそ15分間隔でMACアドレスランダム化が発生し、ペイロード長はMACアドレスが変化しても変わらない。具体的なアルゴリズムの流れをアルゴリズム

1 に示す。このアルゴリズムでは、COCOA の BLE アドバタイジング・パケットを抽出し、その中で MAC アドレスが変わったタイミングから 50 秒以内に新しく出現した、ペイロード長が等しい MAC アドレスを見る。この中で RSSI の平均値を計算し、平均値が最も近い MAC アドレスを変化後の MAC アドレスとして紐付ける。

アルゴリズム 1 路線バス用のアドレス・キャリーオーバー・アルゴリズム

```
1: function ADDRESS_CARRYOVER(observed_dct, noise_eliminate_addrs)
2:   for  $ne \leftarrow noise\_eliminate\_addrs$  do
3:      $o\_addrs \leftarrow observed\_dct$  のアドレス
4:      $lo\_time \leftarrow observed\_dct[ne]$  の観測終了時刻
5:      $up\_time \leftarrow lo\_time + extract\_int$ 
6:     for  $o \leftarrow o\_addrs$  do
7:        $fst \leftarrow observed\_dct[o]$  の観測開始時刻
8:       if  $lo\_time \leq fst$  and  $fst \leq up\_time$  then
9:          $addrs$  に  $o$  を追加
10:      end if
11:    end for
12:     $carryover\_dct$  に  $\{ne : addrs\}$  を追加      ▷  $ne$  がキーで  $co\_addrs$  が値の辞書
13:    for  $co\_addrs \leftarrow carryover\_dct[ne]$  do
14:       $rss\_df \leftarrow |disappeared\_dct[ne]$  の RSSI  $- observed\_dct[co\_addrs]$  の RSSI|
15:    end for
16:     $min\_rss\_df \leftarrow \min rss\_df$ 
17:     $rss\_idx \leftarrow rss\_df$  で  $min\_rss\_df$  と一致する値の添え字
18:     $target\_addr \leftarrow carryover\_dct[ne][rss\_idx]$ 
19:    if  $min\_rss\_df < rss\_th$  and  $observed\_dct[target\_addr]$  のペイロード長 =
       $observed\_dct[ne]$  のペイロード長 then
20:       $observed\_dct[target\_addr]$  の変更前の MAC アドレス  $\leftarrow observed\_dct[ne]$  の
      MAC アドレス
21:    else
22:       $observed\_dct[ne]$  の降車フラグを True にする
23:    end if
24:     $carryover\_dct[ne]$  を削除
25:    for  $c\_key \leftarrow carryover\_dct$  のキー do
26:      if  $target\_addr \in carryover\_dct[c\_key]$  then
27:         $carryover\_dct[c\_key]$  から  $target\_addr$  を削除
28:      end if
29:    end for
30:  end for
31: end function
```

2.3 パケット発信源のバス車内外分類手法

パケットを利用したバス OD データ自動取得手法では、アドレスキャリーオーバーアルゴリズムの精度を高めるために、観測したパケットの送信源を、バス車

内かバス車外のスマートフォンの MAC アドレスであるかに分類する必要がある。Oransirikul らはバス車内の乗客数推定を目的として、Wi-Fi プローブリクエストパケットを利用し、そのパケットの発信元がバスの車内外どちらであるか分類する手法を提案した [9]。この手法は、バス内に設置した Raspberry Pi とトランシーバを用いてスマートフォンが定期的送信するプロブリクエストパケットを収集する。収集したパケットから送信元 MAC アドレス、RSSI、受信時刻を抽出し、バスの車内外を分類するための特徴量とする RSSI が -40dBm 以上かつパケット数 2 以上、あるいは RSSI が -60dBm 以上かつ検出時間が 90 秒以上のスマートフォンを乗客のスマートフォンとして分類する。提案手法により、6 路線のバスで取得したパケットに対して評価を行い、バス車内外分類の結果 F1 スコア 75% で乗客数の推定に成功している。

Kawashima らは公共バスの乗客追跡を目的としたアドレスキャリーオーバー手法の研究において、時刻情報を用いて、BLE アドバタイジング・パケットの発信元がバスの車内外のどちらであるか分類する手法を提案している [7]。提案手法は、車外スマートフォンの MAC アドレスは出現時間が短いとして 50 秒未満観測された MAC アドレスを車外の MAC アドレス、50 秒以上観測された MAC アドレスを乗客の MAC アドレスとした。

2.4 バス車内外分類手法の課題

パケットのバス車内外分類を、Oransirikul らの手法で使用されている RSSI のような車内環境に依存する値で閾値処理する場合、車内環境に合わせた閾値の調整が必要であり、汎用的なバス車内外分類は難しい。また、パケットの受信時間を使った手法では、車内環境による閾値調整が不要であるが、現状車外の状況を十分に考慮できていない。

そこで、閾値調整無しでバス車外の状況を踏まえてパケットをバス車内外に分類する手法について考える。バスの環境をバス車内とバス車外で分けて考えたとき、バス車内は信号やバス停で停車する移動体のため、MAC アドレスの出現から消失するまでには渋滞時にバスの横を通る歩行者の動きやバス停で止まっている歩行者の動きといった、パケットが出現する時間や受信間隔に特徴が現れると考えられる。すなわち、同一 MAC アドレスのタイムスタンプを見たとき、パケットの出現時間や受信間隔に特徴が現れると考えられる。既存の単純な時間情報を用いた手法ではこのような時系列の特徴は考慮されていない。

したがって、バス車内とバス車外でパケットの特徴にどのような違いがあるか

考える。バス車内の環境は、利用者が着席または定位置に滞在し、動きがほとんど発生しない。加えて、バスを乗車・降車するタイミングで、RSSIが単調増加・単調減少するような変化が発生すると予想できる。対して、バス車外の環境は、バスの速度に対し歩行者は相対的に遅く、走行している自動車は相対的に速いことが考えられる。つまり、パケットの出現時間が非常に短いことが予想できる。加えて、一度追い越し等が発生した後に信号やバス停で追いつきが発生した場合も、同じパケットが一瞬だけ出現し、再度確認できる、というような動きが考えられる。このような特徴を利用することで、バスの車内分類性能を向上できると考える。加えて、このような特徴はRSSIのようにバスの車内環境に依存しないため、閾値の調整が不要である。

本研究では以下の二つの課題に取り組む。

1. バス車外のパケットの時系的な特徴を明らかにする。
2. 1で明らかにした時系列的な特徴を基に、バスの車内環境に合わせた閾値調整をせずにバス車内外分類を行う。

3. BLE アドバタイジング・パケットに関する調査

本章では、3.1 節でバス車内外分類の前提として、本研究で想定するバス OD データ自動取得手法を説明する。3.2 節では、想定するバス OD データ自動取得手法の実現可能性を検証する。3.3 節では、BLE アドバタイジング・パケットのバス車内外分類のために、既存のバス路線で取得したデータ (バス路線データ) を用いて 2.4 節で示した仮説について検証する。

3.1 想定するバス OD データ自動取得手法

本研究では、Kawashima らのバス OD データ自動取得手法を参考にバス OD データを推定する。具体的には、バス内に設置したスキャナーで BLE アドバタイジング・パケットを取得し、BLE アドバタイジング・パケットの中に含まれているランダム化された MAC アドレスを同一スマートフォンとして紐づける。紐づける際に、最後に受信できた時間から任意の秒数の間に出現した MAC アドレスから紐付け先を選択するが、この部分は BLE アドバタイジング・パケットの仕様の違いを考慮して設定し直す。紐付けた MAC アドレスの組み合わせの中で、あるバス停の停車期間に初めて検出され、別のバス停の停車期間に最後に検出されたものを乗客のスマートフォンが送信した BLE アドバタイジング・パケットとすることでバス OD データを推定する。

Kawashima らの手法で使用されている BLE アドバタイジング・パケットはコロナウイルス陽性者の全数見直しにより陽性登録可能な人が限られることから、現在利用が停止されている [10]。そこで、Kawashima らが用いている BLE アドバタイジング・パケットの代わりに、別の BLE アドバタイジング・パケットを使用することを考える。BLE アドバタイジング・パケットについて調査した結果、現在は iOS の Find My と呼ばれる BLE アドバタイジング・パケット (Find My パケット) と Android の Google LLC のサービスとして登録されている UUID が 0xFEED3 である BLE アドバタイジング・パケット (0xFEED3 パケット) がアドレスキャリーオーバーアルゴリズムに適用可能であることが分かった。以降では、本研究で想定する BLE アドバタイジング・パケットを導出するために、BLE の仕様、BLE アドバタイジング・パケットの仕様、およびアドレスキャリーオーバーアルゴリズムに適用可能な BLE アドバタイジング・パケットの調査について述べる。

3.1.1 Bluetooth の概要と規格

Bluetooth とは近距離の無線通信規格の一つである。同じ無線通信規格に Wi-Fi があるが、以下の点で異なる。

- 帯域
- 周波数
- 通信距離
- 消費電力

Wi-Fi は通信速度が速く、長距離通信が可能である。対して、Bluetooth は通信速度と通信距離は Wi-Fi に比べ劣るものの、消費電力が小さいメリットがある。また、Wi-Fi は周波数 2.4GHz 帯、5GHz 帯、60GHz 帯を使用しているのに対し、Bluetooth では 2.4GHz 帯を使用している。

Wi-Fi ではプロブクエストに MAC アドレスが含まれるが、MAC アドレスランダム化の間隔が非常に短く、これを識別子として追跡することは難しい。対して、Bluetooth の MAC アドレスランダム化の間隔は数分から数十分と比較的長い。そこで本研究では Bluetooth を使用することとする。

Bluetooth の規格の中でも二つの通信規格が存在する。最初にできた規格は Bluetooth Classic と呼ばれ、主にワイヤレス電話接続、ワイヤレスヘッドホン、ワイヤレススピーカーなどのオーディオ分野で使用されている。対して、BLE は Bluetooth Classic をより省電力方面に寄せた規格であり、ウェアラブルデバイスやキーボードといった分野で使用されている。また、条件によっては 200 ~ 300m の長距離通信も可能である [11]。

3.1.2 BLE アドバタイジング・パケット

アドバタイズとはブロードキャスト通信のことであり、不特定多数の相手にデータを送信する通信方式である。アドバタイズにより送信されたパケットにより、受信者はどんな機器が存在しているか、どんなメッセージを送信しているかを知ることができる。

BLE が通信に利用する周波数帯である 2.4GHz 帯には、2402MHz から 2480MHz の 40 個のチャンネルがあり、チャンネル間の帯域幅は 2MHz である。2402MHz, 2426MHz, 2480MHz はアドバタイズで使用され、それ以外は対一の通信で利用される。ア

ドバタイズの3チャンネルはWi-Fiで頻繁に利用される周波数を避け、できるだけ干渉の影響がないように配置されている。加えて、一回のアドバタイズで3チャンネル全てでアドバタイズが行われており、チャンネルが混雑していても空いているチャンネルでデータの送信が可能である。

BLEアドバタイジング・パケットは、スキャナーやBLEアドバタイジング・パケットを通信に介入せず傍受するツール（スニファア）を使ってスキャンすることで受信できる。アドバタイズのスキャンとは、アドバタイズ用のチャンネルを定期的にホッピングし、パケットが来るのを待つ。パケットが来た場合はそのパケットを受信する。パケットを受信する方法は以下の二つがある。

- パッシブスキャン: BLEアドバタイジング・パケットを単にスキャンする。
- アクティブスキャン: アドバタイズパケットのPDU Typeがスキャン可能を示していた場合、アドバタイズしてきたデバイスに対しスキャン要求パケットを送信する。これを受け取ったデバイスはスキャン応答パケットをスキャナーに送信する。スキャン要求にはアドバタイズパケットの追加情報を含めることができる。

アドバタイズパケットはBluetooth Special Interest Group (Bluetooth SIG) によって定義されたフォーマット [12] がある。図1にBLEパケットの基本フォーマットを示す。パケットは四つの必須フィールドで構成されている。それぞれのフィールドについて以下に述べる。

プリアンブル プリアンブルは0と1のビットが交互に並ぶ固定シーケンスである。周波数同期、シンボルタイミングの推定といった、受信機が後続のデータを正確に読み取るために使われる。プリアンブルは変調が1MHzの時1バイト、変調が2MHzの時2バイトとなる。

アクセスアドレス 受信したパケットがどの通信に属するパケットかを区別する目的で使用される4バイトの識別子である。アドバタイジング・パケットでは固定のアドレス (0x8E89BED6) が使用される。

PDU パケットが運ぶデータを指定するヘッダー部分が2バイト、ペイロード部分が1~255バイトのフィールドである。PDUヘッダーには、アドバタイズパケットの種類を表す4ビットのPDU Type、通信チャンネル選択アルゴリズムを

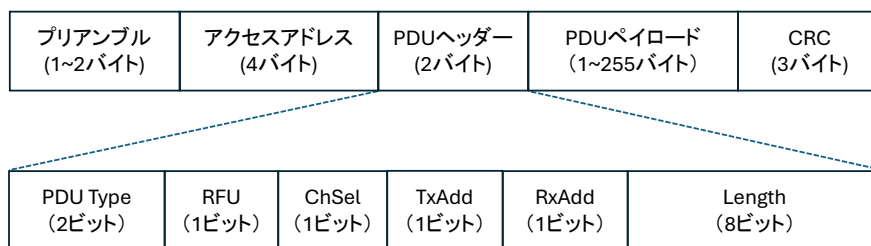


図 1: BLE のアドバタイズパケットフォーマット

表す 1 ビットの Channel Selection (ChSel) , 送信アドレスタイプを表す 1 ビットの TxAdd, 受信アドレスタイプを表す 1 ビットの RxAdd, ペイロードの長さを表す 8 ビットの Length が含まれている. なお, RFU は現在使用されていない 1 ビットの未定義領域である. PDU のペイロードには, 後述する製造者・サービス提供者のデータの中身等が格納されている.

Cyclic Redundancy Check (CRC) エラー検出用の 3 バイトのフィールドである. パケットの破損が発生した時それを検出することができる.

PDU ペイロードの構成は指向性アドバタイズと非指向性アドバタイズで分けられる. 指向性アドバタイズとは特定のデバイスに向けたアドバタイズである. 指向性アドバタイズの PDU ペイロード部分のフォーマットを図 2 に示す. フィールドには送信元 MAC アドレスを示す Advertiser Address (AdvA) , 送信先 MAC アドレスを示す Target Address (TargetA) があり, とともに 6 バイトである.

対して, 非指向性のアドバタイズパケットの PDU ペイロード部分のフォーマットを図 3 に示す. フィールドには 6 バイトの AdvA と 0 ~ 31 バイトのアドバタイズデータ (AdvData) がある. AdvData のフォーマットを図 4 に示す. アドバタイズデータは Advertising Data Structure (AD Structure) と呼ばれる構造ごとに分かれており, 各構造に AD Structure の長さを表す Length, AD Structure のタイプを表す AD Type, Advertising Data (AD Data) が含まれている. Length, AD Type は固定でそれぞれ 1 バイトであり, AD Data は AD Type によって様々なデータが格納される.

次に, PDU ペイロードの AD Structure 部分について, 今回使用する iOS デバイスのパケットの詳細について述べる. なお, Android デバイスが発するパケットに関する文献は調査した限り存在しなかった. そこで, Android パケットの詳細については 3.2 節で調査した結果をもとに説明する.

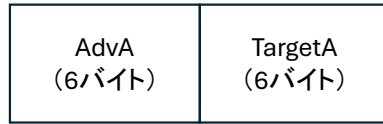


図 2: 指向性アドバタイズの PDU ペイロード

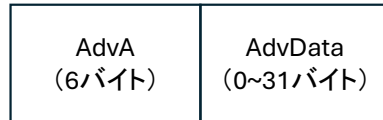


図 3: 非指向性アドバタイズの PDU ペイロード

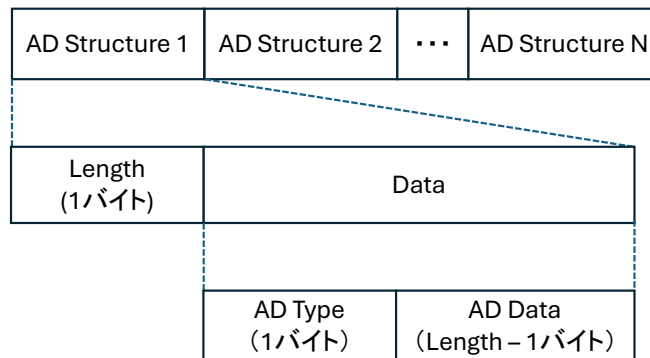


図 4: AdvData のフォーマット

3.1.3 OS ごとの AD Structure の技術仕様と扱うデータの詳細

AD Structure のタイプ (AD Type) には、複数種類が存在する。この中で、本研究で扱う iOS デバイスの BLE アドバタイジング・パケットに含まれる Manufacturer Specific Data, Android スマートフォンの BLE アドバタイジング・パケットに含まれる Service Data と呼ばれる AD Type のデータについて述べる。

3.1.4 iOS の AD Structure のフォーマット

iOS デバイスの PDU ペイロードの AD Structure フォーマットについて、AD Type が Manufacturer Specific の場合のものを図 5 に示す。Length は 1 バイト、AD Type は 1 バイト、Manufacturer ID は 2 バイト、Manufacturer Data は Length で定義されている長さから AD Type と Manufacturer ID を合わせた 3 バイトを引いたバイト数である。Length と AD Type を除くこれらのフィールドについて

Length (1バイト)	AD Type: 0xFF (1バイト)	Manufacturer ID (2バイト)	Manufacturer Data (Length-3バイト)
------------------	-------------------------	---------------------------	------------------------------------

図 5: Manufacturer Specific タイプのフォーマット

以下に述べる.

Manufacturer ID 製造者識別用のフィールドである. Apple Inc. であれば 0x004C, Nordic Semiconductor であれば 0x0059 等その企業に割り当てられた値が格納される. これらの ID は Bluetooth SIG に登録されており, Bluetooth SIG の Assigned Numbers[13] で確認できる.

Manufacturer Data 可変長のフィールドで, 各製造者が独自の形式でデータを定義することができる.

分析に使用する iOS のパケットとして, Nearby と呼ばれる BLE アドバタイジング・パケット (Nearby パケット) と Find My パケットを利用する. それぞれの BLE アドバタイジング・パケットについて説明する.

3.1.5 Nearby パケットの技術仕様

Nearby パケットは Apple デバイスの連携機能である Continuity プロトコルの一部として実装されている. Apple デバイスから送信される BLE アドバタイジング・パケットの一種で, iOS10 から実装された. 具体的な実装について開発者からの発表はされていない. Martin ら [14] はデバイスの状態を他のデバイスに認識させることが意図されていると推測している. 同氏らは Nearby パケットをリバースエンジニアリングし, 以下のような挙動を示すことを明らかにした.

iOS 10/11 及び macOS High Sierra

- 約 30 秒の非アクティブ期間後にタイムアウトする
- タイムアウト後は送信を停止

Type (2バイト)	Length (2バイト)	Location Sharing (1バイト)	Action Code (1バイト)	Variable Length Data
----------------	------------------	-------------------------------	--------------------------	----------------------

図 6: Nearby パケットのフォーマット

iOS 12 及び macOS Mojave 以降

- 以下の条件を満たす限り継続的に送信される
 - BLE サービスが有効化されている
 - デバイスの電源が入っている
 - macOS の場合, 蓋が閉じられていない
- 毎分 200 以上のフレームを送信

また, Manufacturer Data のフォーマットは図 6 のようになっている. Type は Nearby パケットを表す 0x10 が格納されている. Location Sharing にはデバイスが家族や友人と「位置情報を共有」するように設定されているかどうかを示し, Action Code フィールドには Apple デバイスのアクションまたは状態を示す値が格納されている.

3.1.6 Find My パケットの技術仕様

Find My は Apple デバイスの位置追跡のための機能として 2019 年に iOS 13, macOS 10.15, watchOS6 で実装された. 紛失したデバイスの位置情報を他の Apple デバイスを介して所有者に通知するシステムである. Find My は役割で見たとき, 三つのデバイスが存在する.

Lost Device インターネット接続を失ったデバイス. インターネット接続を失った時点で BLE のアドバタイズを開始する. 15 分ごとに新しい公開鍵を使用し, 2 秒間隔で BLE アドバタイジング・パケットを送信する.

Finder Device GPS 機能を搭載した iPhone と iPad. 受信した Lost Device のアドバタイズパケットから暗号化された位置レポートを生成する.

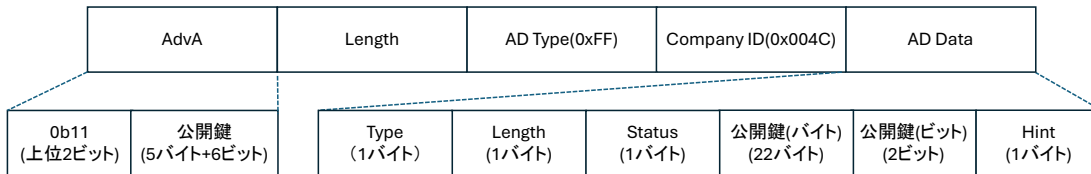


図 7: Find My パケットのフォーマット

Owner Device 同じ Apple ID を共有しているデバイス群. macOS や iOS の Find My パケットを使用することで, 同じ所有者の任意のデバイスを検索することができる.

Heinrich ら [15] は Find My のパケットのフォーマットについてリバースエンジニアリングにより調査した. 同氏らが調査した Find My の PDU ペイロードを図 7 に示す. Type は Find My パケットを表す 0x12 が格納されている. status はバッテリーレベル等デバイスの状態を表し, Hint にはには 0x00 が設定されているが具体的な用途は不明である. Find My パケットの公開鍵は 28 バイトである. この長さそのまま格納すると Adv Data の上限値を超えてしまうため, そのまま格納することができない. これに対処するため, AdvA の 6 バイトのうち BLE 規格の要件として固定値になっている先頭 2 ビット以外に公開鍵の一部を格納する. 残りの 22 バイトと 2 ビットを Manufacturer Data 部分に格納する.

3.2 BLE アドバタイジング・パケットに対するアドレスキャリーオーバーアルゴリズムの適用に関する検証実験

本節では, アドレスキャリーオーバーアルゴリズムに使用可能な BLE アドバタイジング・パケットが存在するかどうかを検証する目的で, iPhone/Android スマートフォンで取得できる BLE アドバタイジング・パケットの送信条件, 構造, ペイロードについて調査する. 次に, アドレスキャリーオーバーアルゴリズムに適用できる BLE アドバタイジング・パケットかどうか確認するために二つの実験を行い, その結果について述べる.

3.2.1 実験方法

本章で行う実験は全て Raspberry Pi 4 Model B と Raspberry Pi 5 に TP-Link の Bluetooth 4.0 ナノ USB アダプタ UB400 を BLE ドングルとして取り付けたス

キャナーを利用して実施した。BLE アドバタイジング・パケットをスキャナーでアクティブスキャンを実行することで収集し、収集したパケットを pcap ファイルで保存する。パケットキャプチャツールとして hcitool version 5.66 と btmon version 5.66 を使用した。iPhone3 台, Android スマートフォン 3 台を使って実施した。それぞれの OS のバージョンとスマートフォン名を表 3 に示す。

OS が異なることで、結果が大きく異なる可能性がある。そこで、異なる OS のスマートフォンを用いることで、OS のバージョンの違いを考慮した検証ができると考えた。実験のために、それぞれのスマートフォンには異なるアカウントを作成し iPhone には iCloud アカウント, Android スマートフォンには Google アカウントでログインしている。3.2.3 節の実験では、スキャナーと同じ位置にスマートフォンを設置し、それぞれ 1 分間アクティブスキャンを実施した。3.2.4 節の実験では、スキャナーから距離を 1.7m 離れた位置にスマートフォンを設置し、3 時間アクティブスキャンを実施した。この時、図 8 に示すように検証対象のスマートフォンには真隣にスキャナーを設置し、観測対象のスマートフォンが BLE アドバタイジング・パケットを送信しているかどうかを確認した。3.2.5 節の実験では、スキャナーから 0m ~ 5.4m の間で 0.6m ずつ距離を変えてそれぞれの距離でアクティブスキャンを実施した。この時、3.2.4 節の実験同様、検証対象のスマートフォンには真隣にスキャナーを設置し、観測対象のスマートフォンが BLE アドバタイジング・パケットを送信しているかどうか確認した。この実験では乗客がバスに乗車している平均時間 (平均乗車時間) を国土交通省の資料 [16][17] から、バスの平均時速を 20km, 乗客の平均利用距離を 8km として、それぞれの距離で 24 分間

表 3: 使用した iPhone/Android スマートフォン

OS	機種名	製造者	バージョン
iOS	iPhone SE2	Apple	17.5.1
	iPad Pro		16.7.4
	iPhone 7		15.8.3
Android	Pixel 4a	Google	13
	AQUOS sense3	SHARP	11
	ROG Phone 5s	ASUS	13

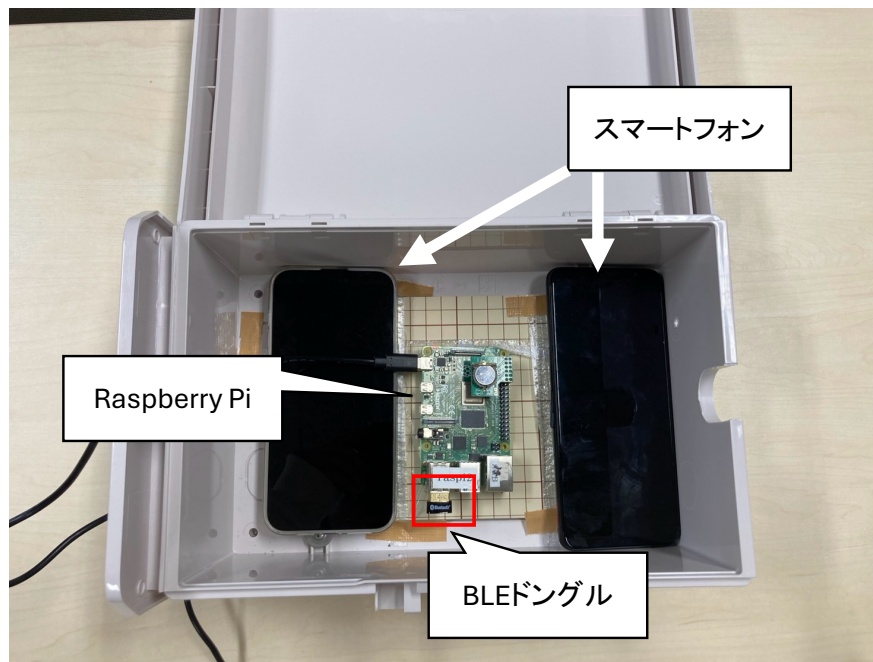


図 8: 検証対象のスマートフォンと確認用のスキャナー

実施する。また、横浜市が公表しているバス車両の大きさに関する情報 [18] と神奈川県中央交通が公表しているバスの座席間隔の情報 [19] から、バスの大きさは 10m 程度、バスの座席間隔は 0.6m とした。これらの条件をもとに、バスの座席間隔ずつスマートフォンをずらした時、RSSI がどのように変化するか調査した。スキャナーの位置は真ん中に設置した場合 5m まで観測できれば十分なため、距離の最大値は 5.4m としている。

3.2.2 実験環境

実験は奈良先端科学技術大学院大学の B 棟 2 階の 208 号室で実施した。部屋は一面が窓ガラス、残り三面がコンクリート壁となっており、部屋の広さは 9m × 9m × 3m 程度である。

3.2.3 パケットの送信条件に関する実験

iPhone と Android スマートフォンについて、iPhone では Nearby パケットと Find My パケット、Android スマートフォンではどのような BLE アドバタイジング・パケットが発されているか確認する目的で、スキャナーと同じ位置にスマート

フォンを設置し、それぞれで1分間アクティブスキャンを実施した。この時、Find My パケットの送信条件を調べるため、iPhone と Android スマートフォンそれぞれについて、「デバイスを探す」機能、位置情報、Bluetooth の有効化・無効化を表4に示す通りに組み合わせた条件下で、iPhone では Find My パケットが、Android スマートフォンでは何らかの BLE アドバタイジング・パケットが、それぞれのオン・オフで挙動が変化するかも確認した。

送信されていたパケットとパケットの送信条件について、iPhone の場合の結果を表5に、Android スマートフォンの場合の結果を表6に示す。

実験に使用した iPhone についてはいずれも Nearby パケットと Find My パケットが取得できていることが確認できた。また、Nearby パケットは Bluetooth を無

表 4: 実験条件: 機能のオン・オフ

実験 No	デバイスを探す	位置情報	Bluetooth
1			
2	×		
3		×	
4			×

表 5: iPhone の BLE アドバタイジング・パケット送信結果

パケット種別	実験 No.	機種名		
		iPhone 7	iPhone SE2	iPhone 15
Find My	実験 1			
	実験 2	×	×	×
	実験 3	×	×	×
	実験 4	×	×	×
Nearby	実験 1			
	実験 2			
	実験 3			
	実験 4	×	×	×

効化しない限り送信され続けていた。対して、Find My パケットはいずれかの機能を無効化すると、送信が観測されなくなった。「デバイスを探す」機能は位置情報を使用しているため、位置情報を無効化すると送信が停止することは予想通りの挙動であると言える。

iPhone の Find My パケットについて確認したところ、Find My パケットは2種類存在していることが分かった。一方は Manufacturer Data が4バイトであり、もう一方は Manufacturer Data が31バイトであった。

Android スマートフォンからは共通して AD Type が Service Data (0x16) を示す BLE アドバタイジング・パケットが送信されていた。AQUOS sense3 を除いて、Bluetooth を無効化しない限り送信され続けていた。AQUOS sense3 のみ、位置情報を無効化すると、送信が観測されなくなった。原因は不明であるが、Android スマートフォンから送信されている AD Type が Service Data (0x16) の BLE アドバタイジング・パケットは位置情報に関連するものであると考えられる。

Android スマートフォンから送信されていたパケットについて、AD Structure のフォーマットを図9に示す。UUID 部分の 0xFEED3 について Bluetooth SIG の Assigned Numbers を確認すると、Google LLC のサービスとして登録されているものであることが分かった。また、Service Data 部分については PDU Type が Scan Response を示すパケットに27バイトの値が格納されていることが確認できた。確認できた Service Data の例を図10に示す。先頭の3バイトは固定値であり、残りの24バイトは変化している変動値であった。

これらの結果から、iPhone では「デバイスを探す」、位置情報、Bluetooth の機能、Android スマートフォンでは、位置情報、Bluetooth の機能の有効化・無効化によって送信される BLE アドバタイジング・パケットが変化することが明らか

表 6: Android スマートフォンの BLE アドバタイジング・パケット送信結果

パケット種別	実験 No.	機種名		
		Pixel 4a	AQUOS sense3	ROG Phone 5s
0xFEED3	実験 1			
	実験 2			
	実験 3		×	
	実験 4	×	×	×

Length (1バイト)	AD Type: 0x16 (1バイト)	16-bit UUID: 0xFEED3 (2バイト)	Service Data (Length-3バイト)
------------------	-------------------------	--------------------------------	-------------------------------

図 9: Android スマートフォンから送信されていた BLE アドバタイジング・パケットのフォーマット

固定値: 4a1723 (3バイト)	変動値 (24バイト)
-----------------------	----------------

図 10: Android スマートフォンから送信されていた Service Data の構成

になった。そこで、これ以降で実施する実験では BLE アドバタイジング・パケットをスマートフォンが送信するようにするために「デバイスを探す」機能、位置情報、Bluetooth は全て有効化して実験する。

実験から、Find My パケットは Manufacturer Data が 4 バイトのものと 31 バイトのものが存在することが分かった。Find My パケットはオフラインの時に公開鍵を含むことが分かっている。そこで、オフライン環境下の条件とそれによる Manufacturer Data の条件について調査した。具体的な実験条件について表 7 に示す。オンライン環境として Wi-Fi による通信、SIM カードによる通信を想定した。SIM カードには Y!mobile のものを使用した。オンライン・オフライン環境に関する条件以外は、前述のスマートフォンの機能のオン・オフに関する実験と同じである。Find My パケットの Manufacturer Data 部分の大きさがどのように変化したかの結果を表 8 に示す。なお、表の数値は Manufacturer Data 部分の長さを示している。

実験結果からいずれかの方法でネットワークに繋がっていれば Manufacturer Data が 4 バイトの Find My パケットを送信し、ネットワークに繋がっていなけれ

表 7: 実験条件: オンライン・オフライン環境

実験番号	Wi-Fi	SIM
実験 5		
実験 6		×
実験 7	×	
実験 8	×	×

ば Heinrich ら [15] が述べている AD Structure と同じ Find My パケットを送信していることが確認できた。Manufacturer Data の 27 バイトから公開鍵の 23 バイトを差し引くと 4 バイトとなることから、Find My パケットはオンライン環境下では公開鍵を含まないパケットを送信し、オフライン環境下では公開鍵を含むパケットを送信していると考えられる。

iPhone は、一般的に有効化された SIM カードが挿入されている。Find My の公開鍵に関する仕様とスマートフォンが SIM カードにより Wi-Fi がない環境でもオンライン状態になることから、Find My パケットによりスマートフォン以外のオフラインデバイスをフィルタリングできると考える。以降は、iPhone の追跡はオンライン環境下で発されている Find My パケットを利用する。

3.2.4 BLE アドバタイジング・パケットの MAC アドレスランダム化の間隔の調査のための実験

BLE アドバタイジング・パケットの MAC アドレスは、同じタイミングで複数変化する、あるいは変化の周期が短い場合、キャリアオーバーする候補が増加するため、紐付けは難しくなる。そこで、BLE アドバタイジング・パケットが同時に変更する確率、MA アドレスの変化周期について検証する。実験環境は 3.2.3 節と同様で、3 時間スキャンを実行し、OS ごとに変化タイミングがどの程度重複するか確認した。なお、変化タイミングが重複するとは、任意の MAC アドレスが変化し、新しい MAC アドレスが出現するまでに、別の MAC アドレスが変化することを指す。

iPhone と Android スマートフォンそれぞれについて、スマートフォンごとに見た時に MAC アドレスが変わった時刻に色を変更して表示した図を図 11 に示す。図中の赤い丸と青い丸の部分が、MAC アドレスの変化タイミングが重複している

表 8: Find My パケットの Manufacturer Data の長さ

実験 No.	iPhone 7	iPhone SE2	iPhone 15
実験 5	4	4	4
実験 6	4	4	4
実験 7	4	4	4
実験 8	27	27	27

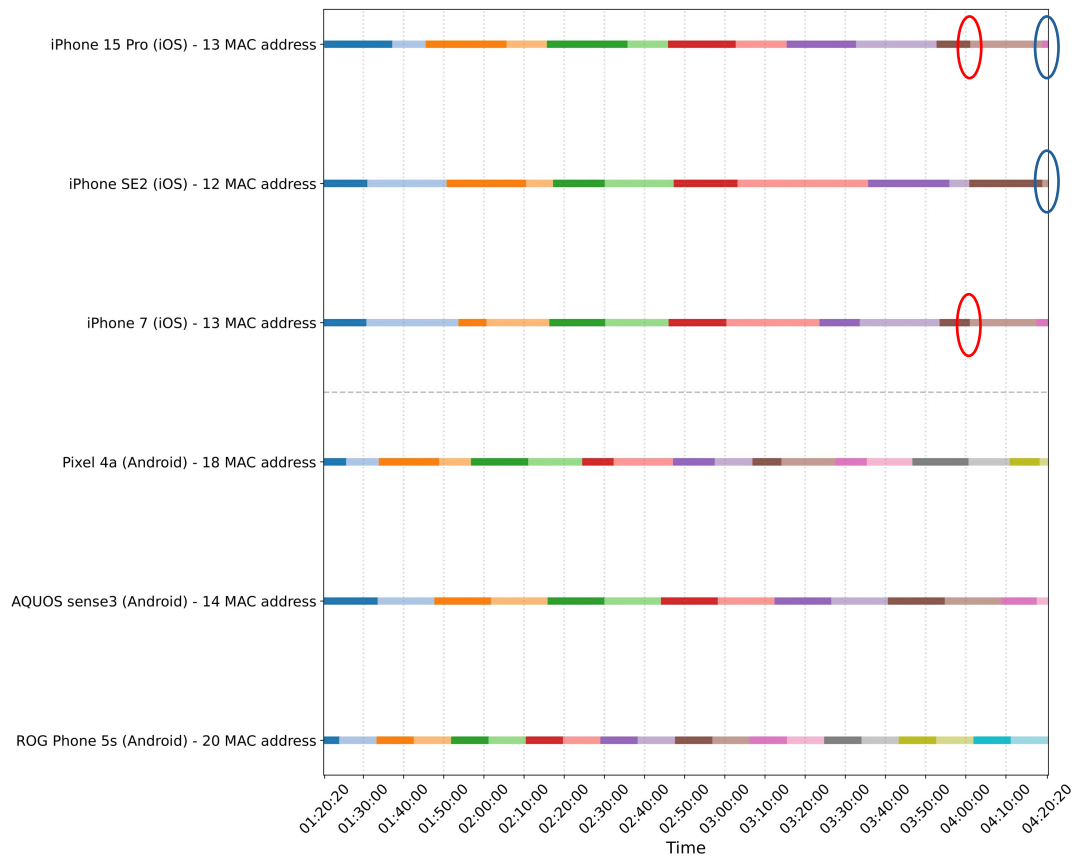


図 11: Find My パケット/0xFEf3 パケットの MAC アドレスランダム化の様子

箇所を示す。実行結果を見ると、どちらの OS も非同期的に MAC アドレスが変化している様子が確認できる。MAC アドレスランダム化が iPhone では 3 台で計 35 回、Android では 3 台で計 49 回発生していた。このうち、MAC アドレスの変化するタイミングが重複したのは iPhone で 2 箇所、Android スマートフォンで 0 箇所である。

次に、MAC アドレスランダム化の発生確率が台数が増えるごとにどのように変化するかを確認する目的で、実験結果で得られた値から Find My パケットの MAC アドレスの変化タイミングが重複する確率のシミュレーションを行った。Find My パケットはおよそ 15 分に一回変化する。また、Find My パケットは送信間隔が約 2 秒であり、5 秒以内に Find My パケットの受信に成功していることが多い。このことから、MAC アドレスが 15 分丁度で変化し、出現時間が 5 秒までのずれで重複した場合 MAC アドレスが同時に変化したとみなすという仮定のもと、MAC アド

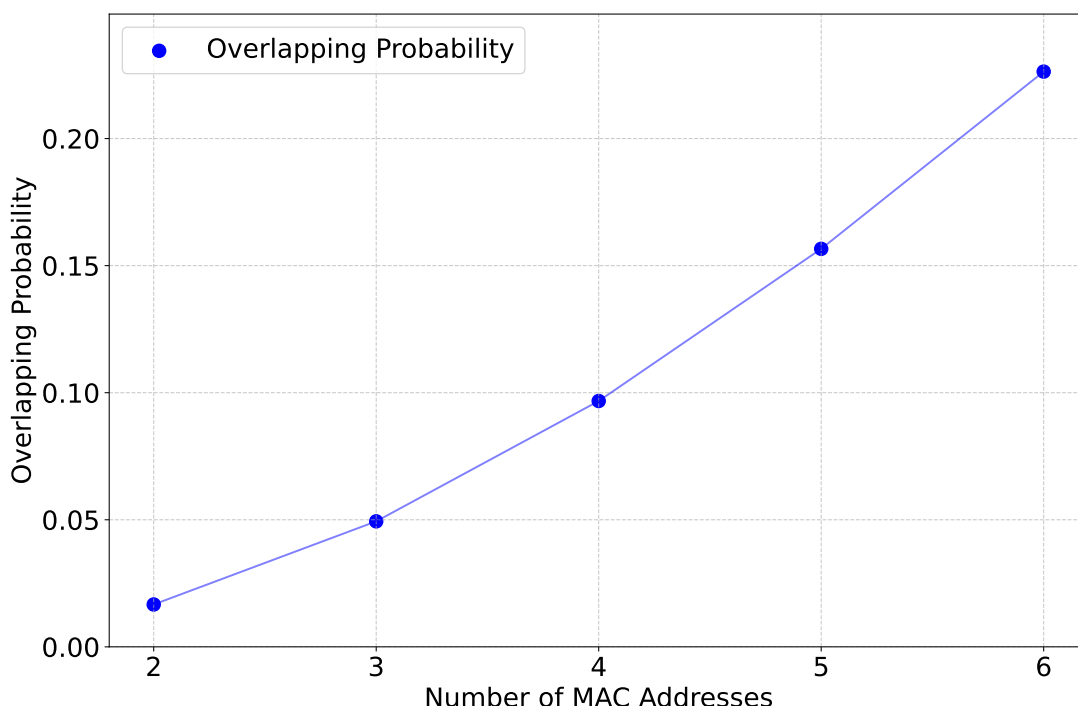
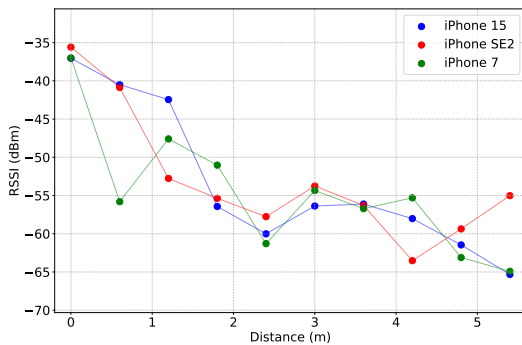


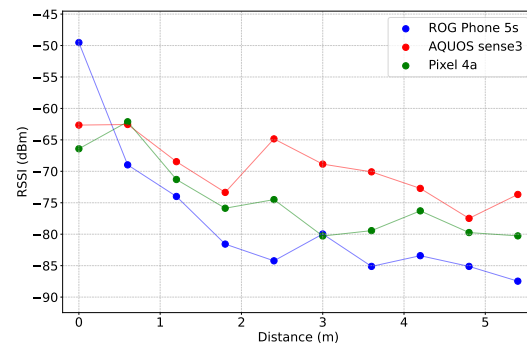
図 12: MAC アドレス数に対する Find My パケットの MAC アドレス変化タイミングが重複する確率に関するシミュレーション結果

レスの数を 2~6 まで増やした時, 2 個以上の MAC アドレスの変化タイミングが重複する確率を総当たりで計算した. 計算結果を図 12 に示す. MAC アドレス数に対して重複確率が単調増加する様子が確認できる. MAC アドレスが 6 台の時点で重複確率はおおよそ 22.6%であった. なお, 0xFEf3 パケットは変化期間はおおよそ 10 分と Find My パケットよりも短い, 送信間隔が 1 秒未満なため, Find My パケットの重複確率よりも小さくなる. 今回実施したシミュレーションは, 簡単化のため Find My パケットの出現時間と重複の条件を固定しているが, 実際には大きく変化する値であり, 重複確率はシミュレーション結果よりも高くなりうる. 加えて, バスの乗車人数が 6 人以上の場面は頻繁に発生するため, MAC アドレスの変化タイミングが重複する確率は更に上がりうる.

アドレスキャリーオーバーアルゴリズムの紐付け候補を絞り込む際, MAC アドレスランダム化が複数のスマートフォンで発生している場合でも, 他の値を使用してスマートフォンの識別が可能であればアドレスキャリーオーバーが可能であると言える. そこで, 次節の実験で RSSI が追加の識別子として機能するか検証す



(a) Find My パケットの平均 RSSI



(b) 0xFEf3 パケットの平均 RSSI

図 13: iPhone・Android スマートフォンの平均 RSSI 比較

る。RSSI は距離に応じて減衰するため、乗客がいる場所によって RSSI が変化する可能性がある。そこで、アドレスキャリーオーバーアルゴリズムで MAC アドレスランダム化のタイミングが被った場合、RSSI で MAC アドレスを紐付けることが有効であるかどうか次節の実験で検討する。

3.2.5 RSSI の距離に応じた減衰に関する検討

RSSI の分析にあたり、iPhone と Android スマートフォンを所持した乗客が平均乗車時間、バス車内の決まった位置に滞在すると仮定し、乗車中に Find My パケットと 0xFEf3 パケットの MAC アドレスランダム化が発生するという条件のもと実験を行った。実験結果について、iPhone 3 台の平均 RSSI を図 13a に、Android スマートフォン 3 台の平均 RSSI を図 13b に、各スマートフォンにおける平均 RSSI の相関係数を表 9 に示す。iPhone/Android スマートフォンともに平均 RSSI に高い負の相関関係があることが確認できる。また、OS ごとで見た時、平均 RSSI に大きな開きはなく、Find My パケットは 0xFEf3 パケットに比べ RSSI が大きいことが確認できる。この結果から、バスの一席単位で MAC アドレスランダム化が同時に発生した場合完全に紐付けることは難しいが、数席離れた位置にある複数スマートフォンの MAC アドレスが同時にランダム化された場合、RSSI で紐付けられると考える。

以上の検証により、Find My パケットと 0xFEf3 パケットの MAC アドレスを対象としたアドレスキャリーオーバーは、100% の完全な紐付けは難しいが、Kawashima らの研究で使用されていた COCOA のパケットと同じように実施することは可能であると考えた。

3.3 既存のバス路線で取得したデータを使った分析

バス車外の MAC アドレスを除去する方法を検討するため、バス路線データを使用して、BLE アドバタイジング・パケットを MAC アドレスごとに観察する。その後、バス車内、バス車外で見られる BLE アドバタイジング・パケットの傾向の仮説について述べる。

3.3.1 既存のバス路線で取得したデータの詳細

岡山県の中鉄バス株式会社が運営しているバス路線の中で、天満屋から国立病院まで運行するバス停およそ 30 箇所、距離およそ 10km の路線において松田講師を代表とする岡山大学の研究チーム (吉村, 林) により BLE アドバタイジング・パケットの収集が行われた。このデータを使った岡山大学の研究チームによる論文投稿はされていないが、実際のバス路線で収集できるデータの分析のために データを提供していただき、本研究で使用した。この BLE アドバタイジング・パケットの収集は岡山大学の学生が Raspberry Pi に BLE ドングルを取り付けたスキャナーでアクティブスキャンにより実施された。スキャナーとそれを操作する MacPC をバッグに入れ、バスの後ろと前に一つずつ席に置かれた。パケット取得ツールには bluepy が使用され、スキャン間隔は 1 秒である。バス OD データの正解値を表 10, 表 11, 表 12 に示す。

表 9: 平均 RSSI の相関係数

OS	機種	相関係数
iOS	iPhone 15	-0.895
	iPhone SE2	-0.773
	iPhone 7	-0.788
Android	Pixel 4a	-0.849
	AQUOS sense3	-0.804
	ROG Phone 5s	-0.813

表 10: 8時 25 分天満屋発のバス OD データ

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
0 天満屋バスセンター				1																							2	
1 NTT 岡山駅前																												
2 岡山駅前(バス)																												
3 岡山駅(東口)														2			1	13										2
4 西川緑道公園前																												
5 岡ビル前																												
6 跨線橋東																												
7 済生会																					1							
8 清心町																												
9 スポーツセンター前																												
10 学南町																												
11 岡山大学筋																												
12 新野																												
13 武道館口																												
14 津島新道																												
15 西坂入口																												
16 明成学院前																												
17 岡山商科大前																												
18 笹ヶ瀬																												
19 津高営業所妙でん寺前																												
20 中原																												
21 エブライ Okanaka 津高前																												
22 小幸田口																												
23 大岩口																												
24 田益																												
25 薬局前																												
26 国立病院																												

表 11: 14時30分天満屋発のバス OD データ

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
0 天満屋バスセンター			1											1													2
1 NTT 岡山駅前																											
2 岡山駅前(バス)																											
3 岡山駅(東口)									1		1		1			2					1			1			2
4 西川緑道公園前																								1			
5 岡ビル前																											
6 跨線橋東																											
7 済生会																	2										1
8 清心町																											
9 スポーツセンター前																											
10 学南町																											
11 岡山大学筋																											
12 新野																											
13 武道館口																											
14 津島新道																							1				1
15 西坂入口																											
16 明成学院前																											
17 岡山商科大前																											
18 笹ヶ瀬																											
19 津高営業所妙でん寺前																											
20 中原																											
21 エブライ Okanaka 津高前																											
22 小幸田口																											
23 大岩口																											
24 田益																											
25 薬局前																											
26 国立病院																											

表 12: 17時37分天満屋発のバス OD データ

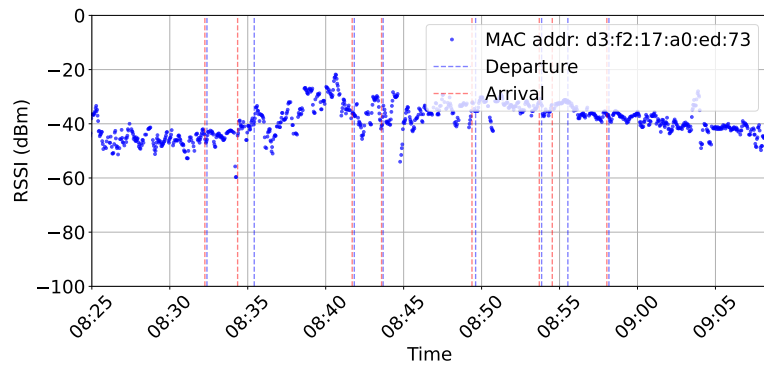
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
0 天満屋バスセンター			1	3				1						1	1			2			1		1	1	1		2
1 NTT 岡山駅前			1	1													1										
2 岡山駅前(バス)																			1								
3 岡山駅(東口)								1									5	1		2		1	3				
4 西川緑道公園前																											
5 岡ビル前																											
6 跨線橋東														1							1						
7 済生会																					1						
8 清心町																											
9 スポーツセンター前																											
10 学南町																											
11 岡山大学筋																											
12 新野																											
13 武道館口																											
14 津島新道																											
15 西坂入口																											
16 明成学院前																											
17 岡山商科大前																											
18 笹ヶ瀬																											
19 津高営業所妙でん寺前																											
20 中原																											
21 エブライ Okanaka 津高前																											
22 小幸田口																											
23 大岩口																											
24 田益																											
25 薬局前																											
26 国立病院																											

収集した BLE アドバイジング・パケットの RSSI について、どのような傾向を示す RSSI があるか、事前に予想していたような傾向を示す MAC アドレスが存在するか確認する目的で、全ての Find My パケットと 0xFE3F パケットを確認した。MAC アドレスが全便を通して変化せず、RSSI が大きい Find My パケットの RSSI について、図 14a に 8 時 25 分発の便で確認できたもの、図 14b に 14 時 30 分発の便で確認できたもの、図 14c に 17 時 37 分発の便で確認できたものを示す。なお、青色の点線は停車したバス停における発車時刻を示し、赤色の点線は停車したバス停における停車時刻を示す。RSSI は分散が大きくそのままデータを表示した場合変化が分かりにくいいため、以降の RSSI に関する図は BLE アドバイジング・パケットとその前後 5 秒間で出現した BLE アドバイジング・パケットで移動平均を取った。図 14a では 8 時 40 分、9 時 5 分頃に、図 14c では 18 時 10 分頃に RSSI が他の時刻に比べ一時的に上昇している箇所が見られる。対して、図 14b ではそのような変化は見られない。BLE アドバイジング・パケットの収集者に確認したところ、スキャナーとバッグに入れた MacPC からパケットが送信されていたこと、8 時 25 分発の便と 17 時 37 分発の便ではバッグを動かしていた可能性があるとのことであった。

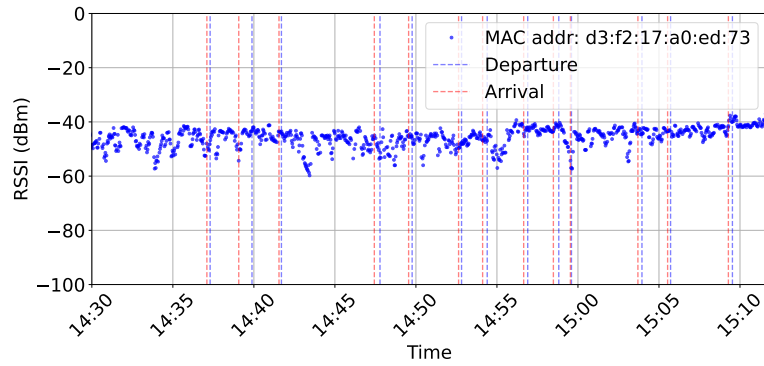
バス停の停車時刻と発車時刻をそれぞれ赤点線と青点線で示し、事前に予想したバスがバス停に停車している間のみ出現している MAC アドレス、バス停車区間とその前後で、RSSI に他の BLE アドバイジング・パケットに見られないような特徴を持つ MAC アドレス、一度定期的に BLE アドバイジング・パケットを送信し、その後一定期間消失したのち再度出現するような動き（間欠的な動き）をする MAC アドレスが存在するかも着目した。このような BLE アドバイジング・パケットの時間変化の動き（パターン）について、いくつか特徴的なパターンを発見した。

一つ目の特徴的なパターンを図 15 に示す。このパターンは一瞬だけ BLE アドバイジング・パケットが出現し、消失している。BLE アドバイジング・パケットの取得間隔が 1 秒であることを考慮すると、一度しか発生していないようなパターンはバスがパケットを発しているスマートフォンを持った歩行者を追い越したとき、バスが BLE アドバイジング・パケットを発しているスマートフォンを持った自動車に追い越されたときなどが考えられる。

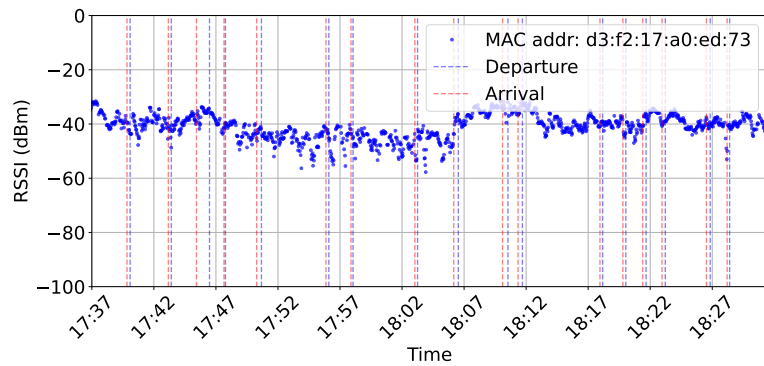
二つ目の特徴的なパターンを図 16 に示す。このパターンは数パケット出現してしばらく消失し、再度出現している。これはバス車内の障害物等によりバス車内のスマートフォンによる BLE アドバイジング・パケットの到達が一定期間失



(a) 8時25分発の便で確認されたMACアドレスが変化していないパターン



(b) 14時30分発の便で確認されたMACアドレスが変化していないパターン



(c) 17時37分発の便で確認されたMACアドレスが変化していないパターン

図 14: 便ごとのMACアドレスが変化していないパターン

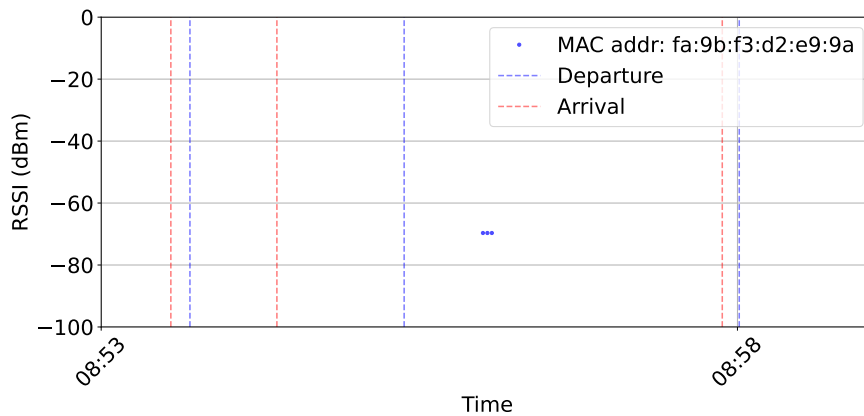


図 15: 非常に短い期間のみ出現が確認された BLE アドバイジング・パケット

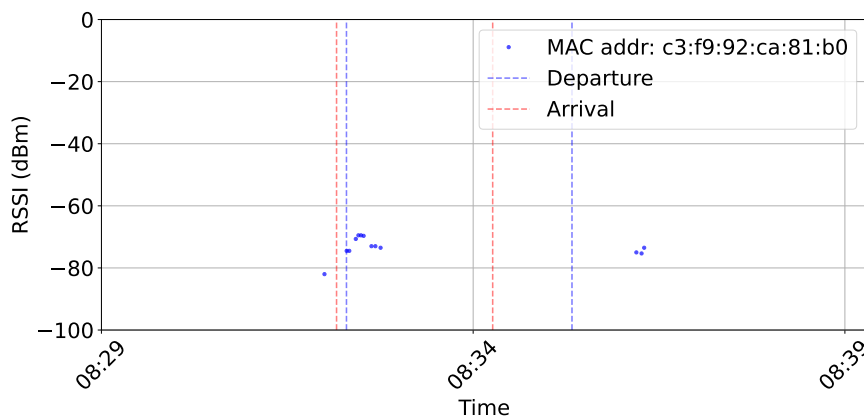


図 16: 一定期間消失している期間がある BLE アドバイジング・パケット

敗したとき、自動車バスを追い越し、信号等でバスに追いつかれるときなどが考えられる。

三つ目の特徴的なパターンを図 17 に示す。このパターンはバス停で停車している間でのみ出現したものである。これはバス停で待っており、乗車しなかったときなどが考えられる。

四つ目の特徴的なパターンを図 18 に示す。このパターンはバスがバス停に留まっている時間に RSSI が大きく上昇または下降するパターンである。これはバスに乗車する、バスから降車するときなどが考えられる。

バス車内の BLE アドバイジング・パケットと考えられる図 14a, 14b, 14c は、最初のバス停出発前から最後のバス停到着後まで受信しており、途中で乗車した

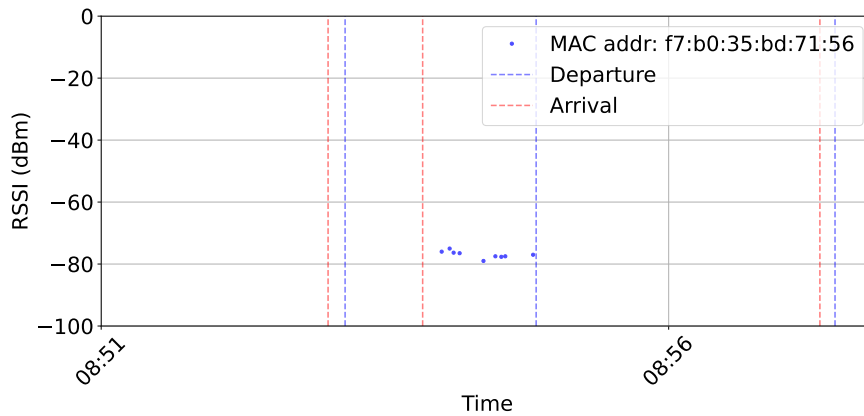


図 17: バス停車期間中のみ出現が確認されたアドバタイジング・パケット

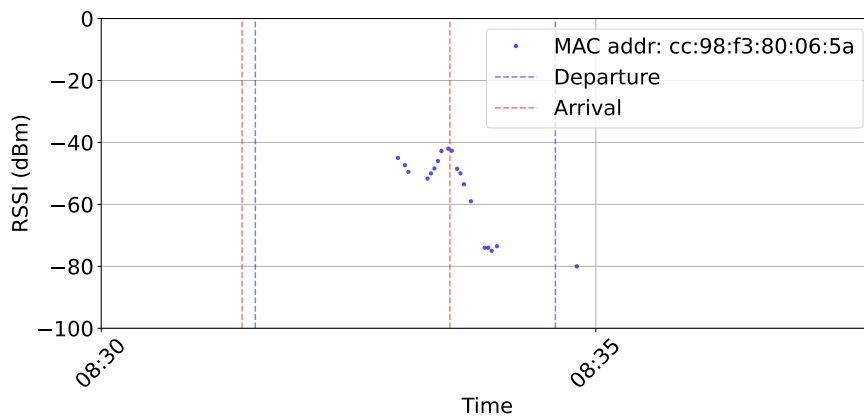


図 18: バス停車期間中のみ出現が確認され、急激な RSSI の上昇が確認されたアドバタイジング・パケット

乗客や途中で降車した乗客の特徴が見えない。そこで、Kawashima らのアドレスキャリーオーバーアルゴリズム 1 を適用し、MAC アドレスの紐付けを行ったのち正解のバス OD データに合致しているものを確認することで、途中で乗車/降車した乗客の特徴について、乗車時/降車時に RSSI が単調に増加/単調に減少するような動きが見られるとする仮説をもとに、アドレスキャリーオーバー後の BLE アドバタイジング・パケットの RSSI の図を確認した。

Kawashima らのアドレスキャリーオーバーにより推定した MAC アドレスの組みのうち、正解バス OD データと一致しているものの一部を図 19 に示す。

紐付けられた MAC アドレスの組みの多くで降車/乗車時に RSSI が上昇/下降

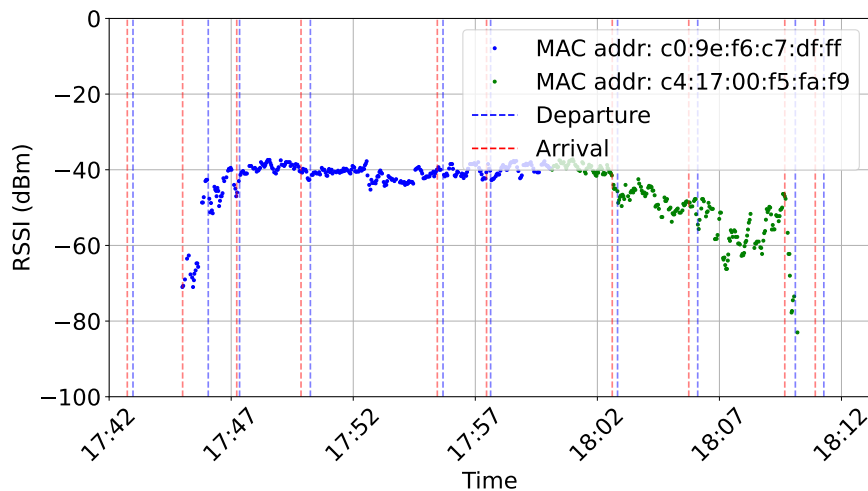


図 19: アドレスキャリーオーバーアルゴリズムで紐付けたアドバタイジング・パケットの組み

している様子が確認できた。また、紐付けに成功したものの RSSI の時間変化についてみたとき、乗車区間は値が安定していることが確認できる。このことから、乗車中は人の動きがほとんど発生しておらず、RSSI の変化が小さいことが考えられる。続いて、不正解だった組み合わせについても分析を行った。不正解の組み合わせには前述の数パケットしか存在しないもの、またバスの乗車/降車地点はバス停停車区間であるものの、正解バス OD データには存在しないバスの乗車・降車の組み合わせと言ったものが複数存在していた。これまでの実データの分析結果から、以下のような予想を立てた。

- ある MAC アドレスが数パケットしか確認できなかった場合、自動車や歩行者が所持しているスマートフォンの可能性がある。
- ある MAC アドレスが長期間消失したのち再出現していた場合、歩行者がバスに追い付く、自動車がバスに追い付かれるといった動きをする人が所持しているスマートフォンの可能性がある。
- ある MAC アドレスがバス停に停止している期間に出現しており、かつ RSSI が単調に上昇している場合、乗車する乗客が所持しているスマートフォンの可能性がある。
- ある MAC アドレスがバス停に停止している期間に消失しており、かつ RSSI

が単調に下降している場合、降車する乗客が所持しているスマートフォンの可能性がある。

- ある MAC アドレスの出現区間が、バスがバス停に停車する時間に収まっていた場合、バス停で留まっている歩行者が所持しているスマートフォンの可能性がある。
- ある MAC アドレスの RSSI の時系列データが乗車区間で安定していた場合、乗客のスマートフォンの可能性がある。

BLE アドバタイジング・パケットが出現する時間や受信間隔において、自動車や歩行者のようなバス車外の BLE アドバタイジング・パケットには、特徴がある可能性がある。このような特徴を使用することで、RSSI のようなバス車内の環境に依存する値を閾値に使用せずに MAC アドレスをバス車外に分類できる可能性がある。そこで、バス車外の動きをパターン化し、定義したいずれかのパターンに当てはまる MAC アドレスをバス車外に分類することを考えた。

4. バス車外のスマートフォン所持者の行動を考慮した バス車内外分類手法

BLE アドバタイジング・パケットを、バス車外のスマートフォン所持者の行動を考慮して設定した閾値によりバス車内とバス車外に分類する手法を提案する。スマートフォン所持者の行動を考慮するために、BLE アドバタイジング・パケットの仕様とバス停付近のバスの速度から閾値を設定した。具体的には、Find My パケットと 0xFE3F のパケットに対して、歩行者や自動車と考えられるようなパターンを BLE アドバタイジング・パケットの仕様とバスの速度を考慮した上で数値的に定義し、これに当てはまる MAC アドレスをバス車外に振り分け、それ以外の MAC アドレスをバス車内に分類する。具体的なアルゴリズムをアルゴリズム 2, 3 に示す。なお、提案手法ではバスの停車時刻を使用するが、これはドアの開閉情報等から取得できるものとする。

3 章から確認できた BLE アドバタイジング・パケットの受信期間や受信間隔の情報から、バス車外のデータのパターンとして以下の三つを定義した。

パターン 1 図 15 のように BLE アドバタイジング・パケットが数パケットのみ出現し消失するような動きをパターン 1 として定義する。このパターンはバスが歩行者を追い越したとき、自動車に追い越されたときのような状況を想定している。

パターン 2 図 17 のように BLE アドバタイジング・パケットの出現がバス停停車期間に発生するような動きをパターン 2 として定義する。このパターンには歩行者が他のバスに乗るためにバス停に滞在しているときのような状況を想定している。

パターン 3 図 16 のように BLE アドバタイジング・パケットの間欠的な動きをパターン 3 として定義する。このパターンには自動車がバスを追い越し、信号等でバスに追いつかれるときのような状況を想定している。

続いて、パターン 1~3 をバス車外の MAC アドレスとして分類するための方法について述べる。パターン 1 について、バスは歩行者と比較して走行速度が大きく、自動車と比較して走行速度が小さい。このことから、数パケットの観測時間で消失した場合、車外の歩行者や自動車内に存在するスマートフォンとしてバス車外に分類する。吉村ら [20] によると、Find My パケットを 24 時間取得し続けたと

アルゴリズム 2 BLE アドバタイジング・パケットの MAC アドレス車内外判定

Require: *mac_list*: MAC アドレスのリスト, *data_dict*: MAC アドレスデータ (受信時刻, OS 種別), *bus_timetable*: バス停時刻表

Ensure: *mac_results*: MAC アドレスごとの車内外判定結果の辞書

```
1: mac_results ← {}                                ▷ 結果を格納する辞書を初期化
2: for mac ∈ mac_list do
3:   mac_results[mac] ← “車外”
4:   appearance_time ← GetAppearanceTime(mac)
5:   if appearance_time ≥ 60 秒 then                ▷ パターン 1 の判定
6:     detection_period ← GetDetectionPeriod(mac)
7:     is_at_stop ← CheckBusStopTime(detection_period, bus_timetable)
8:     if not is_at_stop then                        ▷ パターン 2 の判定
9:       os_type ← GetOSType(mac)
10:      if os_type = Android then
11:        mac_results[mac] ← “車内”
12:      else if os_type = iOS then
13:        packet_intervals ← GetPacketIntervals(mac)
14:        if median(packet_intervals) ≤ 15 秒
           ∧ max(packet_intervals) ≥ 15 秒 then
15:          mac_results[mac] ← “車内” ▷ パターン 3 の判定 (iOS のみ)
16:        end if
17:      end if
18:    end if
19:  end if
20: end for
21: return mac_results
```

きの最小出現時間は1分6秒であった。このことから、1分以内にMACアドレスがランダム化により変更されるとは考えづらい。また、出現時間の閾値を短くすると、車外スマートフォンを車内スマートフォンとしてカウントしてしまう間違いが多く発生する。このことから、Find My パケットの最低出現時間を基に1分以内に消失したMACアドレスは車外データであるとした。対して、AndroidのBLEアドバタイジング・パケットは出現時間の最低値はFind Myパケットより

アルゴリズム 3 バス停停車時刻判定

Require: detection_period: 検出時刻, bus_timetable: バス停時刻表 (到着時刻, 出発時刻のペア)

Ensure: is_at_stop: バス停での停車時間内かどうかの真偽値

```
1: is_at_stop ← False
2: for (arrival_time, departure_time) ∈ bus_timetable do
3:   window_start ← arrival_time - 10 秒
4:   window_end ← departure_time + 10 秒
5:   if window_start ≤ detection_period ≤ window_end then
6:     is_at_stop ← True
7:     break
8:   end if
9: end for
10: return is_at_stop
```

も長かった。こちらも閾値を大きくしすぎると車外スマートフォンを車内スマートフォンとしてカウントしてしまう間違いが多く発生する可能性があることから、1 分間以内に消失した MAC アドレスは車外データとした。

パターン 2 について、バスはバス停に到着し、発車するまで必ず静止し続ける。これに対し、バス停には当該バスの乗客の他に、他のバスを利用する人、周辺を歩いている人等が存在する。このようなバス停付近に滞在し続ける人は、バス停で止まっている時のみ確認でき、バス停到着前、バス停発車後には確認できなくなるような BLE アドバタイジング・パケットの挙動が予想できる。そこで、このようなバス停停車期間に確認できた BLE アドバタイジング・パケットに含まれる MAC アドレスは車外のスマートフォンによる BLE アドバタイジング・パケットのものと判断しバス車外に分類する。バスはバス停付近では徐行運転となり、速度は低下すると考えられる。このことから、およそ 10km/h 程度で走行したと仮定 [21] したとき、10 秒間で 28m 程度の速さとなる。スマートフォンがバスから 30m 離れると、事前実験 3.2 から BLE アドバタイジング・パケットが受信できなくなるのに十分な距離であると考え、バス停停車前後の時間として 10 秒間も停車時間に含めるとする。

パターン 3 について、バスは走行中、信号、バス停、渋滞のような場面で停止する。停止前にバスが自動車に追い越されており、停止時点でバスが自動車に追い

付いた場合、このようなパターンが発生すると考えられる。また、停止前にバスが歩行者を追い越しており、停止時に歩行者がバスに追い付いた場合も、このようなパターンが発生すると考えられる。そこで、今回扱うそれぞれのOSの packets 送信頻度を考慮した閾値を用意し、その閾値の時間 packets の送信が行われていなければ、車外のスマートフォンによる packets と判断しバス車外に分類する。3 時間 3 台の iPhone の Find My packets を取得した結果、受信成功間隔の最大値は 15 秒未満であったことから、15 秒以上受信成功間隔が空いた Find My packets の送信は一般的な送信間隔ではない可能性が高い。このことから、Find My packets の送信が 15 秒以上途切れていた場合車外データであるとする。

ここまでの処理が終了した時点で残りの MAC アドレスは車内のスマートフォンによるものであるとする。パターン 1 の閾値は Find My packets の仕様から決定されたものである。パターン 2 の閾値は一般的な徐行の速度と事前に 30m 離れた地点では BLE アドバタイジング・packets が受信できなくなるのに十分な距離という観点から決定され、閾値の調整は実施していない。パターン 3 の閾値は Find My packets の仕様から決定されたものである。すなわち、バスの車内環境に依存した閾値設定は実施されておらず、閾値の調整も不要である。また、取得する BLE アドバタイジング・packets の仕様と RSSI の大きさは OS によって異なっていることが 3.2 からわかる。そこで、OS ごとにアルゴリズムの処理を変更し、Android スマートフォンについてはパターン 3 の動きのみ通常の BLE アドバタイジング・packets でも距離が離れていれば想定しうると考え、バス車外分類アルゴリズムを実行しない。

5. データセット

本章では、4章で述べた車外にあるスマートフォンのMACアドレスを正しくバス車内外に分類できるか評価するために、正解ラベルがついた任意の車内外のパターンが含まれたBLEアドバタイジング・パケットのデータセットを作成する。5.1節で正解ラベルが付いたバス車内外に存在するスマートフォンのBLEアドバタイジング・パケットを再現するデータセットの内容について説明する。5.2節で奈良先端大の環境を使用して実施した実験と、実験の結果得られたデータセットのパターンごとの評価について述べる。5.3節で実際の運行で使用されているバスを使用して実施した実験と、実験の結果得られたデータセットのパターンごとの評価について述べる。

5.1 データセットの概要

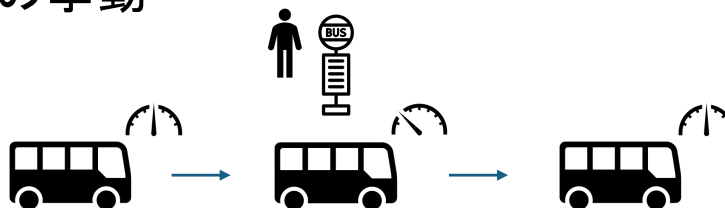
データセットには、バス車内外のスマートフォンを所持した人の動きを想定したRSSIの時系列データが含まれるようにした。データセットの形式を表13に示す。MACアドレス、RSSI、Androidスマートフォン用のService DataとUUID、iPhone用のManufacturer DataとCompany ID、snifferのID、受信日時が含まれている。AndroidスマートフォンはManufacturer DataとCompany IDの部分はNaNに、iPhoneは基本的にService DataとUUIDの部分はNaNになっている。

実験はバス、バスを模した部屋が静止している状態で実施する。また、バス周辺環境は自動車が走行するのに十分な場所を確保することが難しいことから、歩行者、自動車、乗客いずれも人が移動することによってイベントを発生させる。このとき、歩行者の場合もバスが動いているため、相対的には自動車のような動きをすることを考える必要がある。例として、バス停に滞在している歩行者のイベントの発生方法を図20に示す。通常は、バスがバス停まで走り、バス停で停止し、その後出発時間にバス停を出発するという動きをする。これを人で表現するとき、本研究では人がバスまで走り、バスの横で停止し、その後出発時間にバス停から走るという動きをすることとした。

表 13: データセットの形式

No.	MAC addr	RSSI	service_data	uuid	manufacturer_data	company_id	sniffer_id	arrival_time_jst
1	40:67:bf:de:07:9d	-42	NaN	NaN	1005421c3dd155	0x004c	0	14:05:00.021445
2	41:99:ee:d7:00:88	-49	NaN	0xfef3	NaN	NaN	0	14:05:00.021859
3	40:67:bf:de:07:9d	-76	NaN	NaN	12020000	NaN	0	14:05:00.022185
...								

実際の挙動



実験の挙動

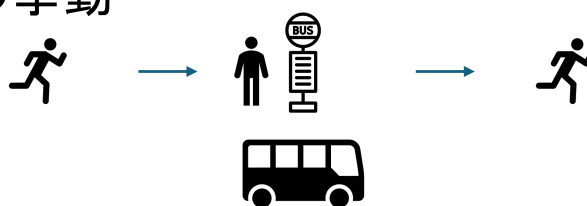


図 20: バス停に滞在する歩行者のイベントの発生方法

バス車内にスニファーを設置し、歩行者、自動車、乗客のイベントを発生させる。BLE アドバタイジング・パケットは状況によってスニファーから距離が離れていても受信できる場合や本来ならば受信できるはずの距離でも障害物等により受信に失敗する場合がある。よって、「見えない場所」としている部分は、その場所まで進むとスニファーの受信範囲から外れる距離の場所と定義し、それ以降 BLE アドバタイジング・パケットを受信しても破棄する。データセットを作成する目的で発生させたイベントについて以下に示す。

歩行者

- バスが停まっている時に横の歩道を通り過ぎる歩行者 (歩行者 1) : バスの横を歩いて通り過ぎてそのまま見えない場所まで移動
- バスが停まっている時にバスの前を横切る歩行者 (歩行者 2) : バスの前を歩いて通り過ぎてそのまま見えない場所まで移動
- バス停に留まっている歩行者 (歩行者 3) : バスが停車している間だけバスの真横に存在

自動車

- バスを追い越す自動車 (自動車 1) : バスの横を大きい速度で通り過ぎてそのまま見えない場所まで移動
- バスを追い越したのち, 信号等で再度合流する自動車 (自動車 2) : バスの横を大きい速度で通り過ぎて見えない場所まで移動したのち, バスの横に大きい速度で移動
- 並走する自動車 (自動車 3) : バスと同じ速度で走行

乗客

- 料金箱での両替 (乗客 1, 2) : バスが停車している時間にバス車内の料金箱に移動し, その後席に座る
- 席の移動 (乗客 1, 2, 3) : バスが停車している時間に席を移動
- 乗車 (乗客 2, 3) : バスがバス停に停まっている時間にバスに乗車
- 降車 (乗客 1, 2, 3) : バスがバス停に停まっている時間にバスを降車

歩行者, 自動車のパターンは無数に存在する. 発生させたイベントは相対速度, BLE アドバタイジング・パケットを受信する間隔をもとにこの無数の動きを抽象化した時の網羅性を考慮して選択した. 加えて, 提案で立てた仮説を検証するために必要なデータという点も満たしている. データセットの BLE アドバタイジング・パケットは iOS で Nearby パケットと Find My パケット, Android で 0xFE33 のパケットで構成されている. Nearby パケットは Find My に比べて取得間隔が小さい. そこで, 発生させたイベントに対して高密度な RSSI の時系列データにどのような違いが現れるか検証するために取得している.

5.2 データセット収集実験 A

データセットの作成はスマートフォンを所持した被験者がバス車内外でスケジュール通りに動き, それぞれのスマートフォンから送信されている BLE アドバタイジング・パケットをバス車内のスニファァで取得することで実現する. 以降

で具体的なイベントを発生させる方法と準備した環境を、データセットの作成実験をもとに説明する。なお、データセットの作成はまず奈良先端科学技術大学院大学の S3 教室と呼ばれる 7m × 7m × 3m 程度の教室と、その部屋周辺の道を使用した実験 (実験 A) を実施し、評価に使用するためのデータセットの作成を行う。

5.2.1 実験 A の方法

実験 A は被験者 3 人がそれぞれスマートフォンを所持して実施された。BLE アドバタイジング・パケットの取得はノルディック・セミコンダクター社製の nRF52840[22] を BLE アドバタイジング・パケットを収集するためのスニファートとして MacBook Pro に取り付けて実施した。BLE アドバタイジング・パケットをパッシブスキャンで収集し、収集したパケットを pcap 形式で保存する。使用したスマートフォンは iPhone 3 台、Android スマートフォン 3 台を使って実施した。それぞれの OS のバージョンとスマートフォン名を表 14 に示す。実験のために、それぞれのスマートフォンにはアカウントを作成し iPhone には iCloud アカウント、Android スマートフォンには Google アカウントでログインしている。

次に、実験 A で発生させるイベントのスケジュールについて説明する。想定するバス車内外のスマートフォン所持者のイベントは実際に自動車バスを追い越す、バスを動かし歩行者を追い越す等、実機を使用することで表現することが理想である。しかし、このような環境を用意するために、広い空間を用意する必要があり、実施が難しい。そこで、人の動きで再現できる方法を考慮した。具体的には、バスによる歩行者の追い越しや、自動車によるバスの追い越しは相対的に大きい

表 14: 大学内の実験で使用した iPhone/Android スマートフォン

OS	機種名	製造者	OS バージョン
iOS	iPhone SE2	Apple	18.1.1
	iPhone 15 Pro		17.6.0
			17.4.1
Android	Pixel 4a	Google	13

速度の動きと考え、被験者が走って移動することにより再現する。

実験 A で再現するイベントの具体的な発生方法を以下に示す。

バス停に留まっている歩行者 BLE アドバイジング・パッケージが届かない地点からバス停到着時刻直前に被験者がバスから離れた場所からバス停に向かって走り始め、バスの入り口前で停止する。バス停発車時刻直後に被験者がバス前から走り始め、BLE アドバイジング・パッケージが見えなくなる地点まで移動する。

バスが停まっている時に横の歩道を通り過ぎる歩行者 BLE アドバイジング・パッケージが届かない地点からバスの左横を歩いて BLE アドバイジング・パッケージが見えなくなるまで移動する。

バスが停まっている時にバスの前を横切る歩行者 BLE アドバイジング・パッケージが届かない地点からバスの前を歩いて BLE アドバイジング・パッケージが見えなくなるまで移動する。

バスを追い越す自動車 BLE アドバイジング・パッケージが届かない地点からバスの左横を走って BLE アドバイジング・パッケージが見えなくなるまで移動する。

バスを追い越したのち、信号等で再度合流する自動車 BLE アドバイジング・パッケージが届かない地点からバスの左横を走って BLE アドバイジング・パッケージが見えなくなるまで移動する。その後、しばらく時間をおいたのち、バスの前方側から、BLE アドバイジング・パッケージが届かない地点からバスの前まで走って移動する。その後しばらく停止し、BLE アドバイジング・パッケージが届かない地点まで走って移動する。

並走する自動車 BLE アドバイジング・パッケージが届かない地点からバスの左横を走ってバスの右横に移動する。その後、バスが走行している間バスの両端を歩いて往復する。バスがバス停で停止したタイミングで BLE アドバイジング・パッケージが届かない地点まで走って移動する。

以上で述べたパターンに加え、バス車内で発生しうる動きについても考慮する必要がある。実験 A では 3 人の乗客が存在するようした。バス車内の乗客の動きについて、バス車内では走行中に乗客の移動は発生しないと仮定して考える。バス停、信号で停車している間は料金箱で両替する、席を移動する等の乗客の動きが

発生しうる。そこで、意図的に信号待ちとバス停車期間を設定し、料金箱の移動や席の移動を発生させ、バス車内の乗客の移動がどのような RSSI の時系列変化の動きが現れるか確認する。

イベントの発生方法を踏まえて、イベントのスケジュールを表 15, 16, 17 に示す。スマートフォンを所持した被験者がイベントを発生させるために用意したスケジュール通りに動く。スケジュール通りに動くことで、バス停車中、バス走行中それぞれ起こり得るイベントを実際にバスが運行されている環境での状況と矛盾なく取得できる。被験者は iPhone/Android スマートフォンを一台ずつ所持している。スケジュールで示したスマートフォンとの対応を表 18 に示す。OS ごとに同じ動作を行うことにより、BLE アドバタイジング・パケットの見え方について比較と検証ができるようにした。

5.2.2 実験 A の環境

実験 A の実験環境について、環境全体の俯瞰図を図 21 に、バス車内を模した部屋の俯瞰図を図 22 に示す。バス車内を見立てた教室の右側を車道側、左側を歩道側としている。バスと自動車を通る場所の間の距離は 7m、バスの前後の距離は BLE アドバタイジング・パケットが自然と見えなくようにバスの後ろに 5.15m、バスの前に 30m 取った。なお、バス車内を見立てた教室は側面右側がガラス張り

表 15: iPhone SE2 & Pixel 4a (スマートフォン 1) のスケジュール表 (実験 A)

時刻	バス停の発着	行動開始時にいる場所	データパターン	行動内容
0:00:00	01 発			
0:04:00			歩道側バス後方	・歩道側バス後方から車道側バス前へ走る
0:05:55		データパターン: 自動車 1 (パターン 3)	車道側バス前	・車道側バス前からバス前まで走り、バス前で 0:07:00 まで止まる
0:07:00			バス前	・バス前から車道側バス前まで走る
0:12:00	02 着	データパターン: 歩行者 1 (パターン 1)	歩道側バス後方	・歩道側バス後方から車道側バス前まで歩く
0:13:00	02 発			
0:13:20		データパターン: 自動車 2 (パターン 1)	歩道側バス後方	・歩道側バス後方から車道側バス前へ走る
0:14:00	03 着			
0:15:00	03 発			
0:17:00			歩道側バス後方	・歩道側バス後方からバスの右横まで走り、バスの右横で 0:20:00 まで止まる
0:20:00			バスの右横	・0:35:00 までバスの右前と後ろを歩いて往復する
0:35:00		データパターン: 自動車 3	バスの右横	・バスの右横で 0:36:00 まで止まる
0:36:00			バスの右横	・0:36:50 まで歩いてバスの右前と後ろを歩いて往復する
0:36:50			バスの右横	・バスの右横から車道側バス前へ走る
0:37:00	04 着			
0:38:00	04 発			
0:40:50			歩道側バス後方	・歩道側バス後方からバス左横まで走り、バスの左横で 0:42:00 まで止まる
0:41:00	05 着	データパターン: 歩行者 3 (パターン 2)		
0:42:00	05 発		バスの左横	・バス左横から車道側バス前へ走る

の壁となっており、壁の下半分は実際のバス車内と同じになるように机を壁がわりに置いている。車内の座席は 60cm の間隔にし、教室の中で設置できる最大数まで椅子を設置した。座席は 4 列あり、そのうちの最前列・最後列・二列目の右側座席にスニファーを取り付けた。検証対象のスマートフォンがパケットを発しているかどうか確認する目的で、リュックサックの中にスマートフォンと Raspberry Pi4 Model B と TP-Link の Bluetooth 4.0 ナノ USB アダプタ UB400 を BLE ドングルとして取り付けたスキャナーを入れた。実験の被験者はこのリュックサック

表 16: iPhone 15 Pro & Pixel 4a (スマートフォン 2) のスケジュール表 (実験 A)

時刻	バス停の発着	行動開始時にいる場所	データパターン	行動内容
0:11:50		データパターン: 乗客 2	歩道側バス後方	・歩道側バス後方からバス左横へ走る
0:12:00	02 着		バス左横	・バスに入り、真ん中の席に移動
0:13:00	02 発			
0:14:00			バス車内	・真ん中の席から料金箱へ移動し、10 秒停止、左側最前席に座る
0:14:00	03 着			
0:15:00	03 発			
0:18:00			バス車内	・左側最前席から真ん中の席に移動
0:35:00			バス車内	・真ん中の席から最前席に移動
0:37:00	04 着		バス車内	・バスを降り、歩道側バス後方まで歩く
0:38:00	04 発			
0:40:50		データパターン: 乗客 3	歩道側バス後方	・歩道側バス後方からバス左横まで走る
0:41:00	05 着		バス左横	・バスへ入り、最前席に座る
0:42:00	05 発			
0:44:00			バス車内	・最前席から最後席に移動
0:46:00			バス車内	・最後席から最前席に移動
0:46:00			バス車内	・バスを降り、歩道側バス後方まで歩く
0:47:00				

表 17: iPhone 15 Pro & Pixel 4a (スマートフォン 3) のスケジュール表 (実験 A)

時刻	バス停の発着	行動開始時にいる場所	データパターン	行動内容
0:00:00	01 発	データパターン: 乗客 1	バス車内	・バス車内の真ん中の席に乗った状態で始める
0:06:00			バス車内	・真ん中の席から料金箱へ移動し、10 秒停止後元の席に戻る
0:10:00			バス車内	・真ん中の席から最後席に移動
0:12:00	02 着		バス車内	・最後席から真ん中の席に移動
0:13:00	02 発			
0:14:00	03 着		バス車内	・バスを降り、歩道側バス後方まで歩く
0:18:00		データパターン: 歩行者 2 (パターン 1)	歩道側バス後方	・横断歩道出発点からバス前真ん中まで行き、横断歩道出発点まで戻る

表 18: スマートフォンの対応

Android	iOS	識別名
Pixel 4a	iPhone SE2	iPhone SE2 (スマートフォン 1)
Pixel 4a	iPhone 15 Pro (version: 17.6.0)	iPhone 15 Pro (スマートフォン 2)
Pixel 4a	iPhone 15 Pro (version: 17.4.1)	iPhone 15 Pro (スマートフォン 3)

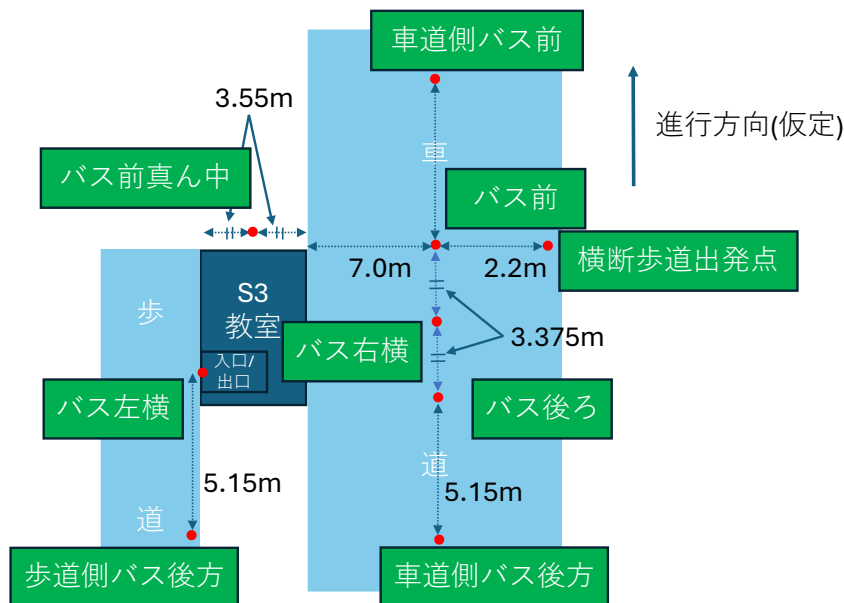


図 21: 実験 A の実験環境

クを持ちスマートフォンを移動させる。パケットキャプチャツールとして、スニフアーは Wireshark4.4.2 を、スキャナーは hcitool version 5.66 と btmon version 5.66 を使用した。

5.2.3 実験 A で取得したデータの分析

取得したデータセットに対して、バス路線データを再現できているか分析した結果を示す。なお、バス車外の Android スマートフォンのアドバタイジング・パケットは受信できていなかったため、iPhone の BLE アドバタイジング・パケットのみを使って分析した。

バス路線データと比較することによるデータの再現性を評価するため、以下の定性的評価基準を設けた。なお、MAC アドレスはランダム化後も同一スマートフォンから送信されているものとして扱う。

バス停での検出 バス停停車中およびその前後で MAC アドレスが検出される場合。これは停留所付近の歩行者を想定した基準である。実データとの比較は図 17 で行う。

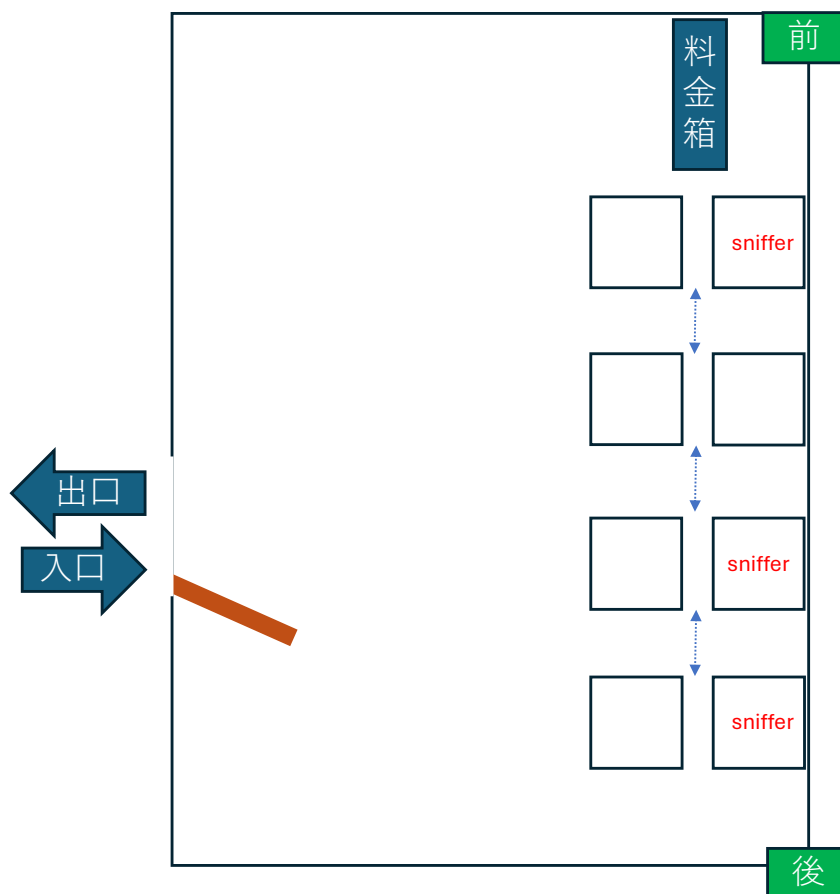


図 22: 実験 A のバス車内を模した部屋内部の俯瞰図

間欠的な検出 同一 MAC アドレスが時間差をおいて再検出される場合. バスと歩行者・自動車が追い越し・追いつきの関係にある状況を想定している. 実データとの比較は図 16 で行う.

一時的な検出 極めて短時間のみ MAC アドレスが検出される場合. バスによる追い越しや横断歩行者等の一時的な接近を想定している. 実データとの比較は図 15 で行う.

乗客パターン 以下を満たす場合を乗客の移動パターンとして評価する. 実データとの比較は図 14a, 14b, 14c, 19 で行う.

- 乗車: バス停停車前後で MAC アドレス出現と RSSI 上昇

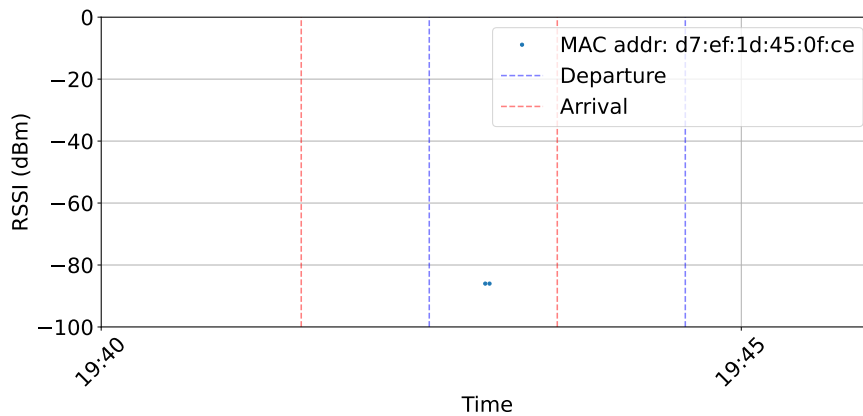


図 23: 実験 A: 自動車の追い越しを模した Find My パケットの RSSI の時系列変化

- 乗車中: RSSI の安定
- 降車: バス停停車前後で MAC アドレス消失と RSSI 減少

バスの並走車は今回実験で発生させる動きから BLE アドバタイジング・パケットの RSSI の分散が大きくなることが予想できる。しかし、実データからは分散の大きさが他と比較して目立って大きく、かつ一定時間出現しているようなものは確認できなかった。そのため、並走車のみ実データと比較することによる定性的な評価は行わない。以降で、取得した BLE アドバタイジング・パケットについて、発生させたイベントごとに MAC アドレス、RSSI、時刻で図示する。三つあるスニファアのうち、最もデータがうまく取得できていた前に設置したスニファアをもとに分析を行う。

バスを追い越す自動車 図 23 に自動車の追い越しを仮定した Find My パケットの RSSI の時系列変化を示す。結果から、1 度のみデータが取得されていることが確認できる。また、事前の予想ではバスに近づくほど RSSI が上昇し、バスから遠ざかるほど RSSI が減少するような動きが見られると考えていた。しかし、実際にはそのような動きは捉えきれていない。

Find My パケットの送信間隔が約 2 秒であることを考え、より送信間隔が小さい Nearby パケットでも RSSI の時系列変化を確認した。図 23 で表示した区間と同じ区間の Nearby パケットデータの RSSI の時系列変化を図 24 に示す。結果から、高頻度で送信されている BLE アドバタイジング・パケットであってもほとんど

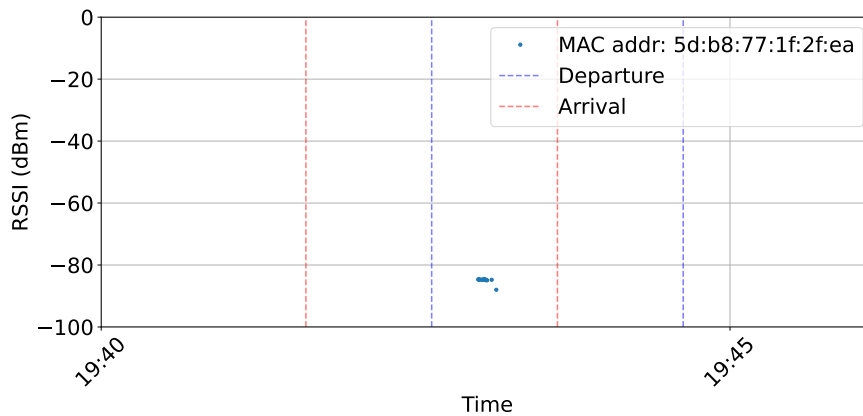


図 24: 実験 A: 自動車の追い越しを模した Nearby パケットの RSSI の時系列変化

ど取得できていないことがわかった。実際の道路幅員がおよそ 3.5m であり、バス車内に設置したスニファーマと隣の車線の自動車の中のスマートフォンの距離が道路幅員と同程度と考えると、バス車内のスキャナと被験者の距離が 7m と大きく離れていたことが原因である可能性がある。一方で、実データを見ると、RSSI が急上昇後に急下降するような値の変化は確認できなかったことから、RSSI でこのような瞬間的な値の変化を捉えにくい可能性があり、その点で実データを適切に再現できていると言える。また、設定した定性的な基準も満たしていると言える。

バスを追い越したのち、信号等で再度合流する自動車 図 25 に自動車の追い越しがあり、その後信号で停車するタイミングで追い越された自動車に追いつき、信号が変わったタイミングで再度距離が離されるという動きを仮定した Find My パケットの RSSI の時系列変化を示す。結果から、一瞬出現したのち消失し、その後時間を置いて再出現するような時間的な動きが確認できる。こちらもバスから遠ざかるような動きをすると RSSI が単調に減少し、バスに近づくような動きをすると RSSI が単調に増加するような動きを想定していたが、そのような動きは見られなかった。実データと比較しても、設定した基準を満たしていると言える。

並走する自動車 図 26 に自動車の並走があり、バスがバス停で停まるタイミングで距離が離されるという動きを仮定した Find My パケットの RSSI の時系列変化を示す。結果から、分散が大きい RSSI の時系列変化であることがわかる。しかし、これは並走を歩いて往復すると表現したことによる可能性もある。加えて、分散が比較的大きい以外の目立った特徴はなく、車内の RSSI の時系列変化と比較した

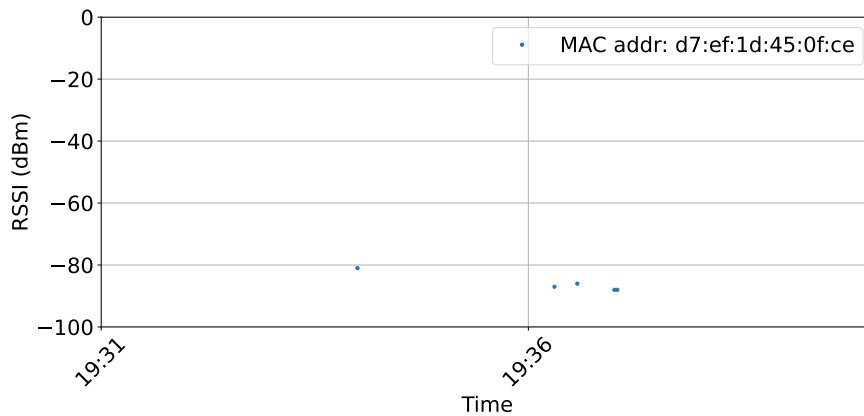


図 25: 実験 A: 自動車の追い越し後, 信号で再度追い付く様子を模した Find My パケットの RSSI の時系列変化

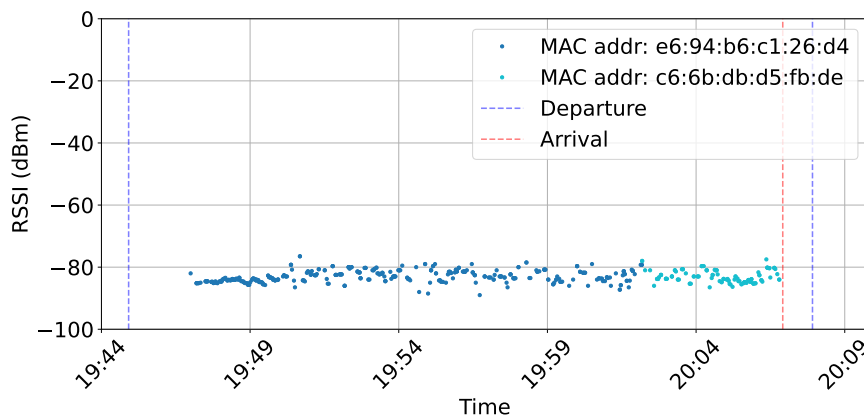


図 26: 実験 A: 並走車を模した Find My パケットの RSSI の時系列変化

ときに車外のスマートフォンからの BLE アドバタイジング・パケットであると判断するのは難しい。

バスが停まっている時にバスの横・前を歩く歩行者 図 27, 28 に歩行者がバス停車中に追い越しを行なったという動きを仮定した Find My パケットの RSSI の時系列変化を示す。図 27 は歩道で追い越しを行い, 図 28 はバスの前に横断歩道があると仮定し, バスの前を通るというシナリオを想定している。結果から, 自動車の追い越し時と似たような動きになっており, 非常に短い間隔で出現していることが確認できた。実データと比較しても, 設定した基準を満たしていると言える。

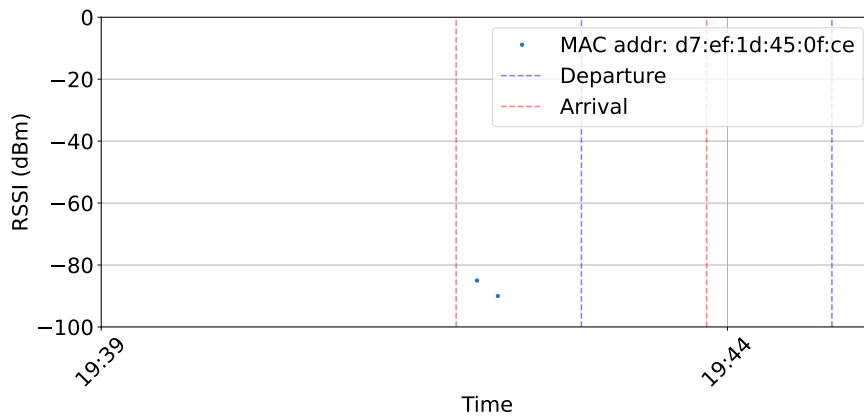


図 27: 実験 A: バス横の歩道を歩く歩行者を模した Find My パケットの RSSI の時系列変化

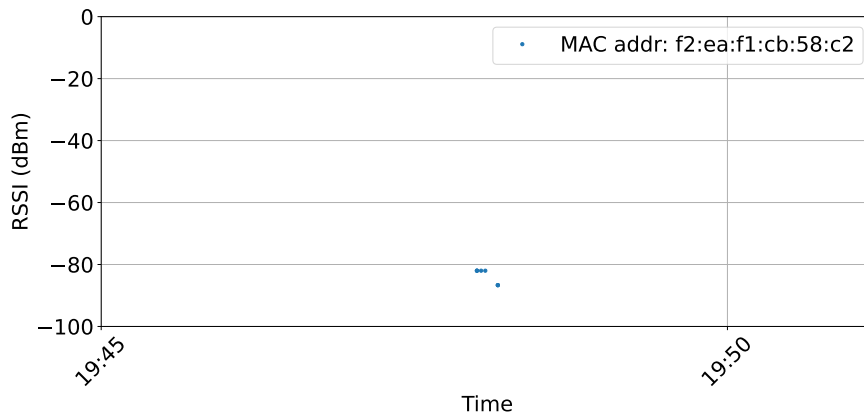


図 28: 実験 A: 横断歩道を歩く歩行者を模した Find My パケットの RSSI の時系列変化

バス停に留まっている歩行者 図 29 にバス停に人がおり、バスに乗らないという動きを仮定した Find My パケットの RSSI の時系列変化を示す。結果から、バスに乗らない場合バス停停車中の期間のみ BLE アドバタイジング・パケットがキャプチャできている様子が確認できる。実データと比較しても、設定した基準を満たしていると言える。

乗客 図 30 に乗客を模した Find My パケットの RSSI の時系列変化を、図 31 に乗客を模した Android スマートフォンの BLE アドバタイジング・パケットの RSSI

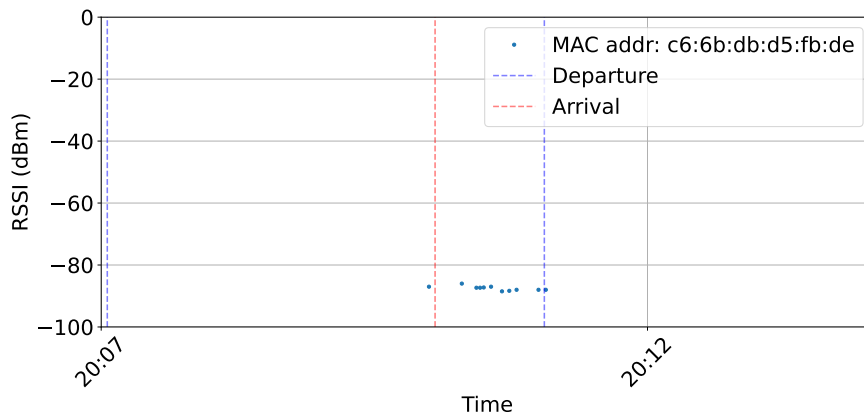


図 29: 実験 A: バス停で待つ人を模した Find My パケットの RSSI の時系列変化

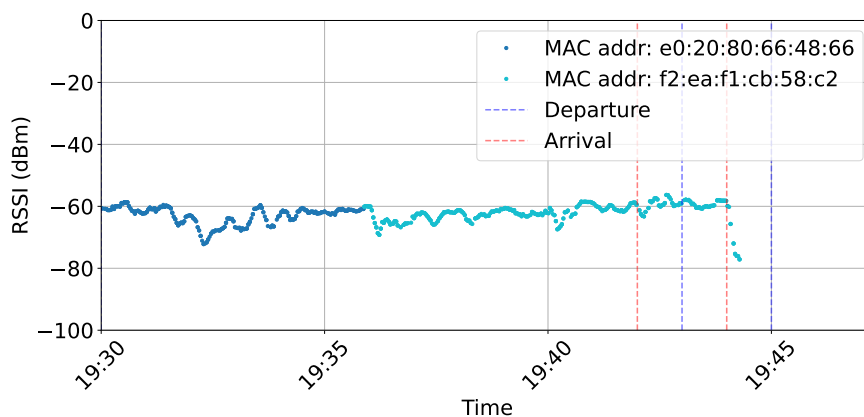


図 30: 実験 A: 乗客 1 の Find My パケットの RSSI の時系列変化

の時系列変化を示す。

乗客は 3 パターン分あるが、今回は以下の二つの動きをしている乗客 1 のデータについて見る。

- バスが信号で停車中に料金箱に両替をしに行く
- バスが信号で停車中に席を移動する

なお、他の乗客も同じような動きを発生させたが、RSSI の時系列変化に大きな違いは見られなかった。まず Find My パケットについて、移動に伴い、RSSI が変動している様子が確認できるものの、BLE アドバタイジング・パケットの送信が途切れる、一瞬消失するといったことは発生しておらず、安定して受信できている。

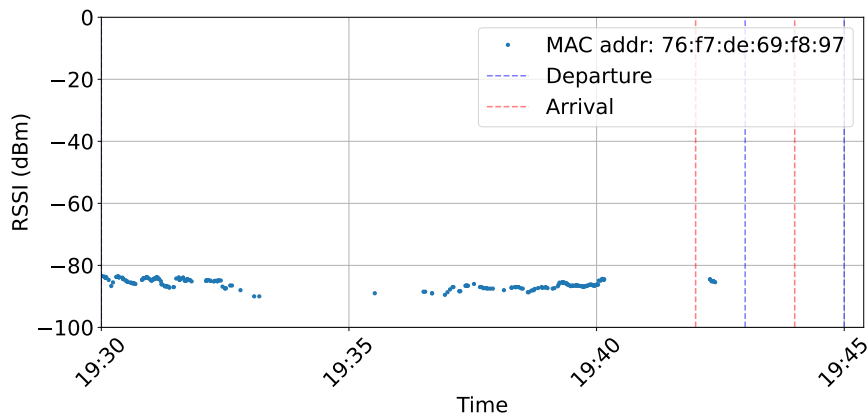


図 31: 実験 A: 乗客 1 の Android の BLE アドバタイジング・パケットの RSSI の時系列変化

このことから、実験 A の実験環境であれば、乗客に何らかの動きが発生した場合であっても、途中で BLE アドバタイジング・パケットがキャプチャできなくなるといったことは発生しない可能性が高い。一方で、今回の実験ではバス車内よりも小さい空間で実施しているため、これが原因となって変動が小さくなっている可能性がある。次に、Android スマートフォンの BLE アドバタイジング・パケットについて、非常に小さい RSSI で遷移している様子が確認できる。Android は元々の RSSI が低いことが 3.2 節から分かっていたため、同じ MAC アドレスの BLE アドバタイジング・パケットを受信した時間の間隔に開きがあった場合でも乗客の可能性があるとと言える。iOS については、実データと比較しても設定した基準を満たしていると言える。乗車中の区間では席の移動に伴って多少の RSSI の上昇・下降傾向が見られたが、これは実データでも同様であり、このような変化はバス車内で起こりうるということ、今回の実験環境ではバス車内で移動が発生したとしても BLE アドバタイジング・パケットの受信が無くなるようなことは発生しないと言える。対して、Android については予想よりも BLE アドバタイジング・パケットが取得できていないことが分かった。バス路線データの中で、アドレスキャリアオーバーアルゴリズムにより Android の BLE アドバタイジング・パケットを紐付けたもののうち、バス OD データの正解と一致しているものの RSSI の時系列変化の例を図 32 に示す。

実験 A で取得した BLE アドバタイジング・パケットと比較して RSSI が大きいことが確認できる。RSSI が実験 A の結果と大きく異なる値を示す理由は不明だ

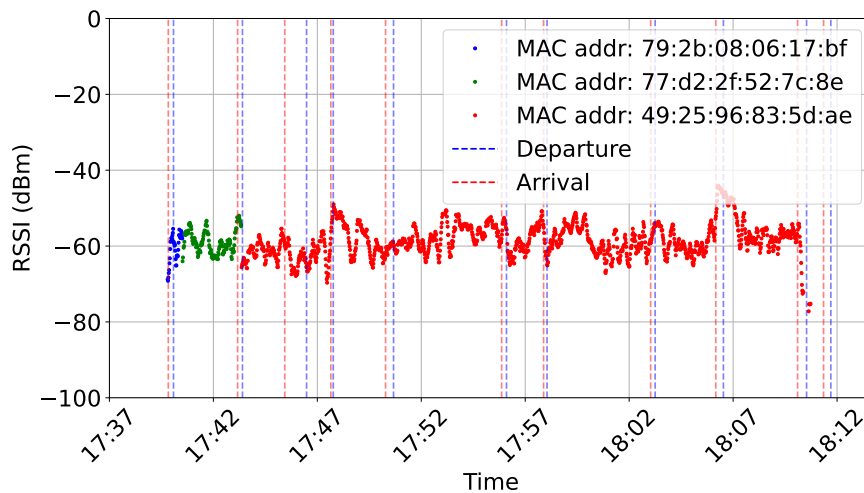


図 32: アドレスキャリーオーバーアルゴリズムで紐付けた Android の BLE アドバタイジング・パケットの組み

が, 0xFE3F3 パケットの RSSI は高い傾向にあるもの, 低い傾向を示すものどちらも存在する。

ここまでのバス車内外データの分析結果から, Android の BLE アドバタイジング・パケットは RSSI が低く, 間欠的な送信の様子が見られる。このことは, Android スマートフォンに対しては間欠的な動きを見せる BLE アドバタイジング・パケットの除去処理を行わないとする提案手法の流れが妥当であると考えられる。

5.3 データセット収集実験 B

5.2.1 節の実験 A をもとに, 実際に使用されているバス会社が保有しているバスを使用した環境で実験 (実験 B) をした。実験 A では被験者は 3 人であったが, 実験 B では 2 人とし, スマートフォン 1 とスマートフォン 2 は同一人物が担当する。なお, スマートフォン 1 とスマートフォン 2 は全て乗客の動きをし, スマートフォン 2 は席の移動や料金箱での両替といったイベントが一切発生しないようにした。このようにすることで, スマートフォン 2 を所持していない場合でも任意の席に置いたままにできるため, どちらのスマートフォンもスケジュール通りに動かすことができる。実験 A と同様, データセットの作成はスマートフォンを所持した被験者がバス車内外でスケジュール通りに動き, それぞれのスマートフォンから送信されている BLE アドバタイジング・パケットをバス車内のスニファーで取

得する。

5.3.1 データセットの概要

5.1 節で述べた実験 A のデータセット同様、データセットの中身には、バス車内外のスマートフォンを所持した人の動きを想定した RSSI の時系列データが含まれるようにした。また、データセットの形式も同様に MAC アドレス、RSSI、Android スマートフォン用の Service Data と UUID、iPhone 用の Manufacturer Data と Company ID、スニファアの ID、受信日時が含まれている。データセットには、バス車外の自動車について 2 種類、乗客について 4 種類の動きをしたデータが含まれている。具体的に発生させたイベントを以下に示す。

自動車

- バスを追い越したのち、信号等で再度合流する自動車 (自動車 1) : バスの横を大きい速度で通り過ぎて見えない場所まで移動したのち、バスの横に大きい速度で移動
- 並走する自動車 (自動車 2) : バスと同じ速度で走行

乗客

- 料金箱での両替 (乗客 4) : バスが停車している時間にバス車内の料金箱に移動し、その後席に座る
- 席の移動 (乗客 1, 3, 4) : バスが停車している時間に席を移動
- 乗車 (乗客 3, 4, 5) : バスがバス停に停まっている時間にバスに乗車
- 降車 (乗客 1, 2, 3, 4, 5) : バスがバス停に停まっている時間にバスを降車

発生させたイベントは乗客と今回提案したアルゴリズムでバス車外と分類できる自動車と車外と分類できない自動車の三つのパターンを用意した。バス車外のイベントについては、提案手法の有効性を示しつつ、正しく分類できないものについての評価も実施できると言う観点からこの二つのイベントを取得した。ま

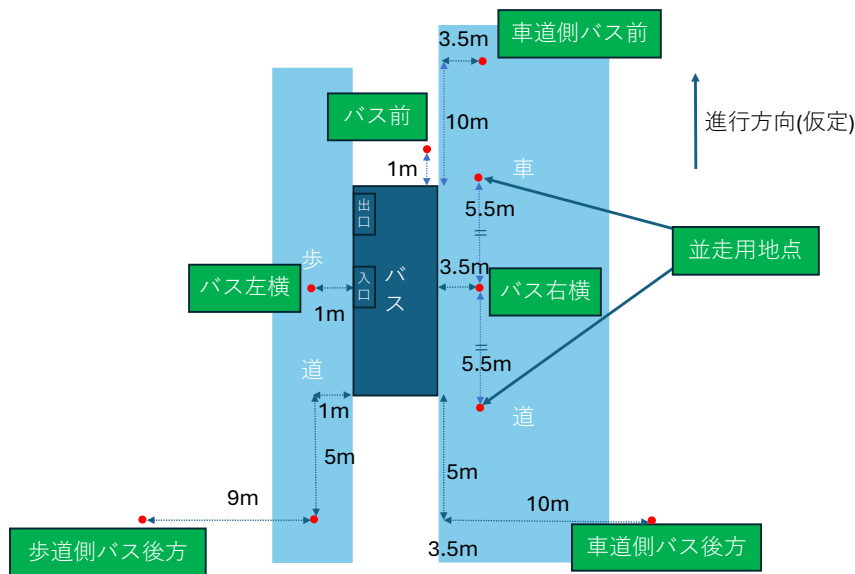


図 33: 実験環境: 環境全体の俯瞰図

た、バス車内の乗客のスマートフォンが一定期間 BLE アドバタイジング・パケットの送信がないような状況が発生しうるかどうかという観点から乗客のイベントを複数取得し、乗客の数も実験 A の 3 人から 5 人に増やした。データセットの作成方法は基本的に実験 A と同様に実施した。対して、実験の環境は異なるものとなっている。そこで、次に実験の環境について述べる。

5.3.2 実験の環境

実験の環境について、環境全体の俯瞰図を図 33、バス車内の俯瞰図を図 34 に示す。

実験場所は奈良交通株式会社に協力していただき、奈良交通株式会社の北大和営業所の敷地の一部とバス一台を借りることで用意した。実験 A 同様、バスの右側を車道側、左側を歩道側としている。バスと自動車を通る場所の間の距離は 3.5m、バスの前後の距離は BLE アドバタイジング・パケットが自然と見えなくなるようにバスの後ろに 5m、バスの前に 10m とり、後ろ側は歩道側・車道側ともに 10m バスから離れた場所から移動を開始するようにした。車内の座席は 70cm から 90cm 程度の距離を空けて並んでいる。バスの出入口は異なる場所を仮定し、真ん中側の扉を入口、前側の扉を出口とした。バス車内の右側の最前席、真ん中の席、最後席の 3 箇所にスニファータを取り付けた。

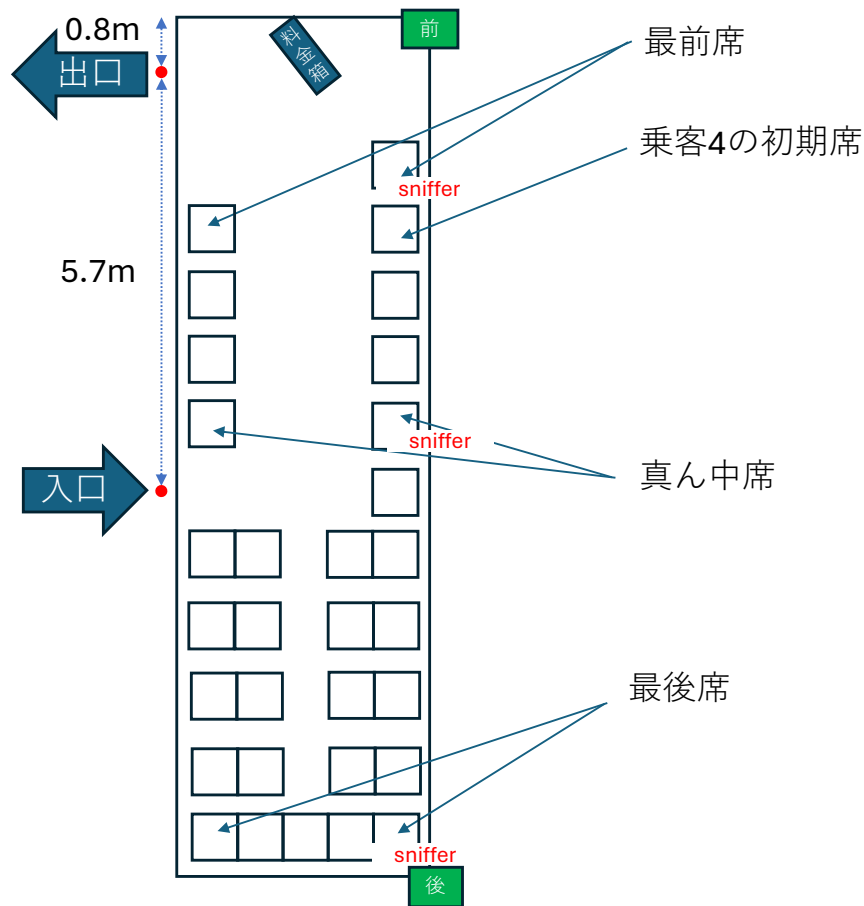


図 34: 実験環境: バス車内の俯瞰図

5.3.3 実験で発生させるイベントのスケジュール

実験でイベントを発生させる方法は実験 A と同様である。イベントのスケジュールを表 19, 20, 21 に示す。

5.3.4 実験で取得したデータの分析

実験終了後に取得したデータセットに対して、実験 A の時と同様にバス路線データを再現できているかの分析を行った。なお、バス車外の間欠的なイベントについては Android スマートフォンの BLE アドバタイジング・パケットをスニファァで受信できていなかったため、iPhone の BLE アドバタイジング・パケットのみを使って分析した。また、今回車外スマートフォンとして使用した iPhone15 Pro

表 19: iPhone SE2 & Pixel 4a (スマートフォン 1) のスケジュール表 (実験 B)

時刻	バス停の発着	行動開始時にいる場所	データパターン	行動内容	
0:00:00	01 発	データパターン: 乗客 1	バス車内	・バス車内の最前席に乗った状態で始める	
0:01:00			バス車内	・最前席から真ん中の席に移動	
0:03:00			バス車内	・真ん中の席から最前席に移動	
0:05:00	02 着		バス車内	・バスを降り、バス左横から歩道側バス後方に歩く	
0:05:30	02 発				
0:07:20		データパターン: 乗客 3	歩道側バス後方	・歩道側バス後方から走り、バス左横で止まる	
0:07:30	03 着		バス車内	・バスに乗り、最後席に座る。	
0:08:00	03 発				
0:11:30			バス車内	・最後席から最前席に移動	
0:14:00	04 着		バス車内	・バスを降り、バス左横から歩道側バス後方に歩く	
0:14:30	04 発				
0:16:20		データパターン: 乗客 4	歩道側バス後方	・歩道側バス後方から走り、バス左横で止まる	
0:16:30	05 着		バス左横	・バスに乗り、初期席に座る	
0:17:00	05 発				
0:18:00			バス車内	・初期席から一席後ろの席に移動し座る	
0:20:00			バス車内	・2 席後ろの席 (真ん中の席) に移動し座る	
0:22:00			バス車内	・真ん中の席から料金箱へ移動し、10 秒停止後元の席に戻る	
0:35:00	06 着		バス車内	・バスを降り、バス左横から歩道側バス後方に歩く	

表 20: iPhone 15 Pro & Pixel 4a (スマートフォン 2) のスケジュール表 (実験 B)

時刻	バス停の発着	行動開始時にいる場所	データパターン	行動内容	
0:00:00	01 発	データパターン: 乗客 2	バス車内	・バス車内の最前席に乗った状態で始める	
0:05:00	02 着				
0:05:30	02 発				
0:07:30	03 着				
0:08:00	03 発				
0:14:00	04 着		バス車内	・バスを降り、バス左横から歩道側バス後方に歩く	
0:16:20		データパターン: 乗客 5	歩道側バス後方	・歩道側バス後方から走り、バス左横で止まる	
0:16:30	05 着		バス左横	・バスに乗り、初期席に座る	
0:17:00	05 発				
0:18:00					
0:20:00					
0:22:00					
0:35:00	06 着		バス車内	・バスを降り、バス左横から歩道側バス後方に歩く	

表 21: iPhone 15 Pro & Pixel 4a (スマートフォン 3) のスケジュール表 (実験 B)

時刻	バス停の発着	行動開始時にいる場所	データパターン	行動内容
0:09:00		データパターン: 自動車 1 (パターン 3)	車道側バス後方	・車道側バス後方から車道側バス前まで速度一定で走る
0:11:00			車道側バス前	・車道側バス前から走り、バス前まで移動し、2 分間停止
0:13:00			バス車内	・バス前から車道側バス前まで加速しながら走る
0:18:30		データパターン: 自動車 2	車道側バス後方	・車道側バス後方から走り、バス右横まで移動し、30 秒停止
0:19:00			バス右横	・並走用地点を 1 分間歩いて往復する
0:20:00			バス右横	・バス右横で 1 分間停止
0:21:00			バス右横	・並走用地点を 1 分間歩いて往復する
0:22:00			バス右横	・バス右横で 1 分間停止
0:23:00			バス右横	・並走用地点を 11 分 55 秒歩いて往復する
0:34:55			バス右横	・並走していた地点から車道側バス前まで加速しながら走る

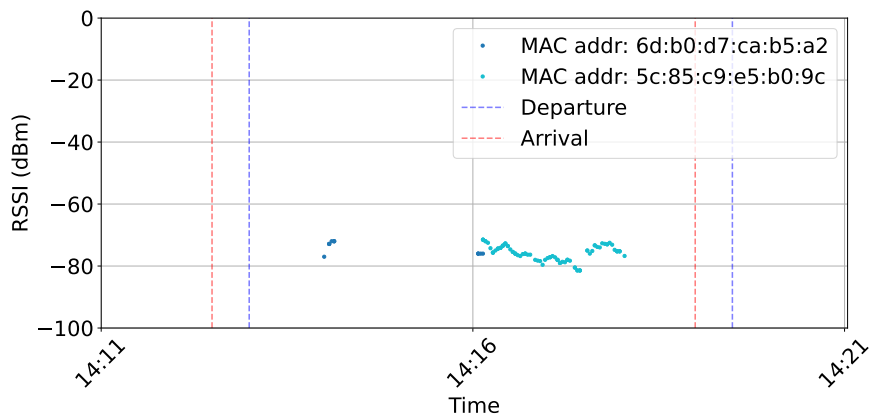


図 35: 実験 B: 自動車の追い越し後, 信号で再度追い付く様子を模した 16 パケットの RSSI の時系列変化

だが, Find My パケット, Nearby パケットどちらも定期的には送信していないことが実験後に分かった. 理由は不明であるが, Find My の判明していない仕様上の問題である可能性がある. 一方で, iPhone15 Pro から Manufacturer Specific Data の Type が 16 を示すパケット (16 パケット) が安定的に長期間送信されていたことが確認できた. 16 パケットは Find My パケットと比較して, RSSI の時系列変化の様子と定期的にパケットが送信されている点で同様の傾向が示されていたため, 16 パケットを評価に使用する. 以降で, 取得した BLE アドバタイジング・パケットについて, 発生させたイベントごとに RSSI を時系列で図示する. 三つあるスニファ－のうち, 実験 A と同様に前に設置したスニファ－をもとに分析を行う.

バスを追い越したのち, 信号等で再度合流する自動車 図 35 に自動車の追い越しがあり, その後信号で停止するタイミングで追い越された自動車に追いつき, 信号が変わったタイミングで再度距離が離されるといふ動きを仮定した 16 パケットの RSSI を示す.

結果から, 実験 A 同様一瞬出現したのち消失し, その後時間をおいて再出現するような動きをしていることが確認できる. 実験 A の時よりも確認できる BLE アドバタイジング・パケットの数が多く, RSSI も大きい値となっている. これは前回よりもバスと自動車との距離が小さくなったことによるものであると考えられる. 実データと比較しても, 設定した基準を満たしていると言える.

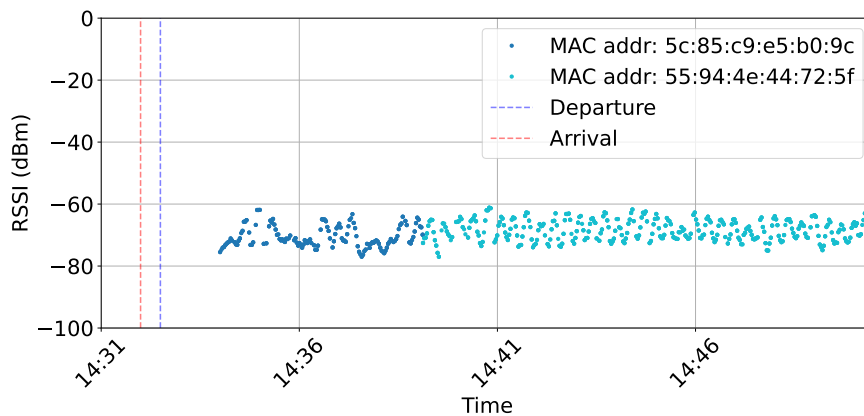


図 36: 実験 B: 並走車を模した 16 パケットの RSSI の時系列変化

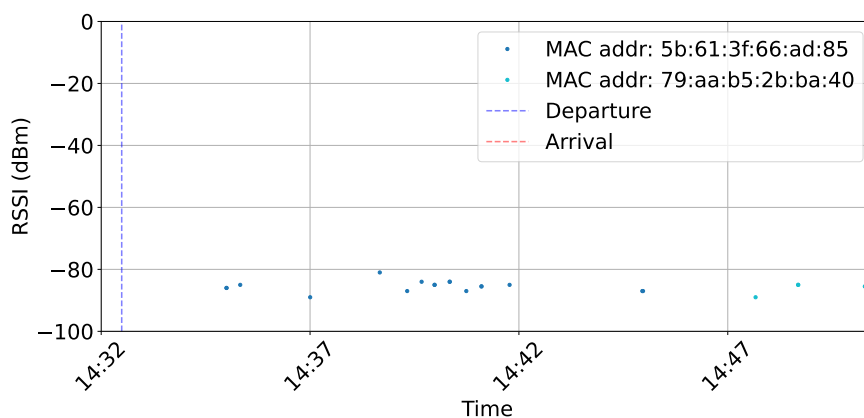


図 37: 実験 B: 並走車を模した Android スマートフォンの BLE アドバタイジング・パケットの RSSI の時系列変化

並走する自動車 図 36 に自動車の並走があり、バスがバス停で停まるタイミングで距離が離されるといふ動きを仮定した 16 パケットの RSSI を示す。結果から、実験 A 時と同様に分散が大きい RSSI の時系列変化であることが分かる。また、実験 A で取得した並走車の RSSI よりも大きい傾向にあることが分かる。間欠的な動きで述べたように、バスと自動車間の距離が小さくなったことによる可能性がある。それ以外の目立った特徴は見られなかった。次に、図 37 に同様の場面を想定した時の Android スマートフォンの BLE アドバタイジング・パケットの RSSI を示す。結果から、分散が大きく、密度が疎な RSSI の時系列変化であることが分かる。実験 A では取得できず、実験 B で取得できたのは、iPhone 同様バスと車道

の距離が小さくなったことにより、バス車外の BLE アドバタイジング・パケットも受信しやすくなったことによる可能性がある。しかし、こちらもそれ以外の目立った特徴は見られなかった。

乗客 図 38, 40 に乗客の動きを模した Find My パケットの RSSI の一例を、図 39, 41 に乗客の動きを模した Android スマートフォンの BLE アドバタイジング・パケットの一例を示す。乗客は 5 パターン分あるが、今回はバスの最後席に設置したスニファーで取得したバス最前席に座った乗客 2 のデータと、以下の三つの動きをしている乗客 4 のデータの 2 パターンを見る。

- バスが信号で停車中に席を一席後ろに移動する
- バスが信号で停車中に席を二席後ろに移動する
- バスが信号で停車中に真ん中の席から料金箱に移動し、元の席に戻る

乗客 2 はバスの最前席に設置したのち一切移動させていない。Find My パケットの RSSI の時系列変化の傾向はおおよそ実験 A で取得した乗客と類似しており、実験間で大きな違いは見られなかった。乗客 2 はスマートフォンの位置とスニファーの位置がバス内で最も離れるように設置したが、安定して BLE アドバタイジング・パケットが取得できていることがわかる。このことから、バス車内に存在するスマートフォンから送信される BLE アドバタイジング・パケットは、受信に際して間欠的な動きを見せることは仕様通りであれば起こり得ないということが分かる。また、乗客 4 は実験 A 同様多少の RSSI の変動はあるものの、安定して受信できていることが確認できる。このことから、バス車内の環境であれば、乗客の移動により途中で BLE アドバタイジング・パケットがキャプチャできなくなるといったことは起こり得ない可能性が高い。実験 B における 0xFEf3 パケットは、実験 A のときよりもバスの空間が広く、スニファーとスマートフォンの距離が離れる場面も多かったが、ほとんど同じような傾向を示していた。原因は不明であるが、壁のような仕切られた環境でなければ多少距離が離れた場合でも受信に失敗するような場面は少ないということが言える。

以上の結果から、所望のバス車内外環境を再現したデータセットを作成できたとと言える。

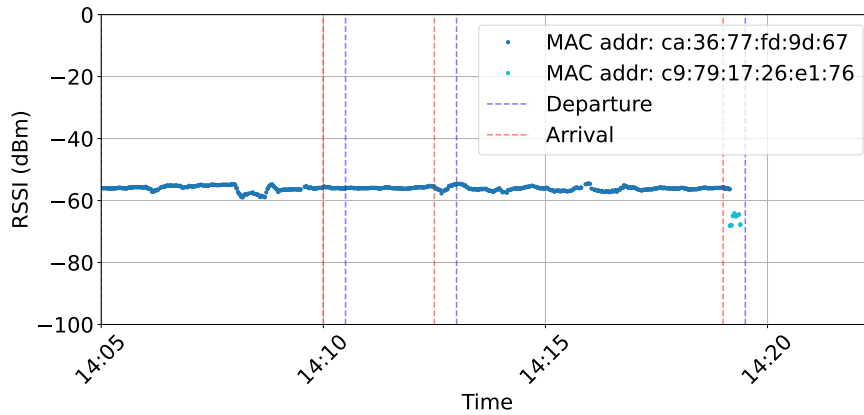


図 38: 実験 B: 乗客 2 の Find My パケットの RSSI の時系列変化

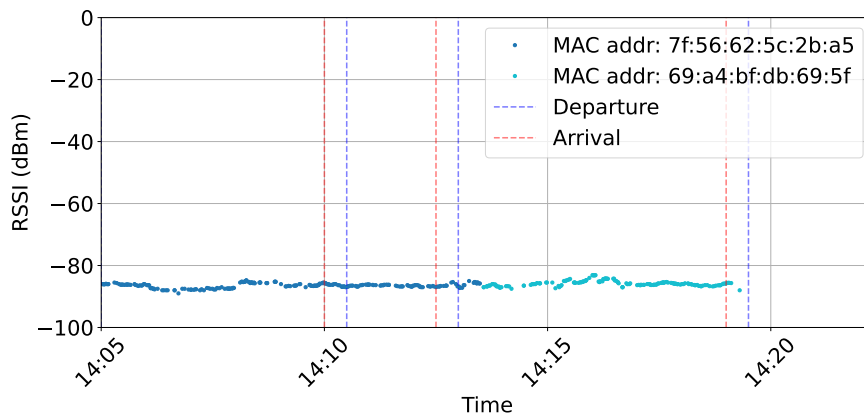


図 39: 実験 B: 乗客 2 の Android スマートフォンの BLE アドバタイジング・パケットの RSSI の時系列変化

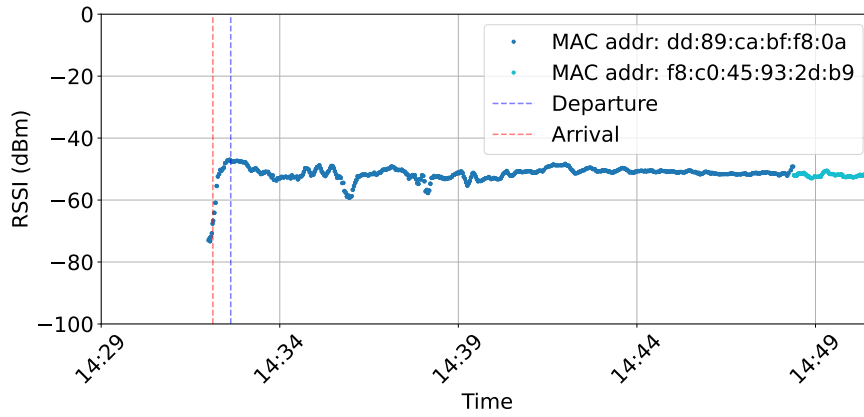


図 40: 実験 B: 乗客 4 の Find My パケットの RSSI の時系列変化

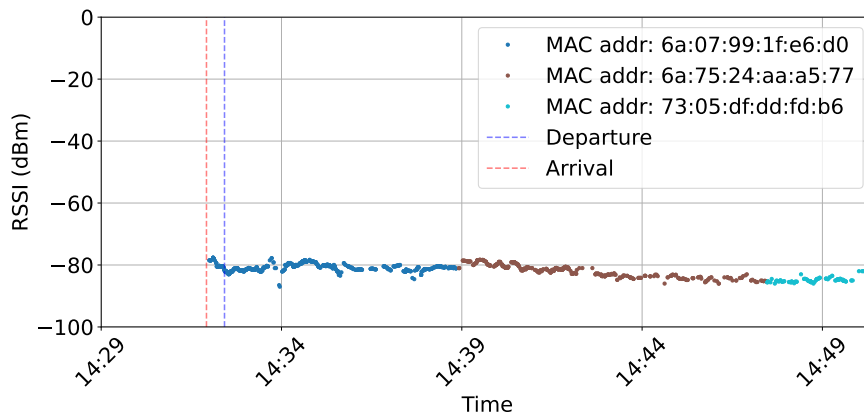


図 41: 実験 B: 乗客 4 の Android スマートフォンの BLE アドバタイジング・パケットの RSSI の時系列変化

6. バス車内外分類手法の評価

本章では、6.1 節で作成したデータセットを用いた提案手法のバス車内外分類の性能の評価方法について述べる。6.2 節で 5 章で作成したデータセットを用いて提案手法のバス車内外分類の性能を再現率、適合率、F1 スコアの観点から評価する。6.3 節で 3.3 節で紹介したバス路線データを使ったバス OD データを推定する実験を実施し、提案するバス車内外分類がバス OD データの推定の精度を改善できる可能性について評価する。

6.1 評価手法

提案手法の車内外分類の性能を評価するために、5 章の実験 A で取得したデータセットと実験 B で取得したデータセットを使用する。既存手法との比較には RSSI の閾値と出現時間・出現回数で分類する Oransirikul らの手法 [9] と出現期間の閾値で分類する Kawashima らの手法 [7] を使用する。なお、Oransirikul らの手法は、リアルタイムに MAC アドレスを車内外に分類するためのものであり、かつ Wi-Fi のプローブリクエストに含まれる MAC アドレスを分類するものである。よって、用途が若干異なっていることに注意する。Oransirikul らの手法は、事前に収集したデータから RSSI の閾値を決定する。そこで、実験 A で取得したデータセットをもとに閾値を決定した。データセットをバス車内外のパターンごとに見た時の RSSI の平均値を表 22 に示す。今回は前側に設置したスニファーマを元に評価を行う。そこで、前側に設置したスニファーマの値を参考にした。Find My パケットは車内であれば -70dBm 以上、車外であれば -80dBm 以上の数値となっていることが分かる。実験 A の教室は実際のバスよりも小さい空間であった。対して、車道は 7m と一般的な距離よりも大きい。また、バス車内で RSSI が -60dBm 以上を示す場面はほとんど見られなかったことから、高い RSSI として設定されている -40dBm を -60dBm に、比較的高い RSSI として設定されている -60dBm を -75dBm に変更した。0xFEf3 パケットについては、車内に存在する場合のみ BLE アドバタイジング・パケットを受信できており、受信できたとしても非常に小さい値を示していた。このことから、0xFEf3 パケットは、高い RSSI として設定されている -40dBm を -80dBm に、比較的高い RSSI として設定されている -60dBm を -90dBm に変更した。

本研究ではバス車内の乗客の MAC アドレスはアドレスキャリーオーバーアルゴリズムの候補として残しつつ、紐付け候補のノイズになるバス車外の MAC アド

レスを可能な限りフィルタリングすることを検討している。そこで、車内の MAC アドレスはバス車外に分類しない上で、どの程度バス車外の MAC アドレスをバス車外に分類できたかが重要である。この前提から、バス車外の MAC アドレスを分類対象とし、これをどれだけ正しく車外に分類できたか、その際バス車内の MAC アドレスの分類ミスが発生していないかという点から評価する。

実験 A, B で収集したデータセットは、MAC アドレスごとに車内・車外のラベルを付与している。ラベル付与後のデータセットの MAC アドレスについて、車内、車外それぞれの数を表 23 に示す。乗客や並走車といった想定している一連の行動パターン通りにパケットをとっていても、乗客が降車した直後に MAC アドレスが変化するような意図しない部分で MAC アドレスランダム化が発生している場合がある。乗客がバスから降車したのち MAC アドレスが変化した場合、当該 MAC アドレスは車外とし、乗客がバスに乗っている状態で MAC アドレスが変化し、そのまま降車した場合、または乗客がバスに乗った後に MAC アドレスが変化

表 22: 実験 A で取得したデータのパターンごとの RSSI 平均値

測定対象	種別	iOS (dBm)			Android (dBm)		
		front	behind	center	front	behind	center
乗客	-	-62.33	-68.19	-66.48	-85.73	-87.48	-84.33
	-	-59.78	-58.71	-57.61	-83.84	-84.42	-80.65
	-	-61.93	-65.95	-55.80	-81.61	-85.47	-81.56
自動車	間欠的な動き	-86.00	-85.50	-85.75	-	-	-
	バスを追い越した	-86.00	-84.00	-	-	-	-
	並走	-83.04	-81.55	-82.03	-	-	-
歩行者	バス停車中に横を歩く	-86.00	-84.00	-	-	-	-
	横断歩道を歩く	-84.00	-82.94	-80.44	-	-	-
	バス停に滞在	-87.55	-81.78	-82.83	-	-	-

表 23: 評価データセットの MAC アドレスの内訳

	実験 A		実験 B		実験 B (全車外含む)	
	iOS	Android	iOS	Android	iOS	Android
車内	7	11	7	11	7	11
車外	6	0	6	2	279	285

した場合、どちらも車内のラベルを付与した。

6.2 バス車内外分類の正解率

既存手法と提案手法で MAC アドレスごとにバス車内外分類を実施した。実験 A で収集したデータセットを使って MAC アドレスの分類性能を評価した。0xFEFE3 パケットは実験 A のデータセットに車外の MAC アドレスが含まれていないため、評価しなかった。それぞれの手法の結果を表 24 に示す。分類結果を見ると、乗客の MAC アドレスは、提案手法、Kawashima らの手法ではどちらも全て車内として分類できていた。対して、Oransirikul らの手法は適合率が低い。これは一部乗客の MAC アドレスを車外として誤分類してしまっていることが原因である。バス車外の MAC アドレスは、提案手法、Kawashima らの手法どちらも再現率が 100% になっていない。これは一部バス車外の MAC アドレスをバス車内に誤分類していることを表す。Kawashima らの手法と提案手法を F1 スコアで比較したとき、iOS において 23.3% 提案手法の方が性能が高いことが分かる。適合率は 1 で同じであるため、再現率の差がそのまま F1 スコアの差に反映されていると言える。また、提案手法と Oransirikul らの手法を F1 スコアで比較した場合でも、提案手法の方が性能が高いことが分かる。

続いて、Oransirikul らの RSSI の閾値を調整し、実験 B で収集したデータセットを使って MAC アドレスの分類性能を評価した。それぞれの手法の結果を表 25 に示す。分類結果を見ると、乗客の MAC アドレスは、提案手法、Kawashima らの手法、閾値調整を実施した Oransirikul らの手法で全て車内として分類できていた。対して、Oransirikul らの手法は実験 A のデータセットに対する分類結果同様、一部乗客の MAC アドレスを車外として誤分類してしまっていることから適合率が低い。再現率に着目すると、Kawashima らの手法が最も再現率が低く、提案手法は閾値調整を実施した Oransirikul らの手法と同程度の性能を達成できている。こ

表 24: 実験 A のデータセットに対する車内外分類結果

手法	OS	推定精度		
		再現率	適合率	F1 スコア
Oransirikul ら	iOS	100.0%	63.6%	77.8%
Kawashima ら		42.9%	100%	60.0%
提案手法		71.4%	100.0%	83.3%

のことから、バス車内の環境に依存しない閾値のみを使って、高い分類性能を達成できていると言える。Kawashima らの手法と提案手法を F1 スコアで比較したとき、iPhone に対し 16.7%提案手法の方が性能が高いことが分かる。実験 A に対する分類結果と同様に、適合率は 100%で同じであるため、こちらも再現率の差がそのまま F1 スコアの差に反映されていると言える。また、提案手法と RSSI の閾値調整を実施した Oransirikul らの手法を F1 スコアで比較した場合、同じ値であり、RSSI を使用せずに同程度の分類性能を達成できていることが分かる。

実験環境に存在する MAC アドレスについて、今回取得対象とした MAC アドレス以外の Find My パケットと 0xFEf3 パケットがスニファァーで取得したデータセットに含まれていた。乗客以外の MAC アドレスは全て車外の MAC アドレスである。そこで、バス車外の MAC アドレスを加え、再度バス車内外分類を実施した。バス車外の MAC アドレスは、実験 B で元々車外ラベルを付けていた MAC アドレスと合わせて Find My パケットで 279 個、0xFEf3 パケットで 285 個である。

これに対するそれぞれの手法の結果を表 26 に示す。なお、分類結果を見ると、バス車外の MAC アドレスは Kawashima らの手法よりも Find My パケット、0xFEf3 パケットどちらも提案手法の方がより正確に分類できていることがわかった。このことから、単純な秒数の閾値設定だけでなく、バス車内から見たバス車外のスマートフォンの動きを考慮することで乗客の誤分類を抑えつつ、バス車外の MAC アドレスの分類性能が向上すると言える。対して、Oransirikul らの手法と比較したとき、提案手法の分類性能は大きく下がっていた。実験時にバス近くに人がい

表 25: 実験 B のデータセットに対する車内外分類結果

手法	OS	推定精度		
		再現率	適合率	F1 スコア
Oransirikul ら (閾値調整なし)	iOS	100.0%	85.7%	92.3%
	Android	0.0%	15.4%	26.7%
Oransirikul ら (閾値調整済み)	iOS	50.0%	100.0%	66.7%
	Android	0.0%	-	-
Kawashima ら	iOS	33.3%	100.0%	50.0%
	Android	0.0%	-	-
提案手法	iOS	50.0%	100.0%	66.7%
	Android	0.0%	-	-

なかったこと、車外の MAC アドレスの数のみが大幅に増えたこと、実験 B ではバスが停車していたことが原因で、再現率が低下したと考えられる。Kawashima らの手法と提案手法を F1 スコアで比較したとき、iOS は 1.5%、Android は 0.2% 提案手法の方が性能が高いことが分かる。しかし、これまでの分類結果と比べ分類性能の差は非常に小さい。また、提案手法と RSSI の閾値調整を実施した Oransirikul らの手法を F1 スコアで比較した場合、iOS で 8.3%、Android で 0.3% 提案手法の方が性能が低いことが分かる。これまでの分類結果と異なり、明確に提案手法の方が分類性能が低いものの、いずれの OS でも分類性能の差は 10% 以内であり、特に Android ではほとんど差がなかった。

6.3 バス OD データの推定性能の評価

提案手法がバス OD データを推定するために使用することで有効に機能するか検証する。検証のために 5 章で分析に使用したデータセットを使用して実際にバス OD データを作成し、再現率と適合率から評価する。バス OD データを作成するにあたり、Kawashima らのアルゴリズムをベースとしたアドレスキャリーオーバーアルゴリズムとバス乗車・降車判定アルゴリズムを作成した。作成したアドレスキャリーオーバーアルゴリズムをアルゴリズム 4 に、バス乗車・降車判定アルゴリズムをアルゴリズム 5 に示す。

アドレスキャリーオーバーアルゴリズムには三つの閾値を設定する必要がある。以下に閾値の意味と設定した値を述べる。

表 26: 実験 B のバス車外 MAC アドレスを全て含めたデータセット分類結果

手法	OS	推定精度		
		再現率	適合率	F1 スコア
Oransirikul ら (閾値調整なし)	iOS	100.0%	99.6%	99.8%
	Android	100.0%	96.3%	98.1%
Oransirikul ら (閾値調整済み)	iOS	98.9%	100.0%	99.5%
	Android	89.1%	100.0%	94.2%
Kawashima ら	iOS	81.4%	100.0%	89.7%
	Android	88.1%	100.0%	93.7%
提案手法	iOS	83.9%	100.0%	91.2%
	Android	88.4%	100.0%	93.9%

- RSSI_{th}: アドレスの紐付けに使用する RSSI の閾値. MAC アドレス単位で見た時, RSSI の平均値の差がこの値より大きい場合紐付けを行わない. Kawashima らの設定した値と同じ 15dBm とした. これは, RSSI の差はスニファーで取得するときに相対的な値の差となるため調整は不要であると考え, 設定した.
- extract_{int}: MAC アドレスが変更されて一定時間内に観測された新しい MAC アドレスを抽出するための時間間隔の閾値. Find My パケットの場合 15 秒, Android スマートフォンの BLE アドバタイジング・パケットの場合 10 秒とした.
- timetable_{int}: 乗客がバスに乗車・降車したと判断するための閾値. 今回はバスの停車時刻は分かっている前提として, 閾値を 30 秒として閾値秒前からバス停車時間に出現し始めた MAC アドレスを乗車, バス停車時間から閾値秒後までに消失した MAC アドレスを降車とした.

アドレスキャリーオーバーアルゴリズムの実行結果について, 表 27 に便ごとの再現率, 適合率, F1 スコアと全体の再現率, 適合率, F1 スコアを示す. なお, Oransirikul らの手法は RSSI の閾値を調整した後の値を使用している. 便ごとに見た時, 8 時 25 分の便は大きな差はないが, Oransirikul らは最も多くの乗客を検出することに成功しており, 再現率も高い値を示していた. 対して, 提案手法は乗客の検出には失敗しているものの適合率 90% を達成しており, 高精度な予測ができています. 14 時 30 分の便も大きな差はなく, ほとんど乗客を検出できていないことが分かる. その中でも, Oransirikul らの手法が最も乗客の間違いを抑制できていることが分かる. 17 時 37 分の便はこれまでの予測結果に比べ Oransirikul らの手法が再現率, 適合率ともに低く, Kawashima らの手法と提案手法が高い再現率, 適合率を示していることが分かる. 最後に, 三つの便を総合してみた時の各指標を見た時, 提案手法が最も高い F1 スコアを達成していることが分かった. このことから, 今回提案したアルゴリズムが, バス OD データの取得性能向上につながる可能性が高いと言える.

アルゴリズム 4 アドレス・キャリーオーバーアルゴリズム (Find My/0xFEf3)

Require: observed_dct, noise_eliminate_addrs, device_type

```
1: if device_type = iOS then
2:   extract_int ← 15
3: else
4:   extract_int ← 10 ▷ Android
5: end if
6: for ne ∈ noise_eliminate_addrs do
7:   o_addrs ← observed_dct のアドレス一覧
8:   lo_time ← observed_dct[ne] の終了時刻
9:   up_time ← lo_time + extract_int
10:  for o ∈ o_addrs do
11:    fst ← observed_dct[o] の開始時刻
12:    if lo_time ≤ fst ∧ fst ≤ up_time then
13:      addrs に o を追加
14:    end if
15:  end for
16:  carryover_dct に {ne: addrs} を追加
17:  for co_addrs ∈ carryover_dct[ne] do
18:    rssi_df ← |disappeared_dct[ne].RSSI - observed_dct[co_addrs].RSSI|
19:  end for
20:  min_rssi_df ← min(rssi_df)
21:  rssi_idx ← rssi_df で min_rssi_df と一致する添字
22:  target_addr ← carryover_dct[ne][rssi_idx]
23:  if min_rssi_df < rssi_th then
24:    target_addr の前 MAC アドレス ← ne の MAC
25:  else
26:    observed_dct[ne] の降車フラグ ← True
27:  end if
28:  carryover_dct から ne を削除
29:  for c_key ∈ carryover_dct.keys() do
30:    if target_addr ∈ carryover_dct[c_key] then
31:      c_key の target_addr を削除
32:    end if
33:  end for
34: end for
```

アルゴリズム 5 OD データを生成する関数

Require: passenger_data, timetable_int

1: **Variables:**

2: od_data: bus_stop_list × bus_stop_list の 2 次元配列 (各要素は 0 で初期化)

3: bus_stop_list ← 時系列順に整列したバス停 (1 運行分)

4: bus_schedule_list ← バス停毎の到着時刻と出発時刻のリスト

5: ids ← passenger_data の利用者 ID

6: **for** id ∈ ids **do**

7: fst ← passenger_data[id] の観測開始時刻

8: lst ← passenger_data[id] の観測終了時刻

9: **for** stop ∈ bus_schedule_list **do**

10: arrival_time ← stop の到着時刻

11: departure_time ← stop の出発時刻

12: **if** (arrival_time - timetable_int) ≤ fst ∧ fst ≤ departure_time **then**

13: origin_idx ← stop のインデックス

14: o_stop ← bus_stop_list[origin_idx]

15: **for** dest_stop ∈ bus_schedule_list[origin_idx+1:] **do**

16: d_arrival_time ← dest_stop の到着時刻

17: d_departure_time ← dest_stop の出発時刻

18: **if** d_arrival_time ≤ lst ∧ lst ≤ (d_departure_time + timetable_int) **then**

19: destination_idx ← dest_stop のインデックス

20: d_stop ← bus_stop_list[destination_idx]

21: od_data[o_stop][d_stop] ← od_data[o_stop][d_stop] + 1

22: **break**

23: **end if**

24: **end for**

25: **end if**

26: **end for**

27: **end for**

28: **return** od_data

表 27: 各手法の OD 検出精度比較

手法	便	推定精度		
		再現率	適合率	F1 スコア
Oransirikul ら	8:25	54.5%	60.0%	57.1%
	14:30	10.5%	66.7%	18.2%
	17:37	22.9%	53.3%	32.0%
	全体	28.9%	57.9%	38.6%
Kawashima ら	8:25	50.0%	61.1%	55.0%
	14:30	10.5%	33.3%	16.0%
	17:37	31.4%	61.1%	41.5%
	全体	31.6%	57.1%	40.8%
提案手法	8:25	40.9%	90.0%	56.3%
	14:30	10.5%	40.0%	16.7%
	17:37	34.3%	63.2%	44.4%
	全体	30.3%	67.6%	41.7%

7. 考察

本章では、7.1 節では6章で実施した車内外スマートフォンデータセットに関する考察について述べる。7.2 節では6章で実施した車内外スマートフォンの分類結果に対する考察について述べる。7.3 節ではスニファァの位置による分類結果の違いについて述べる。7.4 節では受信した BLE アドバイジング・パケットの CRC のバス車内外分類への利用可能性について述べる。7.5 節では6章で実施したバス OD データ作成実験の評価結果に対する考察について述べる。7.6 節で今後の展望について述べる。

7.1 車内外スマートフォン再現データセットに対する考察

バス車内外のデータセットについて、バス車内外の人物を想定し、それに相当するイベントをすることにより、バス路線データで見られるようなデータを取得することができた。本研究ではバス車外のデータに着目していたが、乗客と考えられるデータにも RSSI の時間的な変化が確認できていた。実験 A/B で取得した乗客の RSSI の時間的な変化は、図 30, 38, 40 を見るとバスの乗車/降車時に RSSI が単調に増加/減少していることが分かる。バス路線データでは MAC アドレスが変化していないものでそのような傾向を示すものは今回調査したものの中に存在しておらず、ラベルが付与されたバス乗客のデータは存在しなかったため、実際にバス路線で取得できる BLE アドバイジング・パケットにも実験 A/B の乗客を模したデータと同じような傾向が見られるかは不明であるが、同様の傾向が見られる可能性は高いと考えられる。

7.2 車内外スマートフォンの分類結果に対する考察

今回設定した iOS で三つ、Android で二つの車外判定条件を使用することで、従来手法よりも車内環境に依存した閾値調整なしで高い車内外分類性能を達成することができた。しかし、提案手法が意図した場面でうまく機能しているかどうかは分かっていない。そこで、提案手法が意図したイベントの車外 MAC アドレスを車外と分類できているかを調査した。その結果、車外のパターンとして分類には成功しているものの、意図したアルゴリズムで分類されていなかったもの、データセットの作成時に特定の車外パターンとして作成した車外 MAC アドレスの中で分類に失敗しているもの、データセットの作成時は車外分類できることを想定し

ていなかったが分類結果を見ると車外分類に成功しているもののような、意図していない結果を示しているものが存在することがわかった。また、提案手法のアルゴリズムのどの部分が分類の性能の向上に繋がったかどうかについても調査した。すなわち、提案した三つのアルゴリズムについて、どのアルゴリズムが従来手法で分類できなかったような車外の MAC アドレスを分類し、性能向上に貢献しているかを分析した。最後に、実験 B で収集した対象外の MAC アドレスを含めたバス車内外分類の結果についても調査した。

7.2.1 車外のパターンが意図していない分類結果だったもの

車外のパターンが対応した通りのアルゴリズムで分類できていないものとして、実験 A のデータセットの間欠的な動きをする自動車、並走車の一部、バス停で待つ歩行者が挙げられる。図 25 を見ると、間欠的な動きをする自動車は、定性的な観点から見た時、バス路線データで見られるような動きをしていた。しかし、当該 MAC アドレスの送信間隔は最大値が 138.4 秒、中央値が 26.1 秒であった。このことから、信号で止まる動きが入っていないながら送信間隔が大きく開いてしまっていることが分かる。図 26 で示した並走車は提案アルゴリズムでバス車外に分類することは想定していないが、一部の MAC アドレスが間欠的な動きとしてバス車外に分類されていた。並走車の MAC アドレス二つのうち変化前の MAC アドレスの送信間隔は最大値が 16.0 秒、中央値が 2.0 秒であり、図 29 で示したバス停で待つ歩行者は本来であればバス停停車期間にのみ出現するパターンとしてバス車外に分類されるはずであるが、今回の分類結果では間欠的な動きという点も満たしていることがわかった。当該 MAC アドレスの送信間隔の最大値は 18.0 秒、中央値は 4.0 秒であった。

このような結果になった原因として、実験環境のバスと車道の距離が本来よりも離れていたことが考えられる。バスと自動車の距離は、5.2 節で述べたように 3.5m 程度である。対して、実験 A におけるバスと自動車の距離は 7m としており、本来の 2 倍程度距離が離れていた。加えて、バス停で待つ歩行者は、実験環境を考慮して車道側で再現した。これによりバス車外の三つが通常よりもパケットが受信しにくい状況にあり、受信に失敗した結果、送信間隔が大きくなったと考えられる。

7.2.2 先行研究と比較して有効だったもの

実験 A におけるバス停に留まっている歩行者と並走車の一部の MAC アドレスと、実験 B における間欠的な動きをする自動車の MAC アドレスを、Kawashima らの手法では車外に分類できていなかったが、提案手法ではバス車外に分類することができていた。単純な出現期間のみに着目した手法では、このような MAC アドレスの出現期間は長いにも関わらず、送信間隔や出現時間に特徴があるものを車外に分類することは難しい。この点で、提案手法は MAC アドレスの送信間隔やバスが停まっていることが明らかなバス停車期間を用いることで、出現期間のみで車外に分類することが難しいものの一部を分類することに有効であると言える。

また、Oransirikul らの手法と比較したとき、RSSI の閾値調整を行なった場合は同程度の性能となったものの、閾値調整を実施した場合乗客を車外と分類する誤分類が発生していた。この点において、提案手法は乗客の誤分類は実験内で発生しておらず、乗客の取りこぼしは発生しにくいと言える。これは MAC アドレスランダム化の期間やバス停車期間について、バス車内の MAC アドレスの誤分類がないようにするために制約を厳しく設定したことによると考えられる。閾値を大きくすれば、その分多くの車外の MAC アドレスを車外として分類できるが、乗客の誤分類も増加する可能性がある。

7.2.3 実験 B の対象外 MAC アドレスを含めた分類結果の詳細について

実験 B で収集した全ての Find My パケット、0xFEFE3 パケットの MAC アドレスのバス車外分類の結果について分析した。提案手法と Kawashima らの手法を比較したとき、提案手法の方がバス車内外分類性能は高かった。提案手法で分類に失敗し、Kawashima らの手法で分類に成功していた MAC アドレスについてみたとき、Find My パケットでは間欠的な動きをする MAC アドレスの分類結果で差がみられた。このような MAC アドレスの一部を図 42 に示す。今回の環境はバスが停止しているため、周辺のスマートフォンによる BLE アドバタイジング・パケットもバスの近くにあれば受信し続ける。しかし、距離が離れていると BLE アドバタイジング・パケットが受信できず、図 42 のような並走車のような動きだが受信間隔が長くなってしまった MAC アドレスを車外の MAC アドレスとして分類することができると考えられる。0xFEFE3 パケットでは Kawashima らの手法と分類性能がほとんど変わらなかった。提案手法で分類に成功して Kawashima らの手法で分類に失敗した MAC アドレスは一つのみであり、当該 MAC アドレスは出現時間が 60 秒かどうか、手法の中の出現時間の違いのみの差であった。提案手

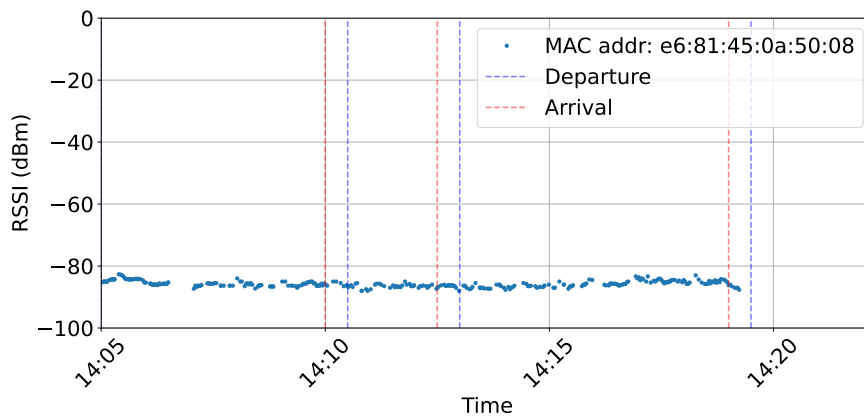


図 42: 提案手法で車外分類に成功した対象外スマートフォンの間欠的な動き (実験 B)

法において 0xFEFE3 パケットは Find My パケットの車外分類条件に比べて条件が一つ少なく, Kawashima らの手法との差別点が少ない。加えて, 今回の実験で設定したバスの停車時間は全て 30 秒であったため, 本来であれば提案手法でしか分類できないようなバス停停車期間中のみ出現していた MAC アドレスを分類する際, 出現時間の閾値だけで車外に分類できてしまっていたことが原因として考えられる。

次に, 提案手法と Oransirikul らの手法を比較したとき, Oransirikul らの手法の方がバス車内外分類性能は高かった。提案手法で分類に失敗し, Oransirikul らの手法で分類に成功していた MAC アドレスについてみたとき, 主に並走車のような動きをしているものと提案手法のアルゴリズムで分類が難しい間欠的な動きを示すものの大きく二つのパターンが確認できた。実験当日は実験に使用したバスの車道側に運転手が乗車したバスが数十分間停止している, 使用したバスの前側で自動車が入り出すといった状況が発生していた。このような状況から, 並走車の動きと類似した MAC アドレスは, 実験に使用したバスの近くにいたバス運転手の方や自動車に乗車していた方のスマートフォンから送出された BLE アドバタイジング・パケットの MAC アドレスの可能性が高い。対して, 間欠的な動きをしていた MAC アドレスはいくつの特徴が見られた。図 43, 44 に実験 B の対象外 MAC アドレスの中で, 間欠的な動きを示していたものの例を示す。Find My パケットの本来の挙動を考えたとき, このような挙動はバスが動いているような環境であれば発生しうるが, バスが静止している環境では事前の分析結果通りに

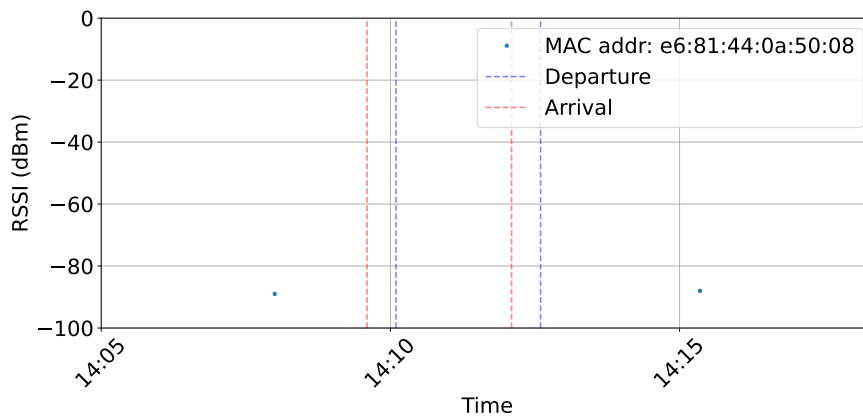


図 43: 二つの BLE アドバタイジング・パケットが確認できる MAC アドレス (実験 B)

送信しているとは考えにくい。Find My パケットの仕様は製作者が明らかにしていないため、この挙動の原因は不明である。加えて、そのような送信間隔を示す MAC アドレスは以下のようにになっていた。

- e6:81:45:0a:70:08
- e6:81:45:0e:50:08
- e6:81:44:0a:50:08
- f6:81:45:0a:50:08
- e6:81:45:0a:56:08
- e6:81:45:0a:5c:08

確認できたものを見ると、どの MAC アドレスも非常に類似した値であることがわかる。このことから、なんらかのスマートフォンが発生させている特殊な Find My パケットの可能性がある。また、送信時間を見たとき、それぞれの MAC アドレスの送信時間は重複している部分がある。このような Find My パケットがどのようなスマートフォンから送出されているかわかれば、MAC アドレスの送出元の数や、MAC アドレスの車外分類方法等、より詳細化できる可能性がある。

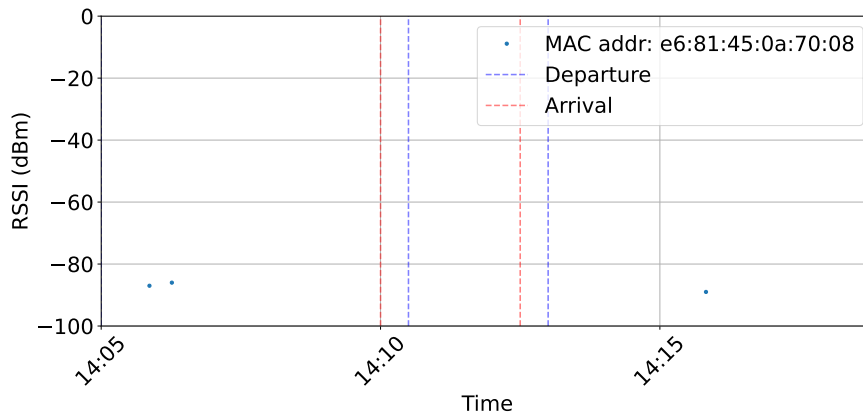


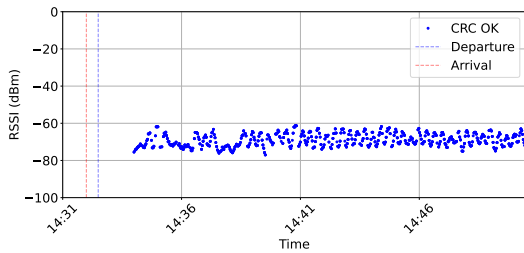
図 44: 三つの BLE アドバタイジング・パケットが確認できる MAC アドレス (実験 B)

7.3 スニファアの位置について

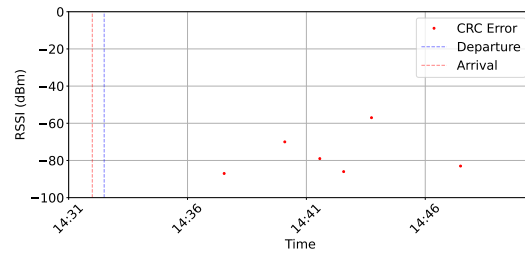
スニファアの位置について、実験 A/B 通して前側に設置したスニファアを使用して分類性能を検証した。前側に設置したスニファア以外を使用した場合、実験 B において並走車の MAC アドレスの平均 RSSI が小さくなり、Oransirikul らの手法の分類性能が向上する。この点で、RSSI を使用する手法はスニファアの設置位置によっても性能が左右されることがわかる。対して、提案手法はスニファアの位置といった車内環境で性能が上下することではなく、車内環境に依存しない車内外分類ができていた。しかし、分類性能の面で見ると、いずれの場所に設置しても Orasirikul らの手法が提案手法の性能を下回ることはなかった。このことから、閾値調整ができることが前提であれば、RSSI を使用した方が車内外分類性能が高い可能性がある。

7.4 バス車内外分類における CRC の利用可能性について

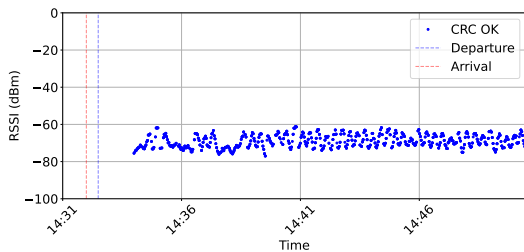
車外の並走車の MAC アドレスについては、渋滞時にバスは多数の車と並走状態になることが予想され、そうなった場合提案手法の分類性能は大きく下がることが予想される。この点で、スニファアで取得できる CRC の値はパケットの破損を示し、これを活用することで車外の並走車内にあるスマートフォンを車外スマートフォンとして識別できる可能性があると考えている。CRC はスニファアで BLE アドバタイジング・パケットをキャプチャすることで確認することができ、破損



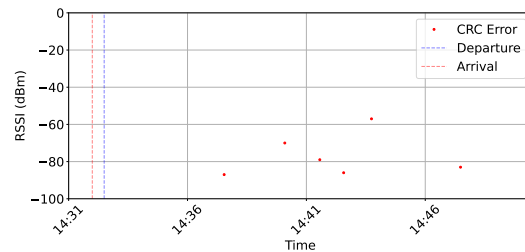
(a) CRC エラーが発生していない乗客の Find My パケット



(b) CRC エラーが発生している乗客の Find My パケット



(a) CRC エラーが発生していない並走車の 16 パケット



(b) CRC エラーが発生している並走車の 16 パケット

していた場合 CRC エラーとなる。この CRC が正常なものとエラーとなっているものについて、乗客 4 の CRC が正常なものの BLE アドバイジング・パケットと CRC エラーが発生している BLE アドバイジング・パケットの様子を図 45a, 45b に、並走車の CRC が正常なものの BLE アドバイジング・パケットと CRC エラーが発生している BLE アドバイジング・パケットの様子を図 46a, 46b に示す。なお、乗客 4 は Find My パケット、並走車は 16 パケットを表示しており、BLE アドバイジング・パケットの送信間隔が異なることに注意する。乗客はほとんど CRC エラーが発生していないのに対し、並走車は複数の CRC エラーが発生した BLE アドバイジング・パケット送信していることが確認できる。次に、車内外全てのスマートフォンに対して、スニファーごと、パターンごとに CRC エラー率を計算した。計算結果を表 28 に示す。スニファーごとに見た時、確実に並走車を車外のスマートフォンとして分類することは難しいが、スニファー全体で見た時、1%を超える CRC エラー率となっているパターンは並走車のみであったことから、RSSI などの特徴量とうまく組み合わせることで分類性能に繋がれる可能性がある。CRC エラーの発生率は RSSI の減衰と同じように壁の反射等が原因で高くなる。また、Signal-Noise Ratio (SNR) がバス車内で低く、これにより

CRC エラー率が高く出た可能性がある。

7.5 バス OD データ自動取得手法の実験評価に対する考察

バス OD データの推定性能については、評価指標の観点から、提案手法の F1 スコアが全体を通して高い傾向にあることが分かった。しかし、MAC アドレスに正

表 28: スニファ－の設置位置による比較

前方設置			
パターン	総パケット数	CRC エラーパケット	CRC エラー率
乗客 1	378	1	0.26%
乗客 2	905	2	0.22%
乗客 3	376	2	0.53%
乗客 4	1595	1	0.06%
乗客 5	1612	1	0.06%
乗用車 1	104	2	1.92%
乗用車 2	1313	6	0.46%
後方設置			
乗客 1	391	2	0.51%
乗客 2	1131	2	0.18%
乗客 3	496	0	0.00%
乗客 4	1579	1	0.06%
乗客 5	1582	1	0.06%
乗用車 1	162	3	1.23%
乗用車 2	1284	2	0.16%
中央設置			
乗客 1	429	0	0.00%
乗客 2	1154	0	0.00%
乗客 3	529	0	0.00%
乗客 4	1602	1	0.06%
乗客 5	1623	1	0.06%
乗用車 1	141	3	2.13%
乗用車 2	1257	18	1.43%

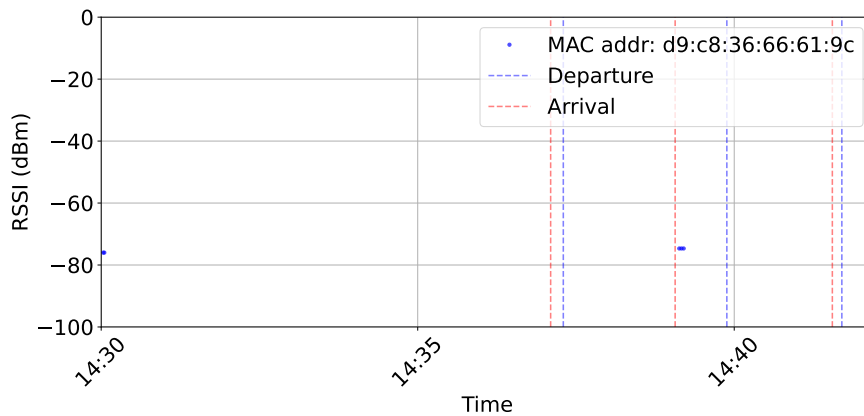


図 47: 間欠的な動きをするバス OD データの不正解パターン

解ラベルが存在しないため、実際に提案手法によって車内外分類がうまくいったことにより性能が向上したかどうかは不明である。そこで、アドレスキャリーオーバーアルゴリズムにより紐付けられた MAC アドレスについてバス OD データの正解と比較しながら分析した。Kawashima らの手法でバス車内外分類を実施したのち、バスの乗車条件・降車条件を満たしているとは判定されたにも関わらず、乗り降りがなかった区間の MAC アドレスの例を図 47 に示す。提案手法で述べたような、送信間隔に大きな開きがあり、それがバス停車区間で見えている様子が分かる。このような失敗パターン以外にも、間欠的な動きを示す MAC アドレスの紐付けの結果、バスの乗車・降車区間に合わない紐付けも多数見られた。このような紐づけにより、バス OD データの推定性能の低下が考えられる。この観点から、間欠的な動きをしているパターンはバス車外として分類し、削除することが有効であると考えられる。

対して、提案手法の分類方法により、他の手法で繋ぎ合わせることができていた正解のバス OD データの推定に失敗しているものも存在した。送信間隔に開きがあるにも関わらず、正解のバス OD データと一致していた例を図 48 に示す。これまでの実験から Find My パケットの送信間隔は 2 秒程度で、最大でも 15 秒未満であったが、例に示したものは 1 分程度の間隔を空けて定期的に BLE アドバタイジング・パケットを送信している様子が確認できる。実験データ全体でもこのような傾向が見られるものはほとんどなかったが、Find My の仕様が不明なため、なぜこのような挙動をしているか、その条件は不明である。しかし、一定間隔で送信していることは挙動から確認できるため、今後は送信間隔が一定かどうか、

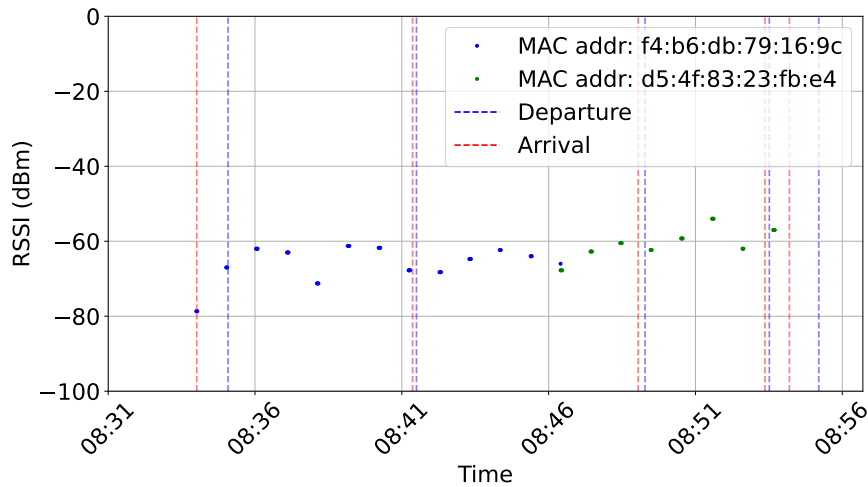


図 48: 受信間隔が大きいバス OD データと一致しているデータパターン

という条件を加えるなどすることで対応できると考えられる。また、提案手法と Oransirikul らの手法を比較したとき、特に 17 時 37 分発の便で Oransirikul らの手法は大きく性能を落としていた。バス OD データの正誤について確認したところ、提案手法でバス車内分類を実施したものと比べ、Oransirikul らの手法は乗客の正解が 4 人少なかった。提案手法の予測結果と Oransirikul らの予測結果を比較したとき、Oransirikul らの手法の方が Find My パケットのバス OD データの組み合わせが少ないことが分かった。これは、RSSI の閾値により、乗客の MAC アドレスを過剰に除去してしまった可能性がある。

7.6 今後の展望

今後の展望として、CRC のような他の特徴を組み合わせたスマートフォンのバス車内分類手法、乗客の特徴を活用したバス車内の BLE アドバイジング・パケットのフィルタリングが考えられる。提案手法では車外の BLE アドバイジング・パケットの時系列的な特徴に着目したが、6 章で示した通り完全な車内外分類は難しい。そこで、様々なパターンを想定し、より信頼性が高いバス車外の BLE アドバイジング・パケットをフィルタリングすることが理想である。そのために、本研究で提案した受信間隔のパターンマッチングの条件に CRC の条件を追加する、時系列的な特徴をフィルタリングするための閾値条件を調整するといった工夫を行うことで、バス車内外分類性能の向上につながる可能性がある。別の観

点でみると、このように完全に車外の BLE アドバタイジング・パケットを完全にフィルタリングすることは難しいため、車内の MAC アドレスを残すようなアルゴリズムを追加することで、バス車内外分類の性能を向上させられる可能性がある。

また、実験 B の結果から、スマートフォンによっては想定と異なる挙動をすることがわかった。バス路線データにおいても、パケットの送信時間が長い状態で安定しているような、Find My パケットの本来の仕様と異なるような挙動をしているものも見られている。これらは製造者が非公開にしており、状況に応じた詳細な挙動は分かっていないため、今後調査することで、複数種類の BLE アドバタイジング・パケットを組み合わせつつ、種別ごとに異なるアルゴリズムを適用することで更なる車内外分類の汎用性を高めることができる可能性がある。

8. おわりに

本研究ではバス車内外の MAC アドレスの分類性能を, RSSI のようなバス車内の環境に依存する値を使用せずに向上させることを目的とした, BLE アドバタイジング・パケットのバス車内外分類に取り組んだ. 車外のスマートフォンから送信されている BLE アドバタイジング・パケットは, 自動車で追い越しをする, バス停でバスを待つといった行動をとる人が所持している場合, 受信間隔や出現期間に特徴があることが明らかになった. このような特徴を考慮した閾値を使ってバス車内外分類をすることで, バス車内の環境に依存する変数を使用することなく, バス車外の BLE アドバタイジング・パケットをフィルタリングできると考えた. これを検証する目的で, バス車内外のスマートフォンの動きを再現し, バス車内外のスマートフォンから発されている BLE アドバタイジング・パケットの RSSI と BLE アドバタイジング・パケットの送信間隔を実際に運行されているバスで取得したデータと比較し, バス車外特有の BLE アドバタイジング・パケットの受信パターンを定義した. パターンをもとに複数の閾値処理を BLE アドバタイジング・パケットに対して行い, バス車内外分類を実施した結果, 提案手法は既存の閾値処理と比較して, RSSI のような車内環境に合わせた閾値調整が不要でありながら, F1 スコアの観点から最大 23.3% の分類性能の向上を達成することができた.

謝辞

主指導教員であり、バス OD データの必要性や研究の妥当性を常日頃から意識するよう指導していただきました。本学情報基盤システム学研究室の藤川和利教授に心から感謝いたします。副指導教員であり、バスを使用した環境におけるデータセット作成方法について助言をいただきました。本学ユビキタスコンピューティングシステム研究室の安本慶一教授に心から感謝いたします。副指導教員であり、バス OD データを取得する必要性から実験の進め方に関して適切な助言をいただきました。本学情報基盤システム学研究室の新井イスマイル准教授に心から感謝いたします。

研究の方向性や論文執筆の進め方について、助言や研究の方向性に関する考え方を教えていただきました。本学情報基盤システム学研究室の遠藤新助教に心から感謝いたします。研究時に使用する機器に関する知識や研究資料に関する多くの助言をいただきました。本学情報基盤システム学研究室の垣内正年客員准教授に心から感謝いたします。研究の方針に関する助言や研究以外のお話もしていただきました。本学情報基盤システム学研究室の ARAYA Kibrom Desta 助教に心から感謝いたします。日頃の身の回りの補助や研究活動の支援をしていただきました。本学総合情報基盤センターの辻元理恵女史に心から感謝いたします。研究を進めるにあたり、助言だけでなく実験に協力していただいたり研究とは関係ない話をしていただいた情報基盤システム学研究室の学生の皆様に心から感謝申し上げます。

実証実験を行うにあたって事前にバス路線実験のデータを提供していただいた岡山大学の松田裕貴講師、吉村太斗さん、林虎太郎さん、当日の環境を用意してくださった奈良交通株式会社の皆様、奈良交通株式会社の紹介をしていただいた本学ユビキタスコンピューティングシステム研究室の諏訪博彦准教授に心から感謝いたします。

最後に、経済的、心理的な面から多大な支援をいただきました家族に心から感謝いたします。

本当にありがとうございました。

参考文献

- [1] 国土交通省. “地域の公共交通を取り巻く現状と検討の視点・課題”. <https://www.mlit.go.jp/sogoseisaku/transport/content/001728295.pdf>, 2023.
- [2] 福本雅之. “OD 表とは — 公共交通トリセツ”. <https://www.kanachu.co.jp/bus/courtesy/>. 参照: 2025-01-30.
- [3] 中部運輸局. “バスデータ活用大百科”. <https://wwwtb.mlit.go.jp/kanto/content/000166077.pdf>, 2020.
- [4] Di Huang, Jun Yu, Shiyu Shen, Zhekang Li, Luyun Zhao, and Cheng Gong. “A Method for Bus OD Matrix Estimation Using Multisource Data”. *Journal of Advanced Transportation*, Vol. 2020, pp. 1–13, Mar 2020.
- [5] Shunta Komatsu, Ryosuke Furuta, and Yukinobu Taniguchi. “Passenger Flow Estimation with Bipartite Matching on Bus Surveillance Cameras”. In *2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 206–212, Oct 2021.
- [6] Ziyuan Pu, Meixin Zhu, Wenxiang Li, Zhiyong Cui, Xiaoyu Guo, and Yin Hai Wang. “Monitoring Public Transit Ridership Flow by Passively Sensing Wi-Fi and Bluetooth Mobile Devices”. *IEEE Internet of Things Journal*, Vol. 8, No. 1, pp. 474–486, Jul 2021.
- [7] Masato Kawashima, Ismail Arai, Arata Endo, Masatoshi Kakiuchi, and Kazutoshi Fujikawa. “Origin Destination Estimation Carrying Over Rolling Proximity Identifiers with RSSI”. In *2022 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*, pp. 7–12, Dec 2022.
- [8] Johannes Becker, David Li, and David Starobinski. “Tracking Anonymized Bluetooth Devices”. In *Proceedings on Privacy Enhancing Technologies (PoPET)*, Vol. 2019, pp. 50–65, Jul 2019.
- [9] Thongtatt Oransirikul, Ian Piumarta, and Hideyuki Takada. “Classifying Passenger and Non-passenger Signals in Public Transportation by Analysing

- Mobile Device Wi-Fi Activity”. *Journal of Information Processing*, Vol. 27, pp. 25–32, Jan 2019.
- [10] 厚生労働省. “接触確認アプリ (COCOA) の機能を停止します”. https://www.mhlw.go.jp/stf/newpage_29139.html, 11 2022. 参照: 2025-01-30.
- [11] 剛水野, 芳貴清水, 淳三浦. Bluetooth 無線化講座 プロが教える基礎・開発ノウハウ・よくあるトラブルと対策 . 株式会社技術評論社, 04 2024. pp. 124–125.
- [12] Bluetooth SIG. “Bluetooth Core Specification v5.4”, Jan 2023. Version Date: 2023-01-31. Available: <https://www.bluetooth.com/specifications/specs/core-specification-5-4/>.
- [13] Bluetooth SIG. “Assigned Numbers”, Nov 2024. Version Date: 2024-11-06. Available: <https://www.bluetooth.com/specifications/assigned-numbers/>.
- [14] Jeremy Martin, Douglas Alpuche, Kristina Bodeman, Lamont Brown, Ellis Fenske, Lucas Foppe, Travis Mayberry, Erik Rye, Brandon Sipes, and Sam Teplov. “Handoff All Your Privacy – A Review of Apple’s Bluetooth Low Energy Continuity Protocol”. In *Proceedings on Privacy Enhancing Technologies (PoPET)*, 第 2019 巻, pp. 34–53, Oct 2019.
- [15] Alexander Heinrich, Milan Stute, Tim Kornhuber, and Matthias Hollick. “Who Can Find My Devices? Security and Privacy of Apple’s Crowd-Sourced Bluetooth Location Tracking System”. In *Proceedings on Privacy Enhancing Technologies (PoPET)*, Vol. 2021, pp. 227–245. Sciendo, Mar 2021.
- [16] 社会資本整備審議会 都市計画・歴史的風土分科会 都市計画部会都市交通・市街地整備小委員会. “集約型都市構造を支える公共交通の実現に向けて”. https://www.mlit.go.jp/singikai/infra/city_history/city_planning/city_traffic/h18_9/images/shiryoku3.pdf, 2006.
- [17] 国土交通省. “乗合バス 1 人平均乗車キロ及び国民 1 人当たり利用回数の推移”. <https://www.mlit.go.jp/common/001351819.pdf>, 2020.

- [18] 横浜市都市整備局都市交通部都市交通課. “路線バスの基礎知識”.
https://www.city.yokohama.lg.jp/kurashi/machizukuri-kankyo/kotsu/bus_kotsu/kisochishiki.html. 参照: 2025-01-30.
- [19] 神奈川中央交通. “送迎バスシステム”. <https://www.kanachu.co.jp/bus/courtesy/>. 参照: 2025-01-30.
- [20] 吉村太斗, 林虎太郎, 新井イスマイル, 松田裕貴. “BLE のランダム MAC アドレスを用いた OD データ推定に向けた初期検討”. 2024 年度 情報処理学会 関西支部 支部大会 講演論文集, Sep 2024.
- [21] 警視庁. “自転車の交通ルール”. <https://www.keishicho.metro.tokyo.lg.jp/kotsu/jikoboshi/bicycle/menu/rule.html>. 参照: 2025-02-28.
- [22] Nordic Semiconductor. “nRF Sniffer for Bluetooth LE”.
<https://www.nordicsemi.com/Products/Development-tools/nRF-Sniffer-for-Bluetooth-LE>, 2025. 参照: 2025-01-30.