

ระบบระบุตำแหน่งภายในอาคารสำหรับสมาร์ทโฟนด้วยการใช้เทคโนโลยีลายนิ้วมือของสัญญาณวิทยุ  
เชิงกำหนด



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย  
ปีการศึกษา 2563  
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Indoor Localization System for Smartphone using Deterministic Wi-Fi Fingerprint  
Technique



A Dissertation Submitted in Partial Fulfillment of the Requirements  
for the Degree of Doctor of Philosophy in Computer Engineering

Department of Computer Engineering

FACULTY OF ENGINEERING

Chulalongkorn University

Academic Year 2020

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	ระบบระบุตำแหน่งภายในอาคารสำหรับสมาร์ทโฟนด้วยการใช้เทคนิคคลายน์มือของสัญญาณวิทยุพาเยิงกำหนด
โดย	นายธีรภัทร์ วงศ์สุธีรา
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.กุลธิดา โรจนวิบูลย์ชัย

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของ  
ของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

.....	คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)	
คณะกรรมการสอบวิทยานิพนธ์	ประธานกรรมการ
.....	
(ศาสตราจารย์ ดร.ประภาส จงสฤษดิ์วัฒนา)	
.....	อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.กุลธิดา โรจนวิบูลย์ชัย)	
.....	กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ณัฐภูมิ หนูไพโรจน์)	
.....	กรรมการ
(รองศาสตราจารย์ ดร.เกริก ภิรมย์โสภิต)	
.....	กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.อนันต์ ผลเพิ่ม)	

ธีรภัทร์ วงศ์สุธีรา : ระบบระบุตำแหน่งภายในอาคารสำหรับสมาร์ทโฟนด้วยการใช้  
 เทคนิคลายนิ้วมือของสัญญาณวิทยุเชิงกำหนด. ( Indoor Localization System  
 for Smartphone using Deterministic Wi-Fi Fingerprint Technique) อ.ที่ปรึกษา  
 หลัก : รศ. ดร.กุลธิดา โรจนวิบูลย์ชัย

การระบุตำแหน่งภายในอาคารด้วยการใช้เทคนิคลายนิ้วมือของสัญญาณวิทยุมักจะทำงานอย่างไม่มีประสิทธิภาพเมื่อนำไปใช้งานในพื้นที่ขนาดใหญ่ นอกจากนี้ การระบุตำแหน่งดังกล่าวมักพบกับปัญหาตำแหน่งที่ไม่ตรงกัน ซึ่งทำให้เกิดความไม่แม่นยำในการระบุตำแหน่ง ยิ่งกว่านั้นการเปลี่ยนแปลงของสภาพแวดล้อมก็สามารถลดประสิทธิภาพโดยรวมของระบบได้เช่นกัน เพื่อที่จะแก้ปัญหาเหล่านี้ งานวิจัยชิ้นนี้ได้นำเสนอ ระบบระบุตำแหน่งภายในอาคารสำหรับสมาร์ทโฟนด้วยการใช้เทคนิคลายนิ้วมือของสัญญาณวิทยุเชิงกำหนด ระบบนี้ประกอบด้วยกันทั้งหมด 3 ส่วน ส่วนที่หนึ่งคืออัลกอริทึมสำหรับจัดหมวดหมู่พื้นที่ โดยส่วนนี้จะจำแนกคำร้องขอจากผู้ใช้งาน ว่าถูกส่งมาจากภายนอกอาคารหรือภายในอาคารใด เพื่อรองรับคำร้องขอที่ถูกส่งมาจากพื้นที่ภายนอกอาคารหรือพื้นที่ที่ระบบไม่ได้ครอบคลุมออกไป ส่วนที่สองคืออัลกอริทึมระบุตำแหน่งภายในอาคาร ส่วนนี้จะใช้ข้อมูลจากส่วนแรก ในการลดขอบเขตการค้นหาลายนิ้วมือของสัญญาณลง ทำให้สามารถลดระยะเวลาในการประมวลผลเพื่อหาตำแหน่งที่แน่นอนได้อย่างมหาศาล นอกจากนี้อัลกอริทึมในส่วนนี้จะคำนึงถึงปัญหาตำแหน่งที่ไม่ตรงกัน เพื่อลดผลกระทบจากปัญหาดังกล่าว ส่วนที่สามคืออัลกอริทึมตรวจจับบีเอสเอสไอดีที่หายไปจากคำร้องขอจากผู้ใช้งานและปรับปรุงฐานข้อมูล เพื่อให้ระบบสามารถปรับตัวกับสภาวะแวดล้อมที่เปลี่ยนไปได้จากผลการทดลอง ระบบที่นำเสนอสามารถจัดหมวดหมู่พื้นที่ ระบุตำแหน่งภายในอาคารและตรวจจับบีเอสเอสไอดีที่หายไปได้อย่างแม่นยำ โดยอัลกอริทึมสำหรับจัดหมวดหมู่พื้นที่สามารถลดระยะเวลาในการประมวลผลอย่างเห็นได้ชัดเมื่อเปรียบเทียบกับงานวิจัยที่ผ่านมา นอกจากนี้ อัลกอริทึมตรวจจับบีเอสเอสไอดีที่หายไปสามารถเพิ่มประสิทธิภาพให้กับทั้งระบบที่นำเสนอและงานวิจัยที่ผ่านมาได้อย่างมาก

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ลายมือชื่อนิสิต .....

ปีการศึกษา 2563

ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

# # 5971420821 : MAJOR COMPUTER ENGINEERING

KEYWORD: Area classification, Fingerprint, Indoor localization, Indoor localization system, Large-scale, Wi-Fi

Teerapat Vongsuteera : Indoor Localization System for Smartphone using Deterministic Wi-Fi Fingerprint Technique. Advisor: Assoc. Prof. KULTIDA ROJVIBOONCHAI

Generally, fingerprint-based indoor localization works inefficiently when deployed in a large-scale area. Furthermore, the approach can suffer from non-coincident location problems which lead to inaccurate localization. Moreover, the changing environment can degrade overall performance. To tackle these problems, this research proposes an indoor localization system for smartphone using deterministic Wi-Fi fingerprint technique. The system consists of three main parts. The first part is area classification. It identifies an area of the user's queries whether they are outdoor or located in a specific building. The second part is fingerprint-based indoor localization algorithm. It utilizes the information from the first part by searching only the fingerprint in the specific building. Third, our missing-BSSID detector algorithm detects the missing Basic Service Set Identifiers (BSSIDs) in the incoming query and updates a sampling database. This part is for our system to quickly adapt to the changing environment. According to the evaluation results, the proposed system achieves high accuracy of area classification, indoor localization and missing-BSSIDs detection. The area classification algorithm can significantly reduce the overall processing time compared to the previous work. More importantly, the missing-BSSID detector algorithm improves the overall performance to the proposed system as well as other existing systems.

Field of Study: Computer Engineering

Student's Signature .....

Academic Year: 2020

Advisor's Signature .....

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความอนุเคราะห์จากรองศาสตราจารย์ ดร. กุสธิตา วิจารณ์วิบูลย์ชัย อาจารย์ที่ปรึกษาวิทยานิพนธ์ โดยอาจารย์ได้ให้คำปรึกษาต่าง ๆ สำหรับปรับปรุงและพัฒนางานวิจัย อีกทั้งยังให้คำแนะนำเพื่อช่วยแก้ปัญหาที่เกิดขึ้นระหว่างการดำเนินงานวิจัยอีกด้วย

ขอขอบคุณคณะกรรมการสอบวิทยานิพนธ์ ได้แก่ ศาสตราจารย์ ดร. ประภาส จงสถิตย์วัฒนา ผู้ช่วยศาสตราจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์ รองศาสตราจารย์ ดร. เกริก ภริมย์โสสภา และรองศาสตราจารย์ ดร. อนันต์ ผลเพิ่ม ที่ได้ให้คำแนะนำซึ่งเป็นประโยชน์ต่อวิทยานิพนธ์ฉบับนี้

งานวิจัยชิ้นนี้ได้รับการสนับสนุนทุนการศึกษาจากทุนการศึกษาหลักสูตรดุขภูมิต "๑๐๐ ปี จุฬาลงกรณ์มหาวิทยาลัย" ได้รับการสนับสนุนทุนจาก "ทุน ๙๐ ปี จุฬาลงกรณ์มหาวิทยาลัย กองทุนรัชดาภิเษกสมโภช" และได้รับการสนับสนุนทุนจากหน่วยวิจัยโครงข่ายไร้สายและอินเทอร์เน็ตอนาคต (Wireless Network and Future Internet Research Unit)

ขอขอบคุณสมาชิกทุกคนในห้องปฏิบัติการ ได้แก่ คุณกุลิสร์ ณ นคร คุณเกียรติคุณ กาวิลละคุณธนาภูมิ แดนมณีและคุณอัษฎาวุธ ชนะกิจการโชค ที่ให้ความคิดเห็นและข้อเสนอแนะสำหรับการทำวิจัยตลอดระยะเวลา 5 ปีที่ผ่านมา

สุดท้ายนี้ขอขอบคุณคุณพ่อ คุณแม่ ครอบครัวและคนรอบข้างที่เปิดโอกาสให้ได้ศึกษาต่อ เป็นกำลังใจและให้คำแนะนำตลอดระยะเวลาที่ทำการวิจัย จนกระทั่งสำเร็จการศึกษา

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

ธีรภัทร์ วงศ์สุธีรา

## สารบัญ

	หน้า
.....	ค
บทคัดย่อภาษาไทย.....	ค
.....	ง
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญตาราง.....	ญ
สารบัญภาพ.....	ฎ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	4
1.3 ขอบเขตการวิจัย.....	4
1.4 ขั้นตอนและวิธีการดำเนินการวิจัย.....	5
1.5 ประโยชน์ที่คาดว่าจะได้รับการวิจัย.....	5
1.6 ผลงานตีพิมพ์.....	6
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	7
2.1 ทฤษฎีที่เกี่ยวข้อง.....	7
2.1.1 เทคนิคแบบจำลอง (Model-based).....	7
2.1.1.1 ค่าความแรงสัญญาณ (RSSI: Received Signal Strength Indicator).....	7
2.1.1.2 เวลาในการเดินทาง (ToF: Time of Flight).....	8
2.1.1.3 เวลาในการเดินทางที่แตกต่างกัน (TDoF: Time Different of Flight).....	8

2.1.1.4) เวลาในการเดินทางไปและกลับ (RTof: Return Time of Flight) .....	9
2.1.1.5) องศาที่ได้รับ (AoA: Angle of Arrival) .....	9
2.1.2 เทคนิคลายนิ้วมือของสัญญาณ (Fingerprint-based) .....	9
2.1.2.1) วิธีการเชิงกำหนด .....	11
2.1.2.2) วิธีการเชิงความน่าจะเป็น .....	11
2.2 งานวิจัยที่เกี่ยวข้อง .....	11
2.2.1 งานวิจัยเกี่ยวกับการระบุตำแหน่งภายในอาคาร .....	11
2.2.1.1 งานวิจัยเกี่ยวกับการระบุตำแหน่งภายในอาคารที่ไม่คำนึงถึงปัญหาตำแหน่งที่ไม่ตรงกัน .....	11
อัลกอริทึม DiffHit .....	12
อัลกอริทึม FreeLoc .....	13
2.2.1.2 งานวิจัยเกี่ยวกับการระบุตำแหน่งภายในอาคารที่คำนึงถึงปัญหาตำแหน่งที่ไม่ตรงกัน .....	14
อัลกอริทึม RADAR .....	14
อัลกอริทึม WinIPS .....	16
2.2.2 งานวิจัยเกี่ยวกับการจัดหมวดหมู่พื้นที่ .....	16
ส่วนตรวจจับภายใน/ภายนอกบริเวณ (Inside/outside region detection) .....	17
ส่วนจัดหมวดหมู่พื้นที่ (Area classification module) .....	17
บทที่ 3 ระบบระบุตำแหน่งภายในอาคารที่นำเสนอ .....	18
3.1) ภาพรวมของระบบ .....	18
3.2) โมดูลกรองบีเอสเอสไอทีที่ไม่รู้จัก .....	20
3.3) โมดูลจำแนกในอาคาร/นอกอาคาร .....	21
3.3.1 ช่วงของการฝึก .....	22
3.3.2 ช่วงของการระบุตำแหน่ง .....	25



3.4) โมดูลจำแนกอาคาร.....	25
3.4.1) ช่วงของการฝึก.....	25
3.4.2) ช่วงของการระบุตำแหน่ง.....	26
3.5) อัลกอริทึม InHit.....	27
3.5.1) ช่วงของการฝึก.....	28
3.5.2) ช่วงของการระบุตำแหน่ง.....	29
3.6) อัลกอริทึม MissingHit.....	30
บทที่ 4 การวัดประสิทธิภาพการทำงานของระบบ.....	33
4.1) การตั้งค่าการทดลอง.....	33
4.1.1) การตั้งค่าการทดลองสำหรับอัลกอริทึม ExtHit.....	33
4.1.2) การตั้งค่าการทดลองสำหรับอัลกอริทึม InHit.....	35
4.1.3) การจำลอง.....	36
4.2) มาตรฐาน.....	37
4.2.1) ความแม่นยำ.....	37
4.2.2) ระยะทางผิดพลาด.....	37
4.2.3) เวลาสะสมในการประมวลผล.....	38
4.3) ผลการทดลอง.....	38
4.3.1) โมดูลจำแนกในอาคาร/นอกอาคาร.....	38
4.3.2) โมดูลจำแนกอาคาร.....	44
4.3.3) อัลกอริทึม InHit.....	49
4.3.4) อัลกอริทึม MissingHit.....	56
4.3.5) การเปรียบเทียบกับระบบจำแนกพื้นที่อื่น ๆ.....	59
4.3.6) การเปรียบเทียบกับระบบระบุตำแหน่งภายในอาคารอื่น ๆ.....	61
4.3.7) เวลาในการประมวลผลของระบบ.....	63

บทที่ 5 บทสรุปของงานวิจัยและอภิปรายผลการวิจัย .....	67
5.1 สรุปผลการวิจัย .....	67
5.2 ข้อเสนอแนะ .....	67
5.2.1) การใช้ทรัพยากรของระบบ .....	67
5.2.2) ความทนทานของระบบต่อสัญญาณรบกวน .....	69
5.2.3) การใช้งานระบบในอาคารอื่นๆ .....	69
5.2.4) ขนาดของข้อมูลที่ถูกส่ง .....	70
5.2.5) ตำแหน่งอ้างอิงในการระบุตำแหน่ง .....	70
5.2.6) แนวทางพัฒนาระบบในอนาคต .....	70
บรรณานุกรม .....	71
ประวัติผู้เขียน .....	76



## สารบัญตาราง

หน้า

ตารางที่ 1 จำนวนข้อมูลลักษณะสัญญาณวายพายและรายละเอียดของแต่ละชุดข้อมูลสำหรับทดสอบ ExtHit .....	34
ตารางที่ 2 จำนวนข้อมูลลักษณะสัญญาณวายพายและรายละเอียดของแต่ละชุดข้อมูลสำหรับทดสอบ InHit .....	36
ตารางที่ 3 ความแม่นยำสูงสุดและค่าคงที่ที่ใช้ของโมดูลจำแนกในอาคาร/นอกอาคาร .....	40
ตารางที่ 4 ความแม่นยำสูงสุดและค่าคงที่ที่ใช้ของโมดูลจำแนกในอาคาร/นอกอาคารเมื่อสร้างลายนิ้วมือของสัญญาณและทดสอบด้วยเครื่องที่แตกต่างกัน .....	42
ตารางที่ 5 ความแม่นยำของโมดูลจำแนกในอาคาร/นอกอาคารในสถานการณ์ที่บีเอสเอสไอดีหายไป .....	43
ตารางที่ 6 ความแม่นยำของโมดูลจำแนกในอาคาร/นอกอาคารในสถานการณ์ที่ค่าความแรงสัญญาณแปรปรวน .....	44
ตารางที่ 7 ความแม่นยำสูงสุดและค่าคงที่ที่ใช้ของโมดูลจำแนกอาคาร .....	45
ตารางที่ 8 ความแม่นยำของโมดูลจำแนกอาคารในสถานการณ์ที่บีเอสเอสไอดีหายไป .....	46
ตารางที่ 9 ความแม่นยำของโมดูลจำแนกอาคารในสถานการณ์ที่ค่าความแรงสัญญาณแปรปรวน ...	47
ตารางที่ 10 ความแม่นยำสูงสุดและค่าคงที่ที่ใช้ของโมดูลจำแนกอาคารในสถานการณ์ที่สภาพแวดล้อมเปลี่ยนไป .....	49
ตารางที่ 11 ความแม่นยำสูงสุดและค่าคงที่ที่ใช้ของอัลกอริทึม InHit ในการระบุชั้น .....	50
ตารางที่ 12 ระยะทางผิดพลาดเฉลี่ยน้อยสุดของอัลกอริทึม InHit เมื่อค่า k เป็น 1 .....	51
ตารางที่ 13 ระยะทางผิดพลาดเฉลี่ยน้อยสุดของอัลกอริทึม InHit เมื่อสร้างลายนิ้วมือของสัญญาณและทดสอบด้วยเครื่องที่แตกต่างกัน .....	53
ตารางที่ 14 ระยะทางผิดพลาดเฉลี่ยของอัลกอริทึม InHit ในสถานการณ์ที่บีเอสเอสไอดีหายไป .....	55
ตารางที่ 15 ระยะทางผิดพลาดเฉลี่ยของอัลกอริทึม InHit ในสถานการณ์ที่ค่าความแรงของสัญญาณแปรปรวน .....	55

ตารางที่ 16 อัตราการตรวจจับปีเอสเอสไอดีที่หายไปของอัลกอริทึม MissingHit .....	56
ตารางที่ 17 การเปรียบเทียบความแม่นยำระหว่างโมดูลจำแนกอาคารที่ไม่ได้ทำงานร่วมกับ MissingHit และโมดูลจำแนกอาคารที่ทำงานร่วมกับ MissingHit.....	59
ตารางที่ 18 ความแม่นยำในการระบุชั้นของ RADAR และ WinIPS เปรียบเทียบกับ InHit ในสถานการณ์ที่ปีเอสเอสไอดีหายไป .....	62
ตารางที่ 19 เวลาในการประมวลผลสะสมและผลลัพธ์การระบุตำแหน่งของแต่ละสถาปัตยกรรม ....	65
ตารางที่ 20 ความซับซ้อนของอัลกอริทึมสำหรับแต่ละโมดูล .....	68



## สารบัญภาพ

	หน้า
ภาพที่ 1 ตัวอย่างของการระบุตำแหน่งด้วยเทคนิคแบบจำลอง.....	7
ภาพที่ 2 ตัวอย่างของการระบุตำแหน่งด้วยเทคนิคलयนิ้วมือของสัญญาณ.....	10
ภาพที่ 3 ตัวอย่างลายนิ้วมือของสัญญาณของอัลกอริทึม FreeLoc.....	13
ภาพที่ 4 ตัวอย่างลายนิ้วมือของสัญญาณของระบบ RADAR.....	15
ภาพที่ 5 สถาปัตยกรรมของระบบที่นำเสนอ.....	19
ภาพที่ 6 ผลลัพธ์การทดลองที่แสดงลักษณะสัญญาณภายในอาคารและภายนอกอาคาร.....	22
ภาพที่ 7 พื้นที่ยานนิทรรศการ.....	34
ภาพที่ 8 ภาพถ่ายอาคารวิศวกรรมร้อยปี.....	35
ภาพที่ 9 ภาพถ่ายอาคารวิศวกรรมศาสตร์ 3.....	36
ภาพที่ 10 แผนผังอาคารวิศวกรรมศาสตร์ 3.....	36
ภาพที่ 11 ความแม่นยำของโมดูลจำแนกในอาคาร/นอกอาคาร.....	40
ภาพที่ 12 ความแม่นยำของโมดูลจำแนกในอาคาร/นอกอาคารเมื่อสร้างลายนิ้วมือของสัญญาณและทดสอบด้วยเครื่องที่แตกต่างกัน.....	42
ภาพที่ 13 ความแม่นยำของโมดูลจำแนกอาคาร.....	45
ภาพที่ 14 การเปลี่ยนแปลงของจำนวนบีเอสเอสไอทีระหว่างก่อนวันงานนิทรรศการและระหว่างวันงานนิทรรศการ.....	48
ภาพที่ 15 ความแม่นยำของโมดูลจำแนกอาคารในสถานการณ์จริงที่สภาพแวดล้อมเปลี่ยนแปลง... ..	48
ภาพที่ 16 ความแม่นยำในการระบุชั้นของอัลกอริทึม InHit.....	50
ภาพที่ 17 ระยะทางผิดพลาดเฉลี่ยของอัลกอริทึม InHit เมื่อค่า k เป็น 1.....	51
ภาพที่ 18 ระยะทางผิดพลาดเฉลี่ยของอัลกอริทึม InHit ในกรณีค่า k ต่าง ๆ.....	52
ภาพที่ 19 ฟังก์ชันการแจกแจงความน่าจะเป็นแบบสะสมของระยะทางผิดพลาด.....	52

ภาพที่ 20 ระยะทางผิดพลาดเฉลี่ยของอัลกอริทึม InHit เมื่อสร้างลายนิ้วมือของสัญญาณและทดสอบด้วยเครื่องที่แตกต่างกัน..... 53

ภาพที่ 21 ระยะทางผิดพลาดเฉลี่ยของอัลกอริทึม InHit เมื่อสร้างลายนิ้วมือของสัญญาณและทดสอบด้วยเครื่องที่แตกต่างกัน ในกรณีค่า k ต่าง ๆ ..... 54

ภาพที่ 22 ความแม่นยำของโมดูลจำแนกในอาคาร/นอกอาคารที่ทำงานร่วมกับ MissingHit ในกรณีที่มีปีเอสเอสไอดีหายไปร้อยละ 50 ..... 57

ภาพที่ 23 การเปรียบเทียบความแม่นยำระหว่างโมดูลจำแนกในอาคาร/นอกอาคารที่ไม่ได้ทำงานร่วมกับ MissingHit และโมดูลจำแนกในอาคาร/นอกอาคารที่ทำงานร่วมกับ MissingHit..... 58

ภาพที่ 24 ความแม่นยำของโมดูลระบุใน/นอกพื้นที่ของระบบที่นำเสนอใน [26] ในสถานการณ์ที่มีปีเอสเอสไอดีหายไป..... 60

ภาพที่ 25 ความแม่นยำของโมดูลจำแนกพื้นที่ของระบบที่นำเสนอใน [26] ในสถานการณ์ที่มีปีเอสเอสไอดีหายไป ..... 61

ภาพที่ 26 ระยะทางผิดพลาดเฉลี่ยในการระบุตำแหน่งของ RADAR และ WinIPS เปรียบเทียบกับ InHit ในสถานการณ์ที่มีปีเอสเอสไอดีหายไป ..... 63

ภาพที่ 27 เวลาในการประมวลผลสะสมของแต่ละสถาปัตยกรรม ..... 65

ภาพที่ 28 แผนที่ความร้อนแสดงคำร้องขอจากผู้ใช้งานในช่วงเวลาที่ต่างกัน ..... 66

## บทที่ 1 บทนำ

### 1.1 ที่มาและความสำคัญของปัญหา

การเติบโตของสมาร์ทโฟนส่งผลต่อการเพิ่มขึ้นของความต้องการบริการด้านการระบุตำแหน่ง ซึ่งหนึ่งในเทคโนโลยีที่มีชื่อเสียงมากที่สุดสำหรับการระบุตำแหน่งคือเทคโนโลยีจีพีเอส (GPS: Global Positioning System) เนื่องจากจีพีเอสสามารถระบุตำแหน่งของผู้ใช้งานได้อย่างแม่นยำเมื่ออยู่ภายนอกอาคาร แต่อย่างไรก็ตาม เมื่อผู้ใช้งานอยู่ภายในอาคาร สัญญาณของจีพีเอสจะถูกกีดกันด้วย กำแพงและหลังคาของอาคาร จึงทำให้อุปกรณ์ต่าง ๆ ไม่สามารถใช้งานเทคโนโลยีจีพีเอสเมื่ออยู่ภายในอาคารได้

ปัญหาที่กล่าวมาข้างต้น ทำให้มีงานวิจัยจำนวนมากประยุกต์เทคโนโลยีต่าง ๆ เพื่อแก้ไข ปัญหาเกี่ยวกับการระบุตำแหน่งภายในอาคาร โดยเทคโนโลยีบางชนิดจำเป็นที่จะต้องติดตั้งโครงสร้าง พื้นฐานก่อนการใช้งานระบบ เช่น บลูทูธ (Bluetooth) [1-4], อาร์เอฟไอดี (RFID: Radio frequency identification) [5-9] และ สัญญาณเสียง (Acoustic signal) [10] เป็นต้น ซึ่งส่งผลให้มี ค่าใช้จ่ายที่สูงในการเปิดใช้งานระบบระบุตำแหน่ง เพื่อหลีกเลี่ยงค่าใช้จ่ายที่สูงในการเปิดใช้งานระบบ จึงได้มีการประยุกต์ใช้เทคโนโลยีอื่น ๆ ที่สามารถใช้โครงสร้างพื้นฐานที่มีอยู่แล้วหรือไม่จำเป็นต้อง ติดตั้งโครงสร้างพื้นฐานในการระบุตำแหน่งภายในอาคาร เช่น เทคโนโลยีเซ็นเซอร์วัดความเร่ง (Accelerometer) และ เซ็นเซอร์วัดการหมุน (Gyro sensor) [11-14] ในการนับจำนวนก้าวและทิศทางการเดินของผู้ใช้งาน แต่อย่างไรก็ตาม แม้ว่าการใช้เซ็นเซอร์เหล่านี้จะไม่ต้องติดตั้งโครงสร้าง พื้นฐาน แต่จะพบกับปัญหาความผิดพลาดสะสม (Cumulative error problem) [15] ซึ่งส่งผล โดยตรงต่อความแม่นยำในการระบุตำแหน่งของผู้ใช้งาน เทคโนโลยีจีเอสเอ็ม (GSM: Global system for mobile) [16, 17] ใช้โครงสร้างพื้นฐานที่มีอยู่แล้วสำหรับการสื่อสารของโทรศัพท์มือถือในการ ระบุตำแหน่งของผู้ใช้งาน ซึ่งเทคโนโลยีนี้ให้ความแม่นยำในระดับปานกลาง (4 – 5 เมตร) แต่อย่างไร ก็ตาม ความแม่นยำของการใช้จีเอสเอ็มในการระบุตำแหน่งนั้นขึ้นอยู่กับผู้ให้บริการและสมาร์ทโฟน เนื่องจากผู้ให้บริการแต่ละรายจะมีความหนาแน่นของเสากระจายสัญญาณที่ไม่เท่ากัน ซึ่งส่งผล โดยตรงกับความเข้มข้นของสัญญาณที่จะนำมาระบุตำแหน่ง นอกจากนี้สมาร์ทโฟนบางรุ่นจำกัดข้อมูล จีเอสเอ็มที่ผู้พัฒนาจะสามารถเข้าถึงได้ [16] จึงส่งผลให้สมาร์ทโฟนรุ่นเหล่านั้น ไม่สามารถใช้ เทคโนโลยีจีเอสเอ็มในการระบุตำแหน่งได้ เทคโนโลยีสนามแม่เหล็กโลก (Geomagnetic) [18, 19] จะวัดการถูกรบกวนของสนามแม่เหล็กโลกจากวัสดุเหล็กต่าง ๆ ภายในอาคาร เช่น เหล็กที่ฝังตัวอยู่ใน โครงสร้างอาคาร, โตะและเก้าอี้ เป็นต้น อย่างไรก็ตาม ตำแหน่งของเหล็กต่าง ๆ ภายในอาคาร สามารถถูกเปลี่ยนแปลงหรือเคลื่อนย้ายได้ ซึ่งส่งผลให้ความแม่นยำของการระบุตำแหน่งลดลงอย่าง มาก เทคโนโลยีไวไฟ (Wi-Fi) [20-27] เป็นเทคโนโลยีที่ได้รับความนิยมสูงที่สุดในการนำมา

ประยุกต์เพื่อระบุตำแหน่งภายในอาคาร [28-31] เนื่องจากให้ความแม่นยำในการระบุตำแหน่งสูงและสามารถใช้โครงสร้างพื้นฐานวางสายที่มีอยู่แล้วภายในอาคารมาระบุตำแหน่งได้ จึงทำให้มีค่าใช้จ่ายที่ถูกในการเปิดใช้งานระบบ นอกจากนี้สมาร์ตโฟนของผู้ใช้งานนั้นมีส่วนต่อประสานวางสาย (Wi-Fi interface) ฝังตัวอยู่ ทำให้ผู้ใช้งานไม่จำเป็นต้องถืออุปกรณ์พิเศษใด ๆ ในการใช้งานระบบ

งานวิจัยเกี่ยวกับการระบุตำแหน่งภายในอาคารโดยใช้เทคโนโลยีวางสายส่วนใหญ่ประยุกต์ใช้เทคนิคลายนิ้วมือของสัญญาณ (Signal fingerprint) โดยเทคนิคนี้มีสมมติฐานว่าในแต่ละพื้นที่หรือตำแหน่ง จะมีลักษณะสัญญาณหรือลายนิ้วมือของสัญญาณเป็นเอกลักษณ์เฉพาะพื้นที่หรือตำแหน่งนั้น ๆ ซึ่งกระบวนการของเทคนิคลายนิ้วมือของสัญญาณนี้สามารถแบ่งออกได้เป็นสองช่วงหลัก ๆ คือ ช่วงของการฝึก (Training phase) และช่วงของการระบุตำแหน่ง (Localization phase) โดยช่วงของการฝึกจะมีการสำรวจลักษณะสัญญาณวางสายในพื้นที่ที่สนใจ เพื่อนำมาสร้างลายนิ้วมือของสัญญาณสำหรับทุก ๆ พื้นที่เหล่านั้น และเก็บไว้ในฐานข้อมูล (Database) ส่วนในช่วงของการระบุตำแหน่ง ผู้ใช้งานจะวัดสัญญาณวางสายจากสิ่งแวดล้อมขณะปัจจุบันและส่งผลลัพธ์การวัดสัญญาณนั้นเป็นคำร้องมาที่เซิร์ฟเวอร์ (Server) หลังจากนั้นอัลกอริทึม (Algorithm) สำหรับการระบุตำแหน่งจะเปรียบเทียบผลการวัดสัญญาณจากผู้ใช้งานกับทุก ๆ ลายนิ้วมือของสัญญาณที่อยู่ในฐานข้อมูล เพื่อทำนายและคืนค่าตำแหน่งปัจจุบันกลับไปยังผู้ใช้งาน

วิธีการที่เทคนิคลายนิ้วมือของสัญญาณวางสายนำมาประยุกต์ใช้ สามารถแบ่งออกได้เป็นสองวิธีการ [32-35] คือ วิธีการเชิงกำหนด (Deterministic approach) [20, 21, 23, 25-27] และวิธีการเชิงความน่าจะเป็น (Probabilistic approach) [22] โดยวิธีการเชิงกำหนดจะใช้การวัดความคล้ายคลึงระหว่างลายนิ้วมือของสัญญาณในการเปรียบเทียบหาตำแหน่ง ส่วนวิธีการเชิงความน่าจะเป็นจะใช้การอนุมานทางสถิติในเปรียบเทียบหาตำแหน่ง ด้วยเหตุนี้วิธีการเชิงความน่าจะเป็นจะใช้จำนวนครั้งในการวัดค่าสัญญาณต่อตำแหน่งมากกว่าวิธีการเชิงกำหนด ส่งผลให้ความแม่นยำของระบบในการระบุตำแหน่งทนต่อสัญญาณรบกวนมากกว่า แต่ต้องใช้เวลาและแรงงานในการสำรวจลักษณะสัญญาณมากกว่าเช่นกัน

แต่อย่างไรก็ตามการระบุตำแหน่งโดยใช้เทคนิคลายนิ้วมือของสัญญาณวางสาย มักพบกับปัญหาตำแหน่งที่ไม่ตรงกัน (Non-coincident location problem) กล่าวคือหากกระบวนการค้นหาตำแหน่งของอัลกอริทึมเป็นการใช้ตำแหน่งของลายนิ้วมือของสัญญาณที่มีความคล้ายคลึงที่สุดหรือความน่าจะเป็นสูงที่สุดกับคำร้องของผู้ใช้งาน [20, 23] เพียงตำแหน่งเดียวเท่านั้นในการอ้างอิงเพื่อหาตำแหน่ง จะทำให้เกิดความคลาดเคลื่อนในการระบุตำแหน่งในกรณีที่ผู้ใช้งานร้องขอจากตำแหน่งที่ไม่ได้สำรวจลักษณะสัญญาณ ซึ่งวิธีการลดปัญหาดังกล่าวสามารถทำได้โดยการเพิ่มจำนวนตำแหน่งที่ต้องสำรวจลักษณะสัญญาณให้มากขึ้น แต่อย่างไรก็ตามวิธีการแก้ปัญหานี้จะทำให้เพิ่มเวลาและแรงงานอย่างมหาศาลในการสำรวจลักษณะสัญญาณ จึงเห็นได้ว่าปัญหาตำแหน่งที่ไม่ตรงกัน ส่งผล



โดยตรงกับความแม่นยำโดยรวมของระบบ เวลาและแรงงานในการสำรวจลักษณะสัญญาณ ดังนั้น กระบวนการค้นหาตำแหน่งของอัลกอริทึมสำหรับการระบุตำแหน่ง จำเป็นที่จะต้องพิจารณาตำแหน่ง จากหลาย ๆ ลายนิ้วมือของสัญญาณที่คล้ายคลึง ไม่ใช่เพียงลายนิ้วมือของสัญญาณที่มีความคล้ายคลึง มากที่สุดเท่านั้น

นอกจากนี้เมื่อประยุกต์เทคนิคลายนิ้วมือของสัญญาณวายพายในพื้นที่ขนาดใหญ่ เช่น งาน แสดงนิทรรศการหรือศูนย์ประชุมขนาดใหญ่ จะพบกับปัญหาดังนี้ 1) ใช้เวลาในการประมวลผลนาน เนื่องจากพื้นที่ที่ระบบครอบคลุมอาจมีหลายชั้นและหลายอาคาร ซึ่งเมื่อขนาดของพื้นที่ใหญ่ขึ้น จึงทำให้จำนวนลายนิ้วมือของสัญญาณเพิ่มขึ้นเช่นเดียวกัน ดังนั้นทำให้อัลกอริทึมใช้เวลาและทรัพยากร มหาศาลในกระบวนการหาตำแหน่งของผู้ใช้งาน 2) สิ้นเปลืองทรัพยากรโดยเปล่าประโยชน์ เนื่องจาก ในทางปฏิบัติผู้ใช้งานสามารถร้องขอตำแหน่งจากที่ใดก็ได้ กล่าวคือ ภายในอาคารหรือภายนอก อาคาร ซึ่งในกรณีที่ผู้ใช้งานร้องขอตำแหน่งจากภายนอกอาคารนั้น เทคนิคลายนิ้วมือของสัญญาณ วายพายจะใช้เวลาและทรัพยากรจำนวนมากในการเปรียบเทียบคำร้องขอจากผู้ใช้งานกับทุก ๆ ลายนิ้วมือของสัญญาณภายในฐานข้อมูลโดยไม่เกิดประโยชน์อะไรกับผู้ใช้งานเลย และในบางกรณีที่ ผู้ใช้งานอยู่ใกล้กับอาคาร ระบบอาจจะคืนค่าตำแหน่งภายในอาคารทั้ง ๆ ที่ผู้ใช้งานอยู่ภายนอก อาคาร 3) ประสิทธิภาพของระบบลดลงเนื่องจากการเปลี่ยนแปลงของสภาพแวดล้อม เพราะ ลายนิ้วมือของสัญญาณวายพายสามารถเปลี่ยนแปลงได้จากหลายสาเหตุ เช่น การติดตั้งหรือถอนการ ติดตั้งตัวกระจายสัญญาณวายพาย ด้วยสาเหตุเหล่านี้จึงสามารถลดความแม่นยำในการระบุตำแหน่ง ด้วยเทคนิคลายนิ้วมือของสัญญาณวายพายได้อย่างมาก

จากปัญหาข้างต้นที่กล่าวมา งานวิจัยชิ้นนี้จึงนำเสนอระบบระบุตำแหน่งภายในอาคารสำหรับ สมาร์ทโฟนด้วยการใช้เทคนิคลายนิ้วมือของสัญญาณวายพายเชิงกำหนด โดยระบบดังกล่าวประกอบ ไปด้วยกันทั้งหมด 3 ส่วน โดยส่วนที่หนึ่งคืออัลกอริทึมสำหรับจัดหมวดหมู่พื้นที่ (Area classification) มีชื่อเรียกว่า “ExtHit” อัลกอริทึมนี้จะทำการระบุพื้นที่ของผู้ใช้งานว่าอยู่ใน อาคารใดหรือภายนอกอาคาร ดังนั้น ExtHit จึงสามารถคัดกรองคำร้องขอจากผู้ใช้งานที่ถูกส่งมาจาก ภายนอกอาคารหรือนอกเหนือจากพื้นที่ที่ระบบครอบคลุมได้ จึงทำให้ระบบสามารถหลีกเลี่ยงการ ค้นหาลายนิ้วมือถือของสัญญาณวายพายอย่างละเอียดจากทั้งฐานข้อมูลได้ นอกจากนี้ ExtHit ยังไม่ จำเป็นต้องใช้ลายนิ้วมือของสัญญาณวายพายจากภายนอกอาคารที่ทำให้ต้องใช้แรงงานและเวลาเพิ่ม มากขึ้นในการสำรวจอีกด้วย ผลจากการทำงานของอัลกอริทึมนี้จะสามารถลดปริมาณข้อมูลที่ต้องถูก ค้นหาและลดเวลาในการประมวลผลได้ เนื่องจาก ExtHit จะจัดกลุ่มลายนิ้วมือของสัญญาณตาม อาคารที่ลายนิ้วมือของสัญญาณนั้น ๆ ถูกเก็บ ส่วนที่สองคืออัลกอริทึมระบุตำแหน่งภายในอาคาร มี ชื่อเรียกว่า “InHit” อัลกอริทึมนี้ใช้ข้อมูลจาก ExtHit ในการลดเวลาและจำนวนข้อมูลที่ต้องค้นหา จากฐานข้อมูลเพื่อระบุตำแหน่งภายในอาคารอย่างละเอียด ส่วนที่สามคืออัลกอริทึมตรวจจับบีเอสเอ

สไอดี (BSSID) ที่หายไป มีชื่อเรียกว่า “MissingHit” อัลกอริทึมนี้จะตรวจจับบีเอสเอสไอดีที่หายไป จากคำร้องขอจากผู้ใช้งานและปรับปรุงฐานข้อมูล ดังนั้นจึงทำให้ฐานข้อมูลลักษณะสัญญาณวิทยุ หรือฐานข้อมูลลายนิ้วมือของสัญญาณวิทยุจะสามารถปรับตัวเข้ากับสภาพแวดล้อมที่เปลี่ยนไปได้

อัลกอริทึม ExtHit ที่นำเสนอมีความยืดหยุ่น เนื่องจากสามารถทำงานร่วมกับอัลกอริทึม ระบุตำแหน่งภายในอาคารอื่น ๆ ที่ใช้เทคนิคลายนิ้วมือของสัญญาณวิทยุได้ ยิ่งไปกว่านั้น อัลกอริทึม MissingHit ก็สามารถทำงานร่วมกับอัลกอริทึมจัดหมวดหมู่พื้นที่และอัลกอริทึมระบุ ตำแหน่งภายในอาคารอื่น ๆ ได้ ระบบที่นำเสนอนี้ถูกทดสอบในงานนิทรรศการขนาดใหญ่ ซึ่ง ประกอบไปด้วยอาคารทั้งหมด 37 อาคาร ครอบคลุมพื้นที่มากกว่า 486,000 ตารางเมตรและคำร้อง ขอจากผู้ใช้งานมากกว่า 600,000 ครั้ง นอกจากนี้ระบบได้ถูกทดสอบบนการจำลองเพื่อวัดผลใน สถานการณ์ที่มีสภาพแวดล้อมเปลี่ยนแปลงอย่างวิฤต จากผลการทดลอง ระบบที่นำเสนอสามารถ ระบุตำแหน่งภายในอาคารได้อย่างแม่นยำ และ ExtHit สามารถลดระยะเวลาในการประมวลผลของ ระบบอย่างเห็นได้ชัด นอกจากนี้ MissingHit ก็สามารถเพิ่มความแม่นยำให้กับ ExtHit, InHit, อัลกอริทึมจัดหมวดหมู่พื้นที่อื่น ๆ และอัลกอริทึมระบุตำแหน่งภายในอาคารอื่น ๆ ได้อย่างเห็นได้ชัด

## 1.2 วัตถุประสงค์ของการวิจัย

- 1.2.1 เพื่อพัฒนาระบบระบุตำแหน่งภายในอาคารโดยใช้สัญญาณวิทยุ
- 1.2.2 เพื่อพัฒนาวิธีการจัดหมวดหมู่พื้นที่เพื่อให้สามารถระบุตำแหน่งได้อย่างมี ประสิทธิภาพ
- 1.2.3 เพื่อพัฒนาวิธีการระบุตำแหน่งให้ได้ความแม่นยำสูงโดยใช้เวลาและแรงงานน้อย
- 1.2.4 เพื่อพัฒนาวิธีการปรับปรุงฐานข้อมูลสัญญาณวิทยุให้สามารถปรับตัวกับ สภาพแวดล้อมที่เปลี่ยนแปลงได้

## 1.3 ขอบเขตการวิจัย

- 1.3.1 ระบบที่นำเสนอใช้วิธีการลายนิ้วมือของสัญญาณวิทยุเชิงกำหนด
- 1.3.2 ระบบที่นำเสนอถูกออกแบบมาสำหรับแอปพลิเคชัน (Application) ของมนุษย์
- 1.3.3 ในกรณีของการจัดหมวดหมู่พื้นที่นั้น ระบบที่นำเสนอสามารถลดขอบเขตในการ ค้นหาและเวลาในการประมวลผลสำหรับอัลกอริทึมระบุตำแหน่งภายในอาคาร
- 1.3.4 ในกรณีของการระบุตำแหน่งภายในอาคาร ระบบที่นำเสนอสามารถใช้เวลาและ แรงงานที่น้อยลงในการสำรวจลักษณะสัญญาณได้ ในขณะที่ยังให้ความแม่นยำในการระบุตำแหน่งที่ สูง

1.3.5 ในกรณีของการตรวจจับบีเอสเอสไอดีที่หายไป ระบบที่นำเสนอสามารถใช้บีเอสเอสไอดีที่ตรวจพบว่าหายไปในการปรับปรุงฐานข้อมูลของสัญญาณวยพายได้

1.3.6 ข้อมูลสำหรับการวัดผลระบบที่นำเสนอจะถูกเก็บด้วยสมาร์ตโฟนหลายรุ่น

#### 1.4 ขั้นตอนและวิธีการดำเนินการวิจัย

1.4.1 ศึกษางานวิจัยที่เกี่ยวข้องกับการระบุตำแหน่งภายในอาคารและการจัดหมวดหมู่พื้นที่

1.4.2 สำรวจลักษณะของสัญญาณวยพายภายในอาคารและนอกอาคาร

1.4.3 พัฒนาอัลกอริทึมสำหรับการจัดหมวดหมู่พื้นที่

1.4.4 ทำการทดลองเพื่อวัดความแม่นยำของอัลกอริทึมสำหรับการจัดหมวดหมู่พื้นที่

1.4.5 สรุปและอภิปรายผลการทดลอง

1.4.6 พัฒนาอัลกอริทึมสำหรับการระบุตำแหน่งภายในอาคาร

1.4.7 ทำการทดลองเพื่อวัดความแม่นยำของอัลกอริทึมสำหรับการระบุตำแหน่งภายในอาคาร

1.4.8 สรุปและอภิปรายผลการทดลอง

1.4.9 พัฒนาอัลกอริทึมสำหรับตรวจจับบีเอสเอสไอดีที่หายไป

1.4.10 ทำการทดลองเพื่อวัดความแม่นยำของอัลกอริทึมสำหรับตรวจจับบีเอสเอสไอดีที่หายไป

1.4.11 สรุปผลและอภิปรายผลการทดลอง

1.4.12 พัฒนาระบบสำหรับการจัดหมวดหมู่พื้นที่และการระบุตำแหน่งภายในอาคาร โดยมีการตรวจจับบีเอสเอสไอดีที่หายไปเพื่อปรับปรุงฐานข้อมูลของสัญญาณวยพาย

1.4.13 ทำการทดลองเพื่อวัดความแม่นยำของระบบเมื่ออยู่ภาวะที่สภาพแวดล้อมเปลี่ยนแปลง

1.4.14 สรุปและอภิปรายผลการทดลอง

1.4.15 ทำการทดลองเพื่อวัดความเร็วของอัลกอริทึมสำหรับการระบุตำแหน่งภายในอาคารเมื่อมีการจัดหมวดหมู่พื้นที่

1.4.16 สรุปและอภิปรายผลการทดลอง

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

1.5.1 ได้ระบบระบุตำแหน่งภายในอาคารโดยใช้สัญญาณวยพาย

1.5.2 ได้วิธีการจัดหมวดหมู่พื้นที่เพื่อให้สามารถระบุตำแหน่งได้อย่างมีประสิทธิภาพ

1.5.3 ได้วิธีการระบุตำแหน่งให้ได้ความแม่นยำสูงโดยใช้เวลาและแรงงานน้อย

1.5.4 ได้วิธีการตรวจจับบีเอสเอสไอดีเพื่อให้สามารถปรับปรุงฐานข้อมูลของสัญญาณวิทยุไฟให้ปรับตัวกับสภาพแวดล้อมที่เปลี่ยนไปได้

## 1.6 ผลงานตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้ตีพิมพ์เป็นบทความในวารสารระดับนานาชาติ ในบทความชื่อ “Vongsuteera, T., & Rojviboonchai, K. (2021). Adaptive Indoor Localization System for Large-Scale Area. IEEE Access, 9, 8847-8865.”



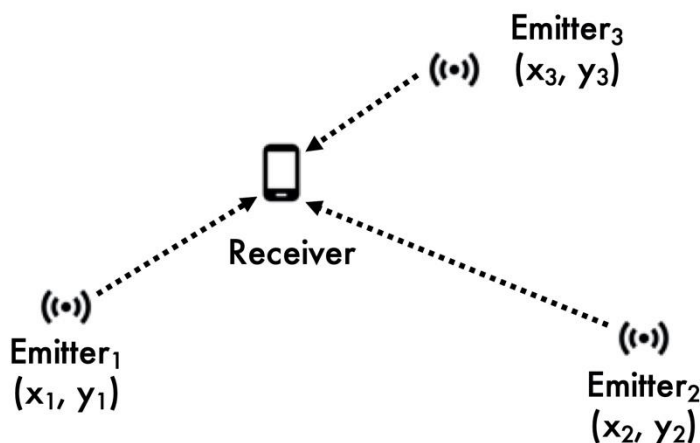
## บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

### 2.1 ทฤษฎีที่เกี่ยวข้อง

เทคนิคที่ถูกประยุกต์ใช้ในการระบุตำแหน่งภายในอาคารโดยใช้สัญญาณวิทยุสามารถแบ่งได้เป็น 2 ประเภท ดังต่อไปนี้

#### 2.1.1 เทคนิคแบบจำลอง (Model-based)

วิธีการของเทคนิคนี้จะใช้แบบจำลองทางคณิตศาสตร์ในการคำนวณระยะทางระหว่างตัวกระจายสัญญาณ (Emitter) อย่างน้อย 3 ตัวและตัวรับสัญญาณ (Receiver) ซึ่งในกรณีนี้ตัวกระจายสัญญาณคือวิทยุแอ็คเซสพอยต์ (Wi-Fi access point) และตัวรับสัญญาณคือสมาร์ทโฟน การระบุตำแหน่งด้วยวิธีนี้ จำเป็นที่จะต้องรู้ตำแหน่งที่แน่นอนของวิทยุแอ็คเซสพอยต์เพื่อเป็นจุดอ้างอิงในการนำมาคำนวณระยะทาง ตัวอย่างของการระบุตำแหน่งด้วยเทคนิคแบบจำลองแสดงได้ดังภาพที่ 1 โดย Emitter<sub>1</sub>, Emitter<sub>2</sub> และ Emitter<sub>3</sub> คือวิทยุแอ็คเซสพอยต์ ซึ่งมี  $(x_1, y_1)$ ,  $(x_2, y_2)$  และ  $(x_3, y_3)$  คือพิกัดของวิทยุแอ็คเซสพอยต์ตามลำดับและ Receiver คือสมาร์ทโฟน ข้อมูลที่สามารถนำมาใช้ประโยชน์ในการคำนวณหาระยะทางมีดังต่อไปนี้



ภาพที่ 1 ตัวอย่างของการระบุตำแหน่งด้วยเทคนิคแบบจำลอง

##### 2.1.1.1 ค่าความแรงสัญญาณ (RSSI: Received Signal Strength Indicator)

คือค่าของความแรงสัญญาณที่ตัวรับสัญญาณหรือสมาร์ทโฟนได้รับ โดยค่าของความแรงสัญญาณที่ได้รับค่าจะแปรผกผันกับระยะทางระหว่างตัวส่งสัญญาณและตัวรับสัญญาณ กล่าวคือ ยิ่งได้รับค่าความแรงสัญญาณมาก ระยะทางระหว่างตัวส่งสัญญาณและตัวรับสัญญาณยิ่งน้อย ซึ่งสามารถแปลงค่าความแรงสัญญาณเป็นระยะทางได้โดยใช้

แบบจำลองอย่างง่ายของการลดทอนการเผยแพร่ (Simple path-loss propagation model) [37] ได้ตั้งสมการที่ 1

$$RSSI = -10n \log_{10}(d) + A \quad (1)$$

โดย  $RSSI$  คือค่าความแรงสัญญาณที่ตัวรับได้รับ,  $n$  คือตัวแทนการลดทอน (มีค่าตั้งแต่ 2 เมื่ออยู่ในพื้นที่โล่ง ถึง 4 เมื่ออยู่ภายในอาคาร),  $d$  คือระยะทางระหว่างตัวส่งสัญญาณและตัวรับสัญญาณและ  $A$  คือค่าความแรงสัญญาณที่ตัวรับได้รับ ณ ตำแหน่งของตัวส่งสัญญาณ

#### 2.1.1.2) เวลาในการเดินทาง (ToF: Time of Flight)

หรือ เวลาที่เดินทางมาถึง (ToA: Time of Arrival) คือระยะเวลาที่สัญญาณใช้ในการเดินทางจากตัวส่งสัญญาณมายังตัวรับสัญญาณ โดยระยะเวลาดังกล่าวเมื่อนำมาคูณกับค่าความเร็วของการเคลื่อนที่สัญญาณ (ความเร็วแสง =  $3 \times 10^8$  เมตร/วินาที) จะได้เป็นระยะทางระหว่างตัวส่งสัญญาณและตัวรับสัญญาณ

จากการคำนวณดังกล่าวจะเห็นได้ว่าการใช้ข้อมูลเวลาในการเดินทางของสัญญาณนั้น จำเป็นที่จะต้องทำการประสานนาฬิกา (Clock synchronization) ด้วยความแม่นยำระดับนาโนวินาทีระหว่างตัวส่งสัญญาณและตัวรับสัญญาณด้วย เนื่องจากสัญญาณเคลื่อนที่ด้วยความเร็วที่สูงมาก ดังนั้นหากเวลาคลาดเคลื่อนเพียงเล็กน้อย จะส่งผลให้การระบุตำแหน่งเกิดความคลาดเคลื่อนได้อย่างมาก

#### 2.1.1.3) เวลาในการเดินทางที่แตกต่างกัน (TDoF: Time Different of Flight)

หรือ เวลาที่เดินทางมาถึงที่แตกต่างกัน (TDoA: Time Different of Arrival) คือความแตกต่างของเวลาที่ใช้ในการเดินทางของสัญญาณจากตัวส่งสัญญาณที่ต่างกัน เมื่อวัดที่ตัวรับสัญญาณ โดยความเร็วในการเคลื่อนที่ของสัญญาณคือความเร็วแสง ซึ่งการใช้ข้อมูลนี้ในการระบุตำแหน่งนั้น จำเป็นที่จะต้องทำการประสานนาฬิกา เช่นเดียวกับการใช้ข้อมูลเวลาในการเดินทาง แต่เป็นการประสานนาฬิการะหว่างตัวส่งสัญญาณด้วยกันเองเท่านั้น

#### 2.1.1.4) เวลาในการเดินทางไปและกลับ (RTof: Return Time of Flight)

คือระยะเวลาทั้งหมดที่สัญญาณใช้ในการเดินทางจากตัวส่งสัญญาณไปยังตัวรับสัญญาณและเดินทางกลับมาที่ตัวส่งสัญญาณ โดยระยะเวลาดังกล่าวเมื่อนำมาคูณกับค่าความเร็วของการเคลื่อนที่สัญญาณ จะได้เป็นสองเท่าของระยะทางระหว่างตัวส่งสัญญาณและตัวรับสัญญาณ โดยข้อดีของการใช้ข้อมูลนี้คือไม่จำเป็นต้องทำการประสานนาฬิกาเลย แต่อย่างไรก็ตาม ถ้าหากการตอบกลับของตัวรับสัญญาณมีความล่าช้าเพียงเล็กน้อย ก็จะทำให้เกิดความคลาดเคลื่อนอย่างมากในการระบุตำแหน่งได้

#### 2.1.1.5) องศาที่ได้รับ (AoA: Angle of Arrival)

จะใช้แถวลำดับของเสาอากาศ (Antennae arrays) ในการประมาณมุมที่ตัวรับสัญญาณนั้นได้รับสัญญาณจากตัวส่งสัญญาณ โดยมุมดังกล่าวสามารถคำนวณได้จากการคำนวณความแตกต่างของเวลาที่ได้รับสัญญาณจากแต่ละเสาอากาศ

จากข้อมูลที่สามารถนำมาใช้ประโยชน์ในการคำนวณข้างต้นจะพบว่า ถ้าหากไม่มีระยะเส้นตรง (Line-of-sight) ระหว่างตัวส่งสัญญาณและตัวรับสัญญาณแล้ว เส้นทางที่สัญญาณเคลื่อนที่จากตัวส่งสัญญาณไปยังตัวรับสัญญาณจะไกลกว่าระยะที่ห่างกันจริง ดังนั้นเมื่อนำมาคำนวณแล้วจะทำให้เกิดความคลาดเคลื่อนในการระบุตำแหน่งได้อย่างมาก จึงทำให้การใช้งานระบบต้องพิจารณา ระยะเส้นตรงด้วย

นอกจากนี้อุปกรณ์วิทยุพายแอนด์เซสพอยต์ที่มีใช้งานอยู่ทั่วไปภายในอาคารนั้น บางรุ่นอาจจะไม่รองรับการประสานนาฬิกาด้วยความแม่นยำในระดับนาโนวินาที จึงทำให้ต้องติดตั้งอุปกรณ์ รวมไปถึงโครงสร้างพื้นฐานใหม่ จึงสรุปได้ว่าการใช้งานเทคนิคแบบจำลองนั้น จำเป็นต้องมีค่าใช้จ่ายสูง และมีความยุ่งยากในการเปิดใช้งานระบบ

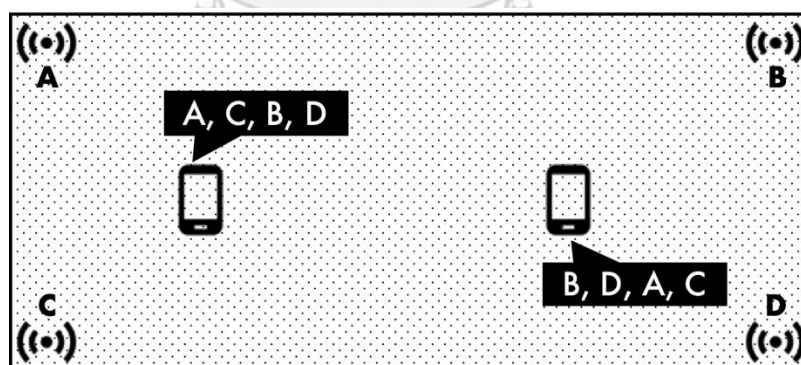
#### 2.1.2 เทคนิคลายนิ้วมือของสัญญาณ (Fingerprint-based)

วิธีการของเทคนิคนี้จะใช้สมมติฐานว่าแต่พื้นที่หรือแต่ละตำแหน่งจะมีลักษณะสัญญาณวิทยุที่แตกต่างกันและเป็นเอกลักษณ์ซึ่งขึ้นอยู่กับสภาพแวดล้อมของพื้นที่นั้น ๆ โดยกระบวนการของเทคนิคจะสามารถแบ่งออกได้เป็นสองช่วงหลัก ๆ ดังนี้

1. ช่วงของการฝึก คือช่วงของการสำรวจและเก็บข้อมูลเกี่ยวกับลักษณะสัญญาณวิทยุในพื้นที่ต่าง ๆ ที่สนใจ หลังจากนั้นจะใช้ข้อมูลดังกล่าวมาสร้างลายนิ้วมือของสัญญาณวิทยุสำหรับแต่ละพื้นที่และเก็บไว้ในฐานข้อมูลสำหรับการระบุตำแหน่งในช่วงต่อไป

2. ช่วงของการระบุตำแหน่ง คือช่วงที่เปิดให้ใช้งานระบบระบุตำแหน่งแล้ว เมื่อผู้ใช้งานต้องการทราบตำแหน่งปัจจุบันของตนเอง จะทำการวัดลักษณะสัญญาณวิทยุจากสิ่งแวดล้อมขณะปัจจุบัน และแนบข้อมูลเหล่านั้นเพื่อร้องขอมาที่เซิร์ฟเวอร์ โดยที่เซิร์ฟเวอร์จะมีอัลกอริทึมสำหรับการคำนวณหาตำแหน่งปัจจุบันของผู้ใช้งาน โดยการเปรียบเทียบข้อมูลจากคำร้องขอนั้นกับทุก ๆ ปลายนิ้วมือของสัญญาณวิทยุในฐานข้อมูลที่ถูกสร้างขึ้นในช่วงของการฝึก หลังจากนั้นอัลกอริทึมจะคืนค่าผลลัพธ์ตำแหน่งโดยพิจารณาจากปลายนิ้วมือของสัญญาณวิทยุที่มีความเหมือนมากหรือความน่าจะเป็นสูงกลับไปหาผู้ใช้งาน

ตัวอย่างของการระบุตำแหน่งด้วยเทคนิคปลายนิ้วมือของสัญญาณแสดงได้ดังภาพที่ 2 โดยกรอบสี่เหลี่ยมคือขอบเขตของพื้นที่ที่สนใจ ในพื้นที่ประกอบไปด้วยวิทยุพาอาศัยเซสพอยต์ 4 ตัวคือ A, B, C และ D ซึ่งในช่วงของการฝึกนั้นได้แบ่งพื้นที่ออกเป็นสองส่วนคือ ส่วนซ้ายของพื้นที่และส่วนขวาของพื้นที่ เมื่อทำการสำรวจลักษณะสัญญาณแล้วจะได้ผลการวัดสัญญาณโดยเรียงลำดับตามความแรงของสัญญาณได้ดังนี้ ฝั่งซ้ายของพื้นที่ ได้ A, C, B และ D ส่วนฝั่งขวาของพื้นที่ได้ B, D, A และ C ต่อมาในช่วงของการระบุตำแหน่ง เมื่อผู้ใช้งานร้องขอตำแหน่งโดยแนบผลการวัดสัญญาณคือ A, C, B และ D จะพบว่าเมื่อเปรียบเทียบกับผลการวัดสัญญาณในช่วงของการฝึกแล้ว ตำแหน่งผู้ใช้งานน่าจะอยู่ฝั่งซ้ายของพื้นที่ จึงคืนผลลัพธ์ตำแหน่งกลับไปยังผู้ใช้งาน



ภาพที่ 2 ตัวอย่างของการระบุตำแหน่งด้วยเทคนิคปลายนิ้วมือของสัญญาณ

วิธีการที่เทคนิคปลายนิ้วมือของสัญญาณวิทยุนำมาประยุกต์ใช้ในการสร้างปลายนิ้วมือของสัญญาณและระบุตำแหน่ง สามารถแบ่งออกได้เป็นสองวิธีการ ดังนี้



### 2.1.2.1) วิธีการเชิงกำหนด

แนวคิดของวิธีการนี้จะใช้การวัดความคล้ายคลึงกันระหว่างคำร้องของผู้ใช้งานกับแต่ละลายนิ้วมือของสัญญาณวายพายในฐานข้อมูล เพื่อพิจารณาว่าผู้ใช้งานอยู่ ณ ตำแหน่งใด โดยผลลัพธ์ตำแหน่งจะขึ้นอยู่กับลายนิ้วมือที่มีความคล้ายสูง ซึ่งในทางปฏิบัติแล้ว ในช่วงของการฝึกจะมีการวัดค่าสัญญาณวายพายไม่เกิน 10 ครั้งต่อหนึ่งตำแหน่งหรือพื้นที่ที่สนใจ เพื่อนำมาสร้างลายนิ้วมือของสัญญาณ ซึ่งลายนิ้วมือของสัญญาณจะมีอย่างน้อยหนึ่งลายนิ้วมือของสัญญาณต่อหนึ่งตำแหน่ง โดยขึ้นกับอัลกอริทึมที่ใช้งาน

### 2.1.2.2) วิธีการเชิงความน่าจะเป็น

แนวคิดของวิธีการนี้ใช้การอนุมานทางสถิติในเปรียบเทียบระหว่างคำร้องของผู้ใช้งานกับแต่ละลายนิ้วมือของสัญญาณวายพายในฐานข้อมูล เพื่อหาลายนิ้วมือของสัญญาณในฐานข้อมูลที่มีความน่าจะเป็นสูง มาพิจารณาหาตำแหน่งของผู้ใช้งาน ซึ่งการอนุมานทางสถิตินั้นจะแม่นยำก็ต่อเมื่อมีการเก็บข้อมูลจำนวนมาก ดังนั้นในทางปฏิบัติในช่วงของการฝึกจะมีการวัดค่าสัญญาณวายพายมากกว่า 100 ครั้งต่อหนึ่งตำแหน่งหรือพื้นที่ที่สนใจ เพื่อนำมาสร้างลายนิ้วมือของสัญญาณ ซึ่งการวัดค่าสัญญาณจำนวนมากจะส่งผลดีในเรื่องของความทนทานต่อสัญญาณรบกวน แต่อย่างไรก็ตาม จะใช้เวลาและแรงงานในการสำรวจมากขึ้นเช่นกัน

จุฬาลงกรณ์มหาวิทยาลัย

## 2.2 งานวิจัยที่เกี่ยวข้อง

หัวข้อนี้นำเสนองานวิจัยที่เกี่ยวข้องกับการระบุตำแหน่งภายในอาคารโดยประยุกต์วิธีการเชิงกำหนดกับเทคนิคลายนิ้วมือของสัญญาณวายพาย ซึ่งสามารถแบ่งงานวิจัยได้เป็นสองกลุ่ม ดังนี้

### 2.2.1 งานวิจัยเกี่ยวกับการระบุตำแหน่งภายในอาคาร

งานวิจัยกลุ่มนี้มีจุดประสงค์ที่จะระบุตำแหน่งที่แน่นอนของผู้ใช้งาน เช่น อยู่ชั้นใดของอาคารและอยู่ ณ ตำแหน่งใดของชั้นนั้น ๆ ซึ่งสามารถแบ่งงานวิจัยกลุ่มนี้ตามกระบวนการในการระบุตำแหน่งได้สองกลุ่ม ดังนี้

#### 2.2.1.1 งานวิจัยเกี่ยวกับการระบุตำแหน่งภายในอาคารที่ไม่คำนึงถึงปัญหาตำแหน่งที่ไม่ตรงกัน

งานวิจัยกลุ่มนี้มีวิธีการในการระบุตำแหน่งของผู้ใช้งานโดยพิจารณาเพียงลายนิ้วมือของสัญญาณที่มีความคล้ายคลึงที่สุด เพียงลายนิ้วมือเดียวเท่านั้น ดังนั้นเมื่อ

ผู้ใช้งานร้องขอตำแหน่งจากตำแหน่งที่ไม่ได้สำรวจในช่วงของการฝึกแล้ว จะทำให้เกิดความคลาดเคลื่อนในการระบุตำแหน่งได้ ตัวอย่างของงานวิจัยในกลุ่มนี้ ได้แก่

#### อัลกอริทึม DiffHit

DiffHit [20] คืออัลกอริทึมสำหรับการระบุตำแหน่งภายในอาคาร โดยมีเป้าหมายในการระบุชั้นปัจจุบันของผู้ใช้งาน หลักการทำงานของ DiffHit สามารถแบ่งออกได้เป็นสองช่วงคือช่วงของการฝึกและช่วงของการระบุตำแหน่ง ในช่วงของการฝึกนั้นคือการสำรวจลักษณะของสัญญาณวิทยุในพื้นที่ต่าง ๆ และจะนำมาสร้างลายนิ้วมือของสัญญาณวิทยุจากข้อมูลเหล่านั้นด้วยการใช้เทคนิคบีเอสเอสไอดี  $N$  ลำดับแรกที่ทำให้ความแรงสัญญาณสูงสุด (Top- $N$  BSSIDs) ซึ่งลายนิ้วมือของสัญญาณวิทยุจะถูกสร้างจากการอ่านค่าสัญญาณวิทยุเพียงหนึ่งครั้งเท่านั้น และแต่ละพื้นที่หรือตำแหน่ง สามารถมีลายนิ้วมือของสัญญาณวิทยุได้มากกว่าหนึ่งลายนิ้วมือ ต่อมาในช่วงของการระบุตำแหน่ง เมื่อผู้ใช้งานร้องขอตำแหน่งด้วยการแนบผลการอ่านค่าวิทยุมา DiffHit จะใช้เทคนิคบีเอสเอสไอดี  $N$  ลำดับแรกกับคำร้องของผู้ใช้งานก่อน แล้วจึงเปรียบเทียบคำร้องขอ นั้นกับแต่ละลายนิ้วมือของสัญญาณที่ได้มาจากช่วงของการฝึก ถ้าหากมีบีเอสเอสไอดี (BSSID) ใดที่ตรงกันระหว่างคำร้องขอของผู้ใช้งานกับลายนิ้วมือของสัญญาณแล้ว ค่าความเหมือน (Hit score) ของลายนิ้วมือของสัญญาณนั้นจะเพิ่มขึ้นดังสมการที่ 2

$$h = N - d \quad (2)$$

โดย  $h$  คือค่าความเหมือนที่เพิ่มขึ้น,  $N$  คือบีเอสเอสไอดี  $N$  ลำดับแรก และ  $d$  คือความแตกต่างของลำดับที่พบบีเอสเอสไอดีนั้นระหว่างคำร้องขอของผู้ใช้งานกับลายนิ้วมือของสัญญาณ เมื่อเปรียบเทียบกับทุกลายนิ้วมือของสัญญาณแล้วจะคืนชั้นของลายนิ้วมือของสัญญาณที่มีค่าความเหมือนมากที่สุดกลับไปหาผู้ใช้งาน ผลการทดลองระบุว่าความแม่นยำสูงสุดของ DiffHit ในการระบุชั้นอยู่ที่ 100% แต่อย่างไรก็ตามความซับซ้อนในการประมวลผลของ DiffHit นั้นขึ้นอยู่กับจำนวนลายนิ้วมือของสัญญาณ ดังนั้นเมื่อ DiffHit ถูกใช้งานในพื้นที่ขนาดใหญ่ จะเกิดการประมวลผลลายนิ้วมือของสัญญาณที่เกินความจำเป็น ซึ่งส่งผลโดยตรงกับประสิทธิภาพโดยรวมของระบบ ทำให้ระบบใช้ทรัพยากรอย่างมหาศาลเกินความจำเป็นและตอบสนองผู้ใช้งานช้าลง

### อัลกอริทึม FreeLoc

FreeLoc [21] มีเป้าหมายในการระบุตำแหน่งปัจจุบันของผู้ใช้งาน โดยหลักการการทำงานของ FreeLoc สามารถแบ่งออกได้เป็นสามช่วงหลักคือช่วงของการฝึก, ช่วงของการบำรุงรักษา และช่วงของการระบุตำแหน่ง โดยในช่วงของการฝึกคือการสำรวจลักษณะของสัญญาณวิทยุในพื้นที่ต่าง ๆ และนำมาสร้างลายนิ้วมือของสัญญาณ ซึ่งในแต่ละพื้นที่หรือตำแหน่งจะมีลายนิ้วมือของสัญญาณเพียงลายนิ้วมือเดียวเท่านั้นที่เป็นตัวแทนของแต่ละพื้นที่ โดยโครงสร้างลายนิ้วมือของสัญญาณจะเป็นลักษณะลำดับของกุญแจ-ความหมาย (key-value) ซึ่งกุญแจคือค่าบีเอสเอสไอดีและความหมายของกุญแจนั้นคือบีเอสเอสไอดีที่มีค่าความแรงสัญญาณต่ำกว่าของกุญแจเกินค่าเดลต้า ( $\delta$ ) ที่กำหนด ตัวอย่างของลายนิ้วมือของสัญญาณแสดงได้ดังภาพที่ 3 โดยผลการวัดค่าสัญญาณประกอบไปด้วยบีเอสเอสไอดี 6 ตัว และเรียงตามความแรงสัญญาณ เมื่อนำมาสร้างลายนิ้วมือของสัญญาณโดยกำหนดค่าเดลต้าเป็น 10 dBm ความหมายของกุญแจ AP1 จะประกอบไปด้วย AP3, AP4, AP5 และ AP6 เนื่องจากบีเอสเอสไอดีทั้ง 4 ตัวนี้มีค่าความแรงสัญญาณน้อยกว่า AP1 เกิน 10 dBm

Wi-Fi scanning result		Fingerprint	
BSSID	RSSI	Key	Value
AP1	-50	AP1	{AP3, AP4, AP5, AP6}
AP2	-55	AP2	{AP3, AP4, AP5, AP6}
AP3	-67	AP3	{AP5, AP6}
AP4	-72	AP4	{AP5, AP6}
AP5	-88	AP5	{}
AP6	-90	AP6	{}

ภาพที่ 3 ตัวอย่างลายนิ้วมือของสัญญาณของอัลกอริทึม FreeLoc

ช่วงของการบำรุงรักษาคือช่วงที่มีการสำรวจลักษณะสัญญาณเพิ่มขึ้นต่อจากช่วงของการฝึก โดยลักษณะสัญญาณที่สำรวจเพิ่มจากช่วงนี้ จะถูกนำไปรวมกับลายนิ้วมือของสัญญาณที่ถูกสร้างไว้แล้ว โดยถ้าหากสำรวจมาจากพื้นที่เดียวกัน และกุญแจตรงกันแล้ว จะผสานความหมายของกุญแจนั้น ๆ เข้าด้วยกันกลายเป็นลายนิ้วมือของสัญญาณที่เป็นตัวแทนของพื้นที่นั้น ๆ และในช่วงของการระบุ

ตำแหน่ง จะแปลงคำร้องของผู้ใช้งานให้อยู่ในรูปแบบเดียวกันกับลายนิ้วมือของสัญญาณคือกฎแฉ-ความหมาย และในการเปรียบเทียบหาคำตอบนั้น การคำนวณค่าความเหมือนของแต่ละลายนิ้วมือของสัญญาณจะเพิ่มเท่ากับจำนวนปีเอสเอสไอดีในความหมายของกฎแฉที่ตรงกัน เมื่อเปรียบเทียบแล้วอัลกอริทึมจะคืนค่าตำแหน่งของลายนิ้วมือของสัญญาณที่มีค่าความเหมือนมากที่สุดกลับไปหาผู้ใช้งาน ผลการทดลองระบุว่าความคลาดเคลื่อนโดยเฉลี่ยของ FreeLoc อยู่ที่ 1.2 เมตร ซึ่งเมื่อพิจารณาแล้ว จะพบว่า FreeLoc จะพบกับปัญหาตำแหน่งที่ไม่ตรงกัน เนื่องจากจะคืนค่าตำแหน่งของลายนิ้วมือของสัญญาณที่มีความเหมือนมากที่สุดเพียงลายนิ้วมือเดียวเท่านั้น ดังนั้นหากผู้ใช้งานร้องขอตำแหน่งจากตำแหน่งที่ไม่ได้เก็บข้อมูลแล้ว FreeLoc จะระบุตำแหน่งผู้ใช้งานนั้นคลาดเคลื่อนอย่างไม่สามารถเลี่ยงได้ นอกจากนี้ความซับซ้อนในการประมวลผลของ FreeLoc นั้นขึ้นอยู่กับจำนวนลายนิ้วมือของสัญญาณด้วย ดังนั้นในพื้นที่ขนาดใหญ่ จะมีลายนิ้วมือของสัญญาณจำนวนมากที่ถูกประมวลผลโดยไม่จำเป็น ทำให้ประสิทธิภาพโดยรวมของระบบลดลง

#### 2.2.1.2 งานวิจัยเกี่ยวกับการระบุตำแหน่งภายในอาคารที่คำนึงถึงปัญหาตำแหน่งที่ไม่ตรงกัน

##### อัลกอริทึม RADAR

RADAR [25] คือระบบระบุตำแหน่งภายในอาคาร โดยหลักการสร้างลายนิ้วมือของนั้นจะแปลงผลวัดสัญญาณให้อยู่ในปริภูมิเวกเตอร์ (Vector space) โดยมีติของเวกเตอร์จะมีค่าเท่ากับจำนวนปีเอสเอสไอดีทั้งหมดที่พบในการสำรวจสัญญาณ ค่าของเวกเตอร์แต่ละตำแหน่งจะมีค่าเท่ากับความแรงสัญญาณของปีเอสเอสไอดีนั้น ตัวอย่างลายนิ้วมือของสัญญาณแสดงได้ดังภาพที่ 4 โดยกำหนดให้การสำรวจสัญญาณวางพายพบปีเอสเอสไอดีทั้งหมด 6 ตัว ดังนั้นเมื่อผลการวัดค่าความแรงสัญญาณได้ AP1 เป็น -50, AP2 เป็น -55 และ AP3 เป็น -67 แล้ว จึงสร้างเวกเตอร์ที่ค่าดัชนี (Index) ของปีเอสเอสไอดีเหล่านั้นเป็น -50, -55 และ -67 ตามลำดับ เป็นลายนิ้วมือของสัญญาณ

Wi-Fi scanning result		Fingerprint	
BSSID	RSSI	Index	Value
AP1	-50	1	-50
AP3	-55	2	0
AP5	-67	3	-55
		4	0
		5	-67
		6	0

ภาพที่ 4 ตัวอย่างลายนิ้วมือของสัญญาณของระบบ RADAR

เมื่อผู้ใช้งานร้องขอตำแหน่ง RADAR จะแปลงคำร้องขอนั้นให้อยู่ในปริภูมิเวกเตอร์ และคำนวณหาระยะห่างระหว่างคำร้องขอนั้นกับแต่ละลายนิ้วมือของสัญญาณโดยใช้การคำนวณระยะทางยูคลิเดียน (Euclidean distance) ดังสมการที่ 3

$$d(s, f) = \sqrt{\sum_{i=1}^n (s_i - f_i)^2} \quad (3)$$

โดย  $d(s, f)$  คือระยะทางยูคลิเดียนระหว่างเวกเตอร์  $s$  และเวกเตอร์  $f$ ,  $s$  คือเวกเตอร์ของคำร้องขอ,  $f$  คือเวกเตอร์ลายนิ้วมือของสัญญาณ,  $s_i$  คือค่าของเวกเตอร์  $s$  ตำแหน่งที่  $i$ ,  $f_i$  คือค่าของเวกเตอร์  $f$  ตำแหน่งที่  $i$  และ  $n$  คือจำนวนบีเอสเอสไอดีทั้งหมดที่พบในการสำรวจสัญญาณ เมื่อทำการคำนวณระยะทางครบทุกลายนิ้วมือของสัญญาณแล้ว จะพิจารณาตำแหน่งของผู้ใช้งานจากตำแหน่งลายนิ้วมือของสัญญาณที่มีระยะทางยูคลิเดียนน้อยที่สุด  $k$  ลำดับแรกมาหาจุดกึ่งกลาง เพื่อคืนผลลัพธ์ตำแหน่งกลับไปยังผู้ใช้งาน โดยความคลาดเคลื่อนโดยเฉลี่ยในการระบุตำแหน่งของ RADAR อยู่ที่ 2.94 เมตร

จะเห็นได้ว่า RADAR จะสามารถลดความคลาดเคลื่อนในการระบุตำแหน่งจากปัญหาตำแหน่งที่ไม่ตรงกันได้ แต่อย่างไรก็ตาม ประสิทธิภาพในการทำงานของ RADAR จะลดลงเมื่อถูกใช้งานในพื้นที่ขนาดใหญ่ เนื่องจากความซับซ้อนในการประมวลผลนั้นขึ้นอยู่กับจำนวนลายนิ้วมือของสัญญาณ

### อัลกอริทึม WinIPS

WinIPS [27] มีหลักการสร้างลายนิ้วมือของสัญญาณที่คล้ายคลึงกับของ RADAR กล่าวคือมีการแปลงผลการวัดสัญญาณวางพายให้เป็นปริภูมิเวกเตอร์ นอกจากนี้ค่าของแต่ละตำแหน่งของเวกเตอร์จะถูกถอดออกด้วยค่าเฉลี่ยและหารด้วยส่วนเบี่ยงเบนมาตรฐานของเวกเตอร์นั้น ๆ ส่วนในช่วงของการระบุตำแหน่งจะทำการคำนวณค่าดัชนีความโน้มเอียงสัญญาณ (STI: Signal Tendency Index) ระหว่างคำร้องขอจากผู้ใช้งานกับแต่ละลายนิ้วมือของสัญญาณด้วยการคำนวณระยะทางยูคลิเดียน หลังจากนั้นจะคำนวณค่าน้ำหนักของแต่ละลายนิ้วมือสัญญาณ ดังสมการที่ 4

$$w^j = \frac{\frac{1}{STI^j}}{\sum_{j=1}^m \frac{1}{STI^j}} \quad (4)$$

โดย  $w^j$  คือค่าน้ำหนักของลายนิ้วมือสัญญาณที่  $j$ ,  $STI^j$  คือค่าดัชนีความโน้มเอียงสัญญาณของลายนิ้วมือของสัญญาณที่  $j$  และ  $m$  คือจำนวนลายนิ้วมือของสัญญาณทั้งหมด

การพิจารณาตำแหน่งของระบบ WinIPS นั้นจะเลือกลายนิ้วมือของสัญญาณที่มีค่าน้ำหนักมากที่สุด  $k$  ลำดับแรกมาหาตำแหน่งผลลัพธ์ดังสมการที่ 5

$$(x_d, y_d) = \frac{1}{\sum_{i=1}^k w^i} \sum_{i=1}^k (x_i, y_i) \cdot w^i \quad (5)$$

โดย  $(x_d, y_d)$  คือตำแหน่งผลลัพธ์,  $w^i$  คือค่าน้ำหนักของลายนิ้วมือสัญญาณที่  $i$  และ  $(x_i, y_i)$  คือตำแหน่งของลายนิ้วมือสัญญาณที่  $i$

จากผลการทดลองระบุว่าความคลาดเคลื่อนโดยเฉลี่ยของ WinIPS อยู่ที่ 1.7 เมตร นอกจากนี้ WinIPS จะสามารถลดความคลาดเคลื่อนในการระบุตำแหน่งจากปัญหาตำแหน่งที่ไม่ตรงกันได้เช่นเดียวกับ RADAR และประสิทธิภาพในการทำงานของ WinIPS จะลดลงเมื่อถูกใช้งานในพื้นที่ขนาดใหญ่เช่นเดียวกัน

#### 2.2.2 งานวิจัยเกี่ยวกับการจัดหมวดหมู่พื้นที่

งานวิจัยที่เกี่ยวกับการจัดหมวดหมู่พื้นที่ [26, 38-40] ถูกนำเสนอขึ้นมาเพื่อต้องการจะระบุให้ได้ว่าผู้ใช้งานอยู่ ณ บริเวณใด เช่น ภายในอาคาร, ภายนอกอาคาร [38, 39] หรือ

อาคารใด [26, 40] โดยการจัดหมวดหมู่พื้นที่นั้นมีจุดประสงค์หลักเพื่อลดปริภูมิการค้นหา (Search space) และการคำนวณของอัลกอริทึมระบุตำแหน่งภายในอาคาร

Shining He et al. [26] นำเสนออัลกอริทึมการจัดหมวดหมู่พื้นที่ที่ใช้เพียงสัญญาณวิทยุเท่านั้น ซึ่งอัลกอริทึมจะใช้เทคนิคลายนิ้วมือของสัญญาณประยุกต์กับวิธีการเชิงกำหนด อัลกอริทึมประกอบไปด้วยสองส่วนสำคัญดังนี้

ส่วนตรวจจับภายใน/ภายนอกบริเวณ (Inside/outside region detection)

ส่วนนี้มีหน้าที่ในการแยกแยะคำร้องของผู้ใช้งานว่าส่งมาจากบริเวณภายในหรือภายนอกบริเวณที่กำหนด ซึ่งในกรณีนี้คือภายในหรือภายนอกอาคาร วิธีการที่ถูกนำมาประยุกต์ใช้และได้ผลลัพธ์ที่ดีที่สุดคือการวิเคราะห์ส่วนประกอบหลัก (PCA: Principal Component Analysis) ซึ่งมีความแม่นยำ 95.69% ในการจำแนกบริเวณ ถ้าหากคำร้องของผู้ใช้งานถูกระบุว่าอยู่ภายในบริเวณที่กำหนดแล้ว จะส่งคำร้องนั้นไปยังส่วนต่อไป

ส่วนจัดหมวดหมู่พื้นที่ (Area classification module)

เมื่อคำร้องของผู้ใช้งานถูกระบุว่าอยู่ภายในบริเวณที่กำหนด จะถูกส่งมาที่ส่วนจัดหมวดหมู่พื้นที่ ส่วนนี้มีหน้าที่ในการระบุคำร้องของผู้ใช้งานว่า ณ บริเวณใดซึ่งในกรณีนี้คืออยู่ ณ อาคารใด วิธีการที่ถูกนำมาประยุกต์ใช้และได้ผลลัพธ์ที่ดีที่สุดคือ เอชวีเอ็ม (SVM: Support Vector Machine) ซึ่งมีความแม่นยำ 92.7% ในการจัดหมวดหมู่พื้นที่

จุฬาลงกรณ์มหาวิทยาลัย

กระบวนการทำงานของงานวิจัยนี้สามารถแบ่งได้เป็นสองช่วงเช่นเดียวกันกับงานวิจัยระบุตำแหน่งภายในอาคารที่ใช้เทคนิคลายนิ้วมือของสัญญาณ โดยข้อมูลการวัดค่าสัญญาณวิทยุที่เก็บในช่วงของการฝึกนั้น สามารถถูกนำมาใช้ประโยชน์ได้ทั้งสองส่วนของอัลกอริทึม ซึ่งข้อมูลเหล่านั้นจะถูกสร้างเป็นลายนิ้วมือของสัญญาณโดยใช้กระบวนการเดียวกันกับ RADAR และ WinIPS กล่าวคือแปลงให้อยู่ในปริภูมิเวกเตอร์

### บทที่ 3 ระบบระบุตำแหน่งภายในอาคารที่นำเสนอ

ระบบหรืออัลกอริทึมระบุตำแหน่งภายในอาคาร ณ ปัจจุบันพบกับปัญหาเกี่ยวกับประสิทธิภาพการทำงานลดลงเมื่อถูกนำไปใช้งานในพื้นที่ที่มีขนาดใหญ่ เนื่องจากความซับซ้อนในการประมวลผลนั้นขึ้นอยู่กับจำนวนลายนิ้วมือของสัญญาณในฐานข้อมูล นอกจากนี้บางอัลกอริทึมจะไม่สามารถเลี้ยงปัญหาตำแหน่งที่ไม่ตรงกันได้ เพราะผลลัพธ์ตำแหน่งขึ้นอยู่กับลายนิ้วมือของสัญญาณเพียงลายนิ้วมือเดียว ยิ่งกว่านั้นการเปลี่ยนแปลงของสภาพแวดล้อมก็ส่งผลต่อความแม่นยำในการระบุตำแหน่ง ดังนั้นงานวิจัยชิ้นนี้จึงได้นำเสนอระบบระบุตำแหน่งภายในอาคารที่ถูกออกแบบมาให้มีความสามารถในการปรับขนาดได้ (Scalability) ต่อขนาดของพื้นที่ที่ระบบครอบคลุม โดยใช้ประโยชน์จากแนวคิดการจัดหมวดหมู่พื้นที่ ระบบที่นำเสนอนี้ยังคำนึงถึงความแม่นยำในการระบุตำแหน่ง เวลาและแรงงานในการสำรวจเก็บข้อมูลสัญญาณวายฟาย โดยการแก้ปัญหาตำแหน่งที่ไม่ตรงกันด้วย นอกจากนี้ระบบดังกล่าวมีการปรับตัวให้เข้ากับสภาพแวดล้อมที่เปลี่ยนแปลงไปด้วย

#### 3.1) ภาพรวมของระบบ

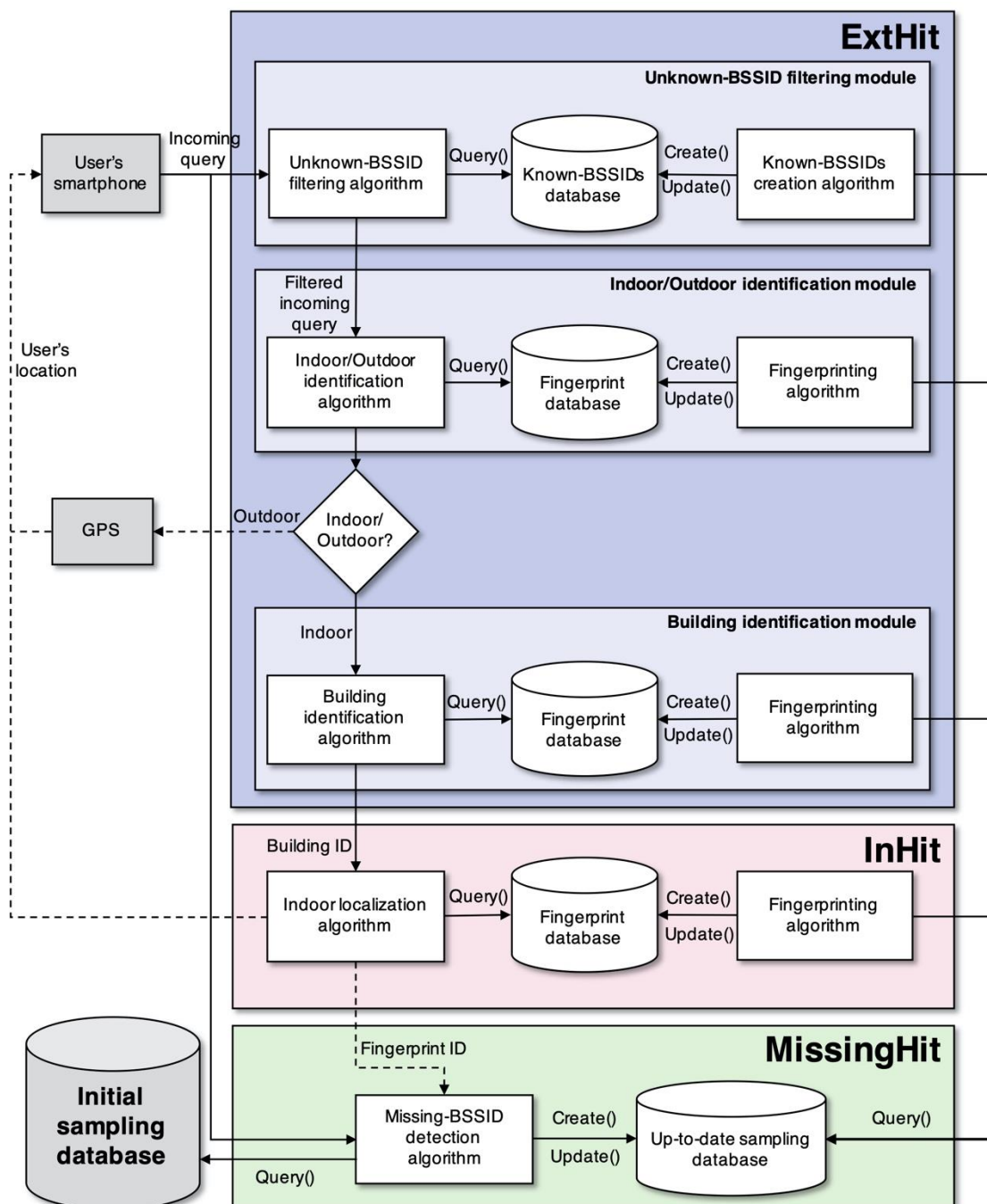
สถาปัตยกรรมของระบบแสดงได้ดังภาพที่ 5 โดยระบบประกอบไปด้วยกันทั้งหมดสามส่วน

- 1) อัลกอริทึมจัดหมวดหมู่พื้นที่ ชื่อว่า ExtHit มีหน้าที่ในการจำแนกพื้นที่ของผู้ใช้งานว่าอยู่ภายนอกอาคารหรือภายในอาคารใด ถ้าอัลกอริทึมสามารถระบุได้ว่าผู้ใช้งานอยู่ภายในอาคารใดแล้ว ข้อมูลระบุอาคาร (Building ID) นั้น ๆ จะถูกส่งไปยังส่วนต่อไป ไม่เช่นนั้นแล้ว บริการสำหรับการระบุตำแหน่งภายนอกอาคารจะถูกเรียกใช้งานแทน เช่น จีพีเอส
- 2) อัลกอริทึมระบุตำแหน่งภายในอาคาร โดยใช้เทคนิคลายนิ้วมือของสัญญาณ ชื่อว่า InHit จะใช้ข้อมูลระบุอาคารจากส่วนที่แล้วในการจำกัดขอบเขตการค้นหาลายนิ้วมือของสัญญาณในฐานข้อมูล หลังจากนั้นระบบจะส่งตำแหน่งของผู้ใช้งานกลับไปเป็นผลลัพธ์สำหรับคำร้องขอจากผู้ใช้งาน
- 3) อัลกอริทึมตรวจจับบีเอสเอสไอดีที่หายไป ชื่อว่า MissingHit มีหน้าที่สำหรับตรวจจับบีเอสเอสไอดีที่หายไปจากคำร้องขอจากผู้ใช้งานและปรับปรุงฐานข้อมูลสัญญาณวายฟาย

ExtHit ประกอบไปด้วยกันสามโมดูล ได้แก่ โมดูลกรองบีเอสเอสไอดีที่ไม่รู้จัก (Unknown-BSSID filtering module), โมดูลจำแนกในอาคาร/นอกอาคาร (Indoor/Outdoor identification module), โมดูลจำแนกอาคาร (Building identification module) โดยโมดูลกรองบีเอสเอสไอดีที่ไม่รู้จักจะกรองบีเอสเอสไอดีที่ระบบไม่เคยพบในการสำรวจสัญญาณวายฟายออกจากคำร้องขอจากผู้ใช้งาน ซึ่งเมื่อทำงานเสร็จเรียบร้อยแล้ว คำร้องขอจากผู้ใช้งานที่ถูกกรองแล้ว (Filtered incoming query) จะถูกส่งไปยังโมดูลต่อไป โมดูลจำแนกในอาคาร/นอกอาคารจะจำแนกคำร้องขอจากผู้ใช้งานที่ถูกกรองแล้วว่าถูกส่งมาจากในอาคารหรือนอกอาคาร ถ้าหากโมดูล



ได้ผลลัพธ์ว่าถูกส่งมาจากภายในอาคารแล้ว จะส่งไปให้โมดูลต่อไปทำงาน โมดูลจำแนกอาคารมีหน้าที่ระบุว่าผู้ใช้งานอยู่ในอาคารใด เมื่อประมวลผลเสร็จแล้ว ExtHit จะส่งข้อมูลระบุมอาคารไปยัง InHit เพื่อระบุตำแหน่งภายในอาคารอย่างละเอียดต่อไป



ภาพที่ 5 สถาปัตยกรรมของระบบที่นำเสนอ

### 3.2) โมดูลกรองบีเอสเอสไอดีที่ไม่รู้จัก

ในทางปฏิบัติ นั้น จะมีบีเอสเอสไอดีที่ระบบไม่เคยพบมากมายปรากฏขึ้นหลังจากที่เปิดใช้งานระบบ เนื่องจากการติดตั้งตัวกระจายสัญญาณวิทยุใหม่หรือเกิดจากการเปิดฮอตสปอต (Hotspot) ยิ่งไปกว่านั้น คำร้องขอจากผู้ใช้งานที่ถูกส่งมาจากพื้นที่ที่ระบบไม่ได้ครอบคลุมนั้น จะเต็มไปด้วยบีเอสเอสไอดีที่ระบบไม่รู้จัก ซึ่งสถานการณ์เหล่านี้สามารถลดทอนประสิทธิภาพโดยรวมของระบบได้

เพื่อที่จะแก้ปัญหาที่เกิดจากบีเอสเอสไอดีที่ไม่รู้จักดังกล่าว จึงได้ออกแบบโมดูลกรองบีเอสเอสไอดีที่ไม่รู้จักเพื่อคัดกรองบีเอสเอสไอดีที่ไม่รู้จักออกจากคำร้องขอจากผู้ใช้งาน โดยโมดูลดังกล่าวมีหลักการทำงานดังนี้ ขั้นตอนแรกอัลกอริทึมสร้างบีเอสเอสไอดีที่รู้จัก จะสร้างรายการของบีเอสเอสไอดีที่รู้จักจากข้อมูลลักษณะสัญญาณที่ถูกเก็บในช่วงของการฝึกและเก็บรายการนั้นไว้ในฐานข้อมูล ต่อมา เมื่อผู้ใช้งานส่งคำร้องขอมาที่ระบบ อัลกอริทึมกรองบีเอสเอสไอดีที่ไม่รู้จักจะคัดกรองบีเอสเอสไอดีที่ไม่รู้จักออกจากคำร้องขอดังกล่าว หลังจากนั้น อัลกอริทึมจะส่งคำร้องขอจากผู้ใช้งานที่ถูกกรองแล้วออกมาเป็นผลลัพธ์จากการทำงาน

อัลกอริทึมสร้างบีเอสเอสไอดีที่รู้จักและอัลกอริทึมกรองบีเอสเอสไอดีที่ไม่รู้จักถูกแสดงในอัลกอริทึมที่ 1 และ 2 ตามลำดับ โดยความซับซ้อนของอัลกอริทึมที่ 1 คือ  $O(sn)$  โดย  $s$  คือจำนวนลักษณะสัญญาณวิทยุที่เก็บในช่วงของการฝึก และ  $n$  คือจำนวนบีเอสเอสไอดีที่สูงที่สุดของลักษณะสัญญาณวิทยุที่เก็บในช่วงของการฝึก ความซับซ้อนของอัลกอริทึมที่ 2 คือ  $O(n)$  โดย  $n$  คือจำนวนบีเอสเอสไอดีของคำร้องขอจากผู้ใช้งาน

#### อัลกอริทึมที่ 1 อัลกอริทึมสร้างบีเอสเอสไอดีที่รู้จัก

---

##### **Algorithm 1** Creating the List of Known BSSIDs

---

**Input:** collection of Wi-Fi scanning results  $sc\_list$

**Output:** known BSSIDs set  $bssid\_set$

- 1: Initialize  $bssid\_set$
  - 2: **for each**  $sc$  **in**  $sc\_list$  **do**:
  - 3:     **for each**  $bssid$  **in**  $sc$  **do**:
  - 4:         Add  $bssid$  to  $bssid\_set$
  - 5:     **end for**
  - 6: **end for**
  - 7: return  $bssid\_set$
-

## อัลกอริทึมที่ 2 อัลกอริทึมกรองบีเอสเอสไอดีที่ไม่รู้จัก

---

### Algorithm 2 Filtering the Unknown BSSIDs

---

**Input:** Wi-Fi scanning result from query *sc*,  
known BSSIDs set from database *bssid\_set*

**Output:** filtered Wi-Fi scanning result *filtered\_sc*

```

1: Initialize filtered_sc
2: for each bssid in sc do:
3:   if bssid is in bssid_set then
4:     Add bssid to filtered_sc
5:   end if
6: end for
7: return filtered_sc

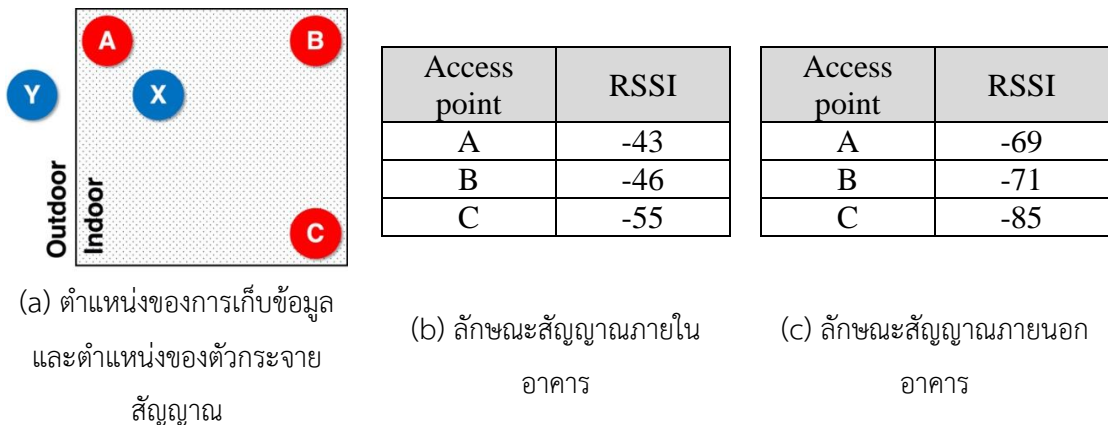
```

---

### 3.3) โมดูลจำแนกในอาคาร/นอกอาคาร

จากผลการทดลองเพื่อออกแบบอัลกอริทึมสำหรับจำแนกในอาคาร/นอกอาคาร จะสังเกตได้ว่าค่าความแรงสัญญาณที่ถูกเก็บจากตำแหน่งภายในอาคารนั้นมีค่าสูงกว่าค่าความแรงสัญญาณที่ถูกเก็บจากตำแหน่งภายนอกอาคารอย่างเห็นได้ชัด ภาพที่ 6 (a) แสดงตำแหน่งที่เก็บข้อมูลและตำแหน่งของตัวกระจายสัญญาณ โดยกรอบสี่เหลี่ยมคือพื้นที่ภายในอาคาร ตำแหน่ง A, B และ C คือตำแหน่งของตัวกระจายสัญญาณ ตำแหน่ง X และ Y คือตำแหน่งที่เก็บข้อมูล ภาพที่ 6 (b) และ ภาพที่ 6 (c) แสดงลักษณะสัญญาณวายฟายของตำแหน่ง X และ Y ตามลำดับ จากผลลัพธ์ดังกล่าวแสดงให้เห็นว่าลำดับของตัวกระจายสัญญาณของสองตำแหน่งนั้นเหมือนกัน แต่ค่าความแรงสัญญาณนั้นแตกต่างกันอย่างชัดเจน ดังนั้นอัลกอริทึมนี้จึงควรจะใช้ประโยชน์จากพฤติกรรมนี้

โมดูลจำแนกในอาคาร/นอกอาคารประกอบไปด้วยกันสองช่วงของการทำงาน ช่วงแรกคือช่วงของการฝึก จะเก็บข้อมูลลักษณะสัญญาณวายฟายในแต่ละตำแหน่งภายในอาคาร เพื่อที่จะสร้างลายนิ้วมือของสัญญาณของแต่ละอาคาร หลังจากนั้นคำนวณค่า *adaptive threshold score* เพื่อที่จะใช้เป็นข้อมูลขาเข้าสำหรับช่วงต่อไป ช่วงที่สองคือช่วงของการระบุตำแหน่ง จะจำแนกพื้นที่ของผู้ใช้งาน



(a) ตำแหน่งของการเก็บข้อมูล และตำแหน่งของตัวกระจายสัญญาณ

(b) ลักษณะสัญญาณภายในอาคาร

(c) ลักษณะสัญญาณภายนอกอาคาร

ภาพที่ 6 ผลลัพธ์การทดลองที่แสดงลักษณะสัญญาณวิทยุระหว่างภายในอาคารและภายนอกอาคาร

### 3.3.1 ช่วงของการฝึก

ในช่วงของการฝึกนั้น จะเก็บลักษณะสัญญาณวิทยุจากแต่ละตำแหน่งภายในอาคารและสร้างลายนิ้วมือของสัญญาณสำหรับแต่ละอาคาร โดยสำหรับแต่ละลักษณะสัญญาณ บีเอสเอสไอดีจะถูกเรียงลำดับตามความแรงของสัญญาณ จึงได้รูปแบบดังนี้

$$S_i = [s_i^1, s_i^2, s_i^3, \dots, s_i^n] \quad (6)$$

โดยที่  $S_i$  คือลักษณะสัญญาณที่  $i$ ,  $s_i^j$  คือบีเอสเอสไอดีของตัวกระจายสัญญาณที่  $j$  ใน  $S_i$ ,  $n$  คือจำนวนบีเอสเอสไอดีใน  $S_i$  และค่าความแรงสัญญาณของ  $s_i^j$  จะสูงกว่าหรือเท่ากับค่าความแรงสัญญาณของ  $s_i^{j+1}$

อัลกอริทึมสร้างลายนิ้วมือของสัญญาณนั้นจะใช้บีเอสเอสไอดีในการจำแนกตัวกระจายสัญญาณแต่ละตัวออกจากกัน อัลกอริทึมจะรวมบีเอสเอสไอดี  $N$  ตัวที่ให้ค่าความแรงสัญญาณสูงที่สุดจากแต่ละลักษณะสัญญาณที่ถูกเก็บภายในอาคารนั้น ๆ เข้าด้วยกันเพื่อสร้างเป็นลายนิ้วมือของสัญญาณของอาคารแต่ละอาคาร อัลกอริทึมสร้างลายนิ้วมือของสัญญาณสำหรับ ExtHit ถูกแสดงในอัลกอริทึมที่ 3 โดย top- $N$  BSSIDs คือบีเอสเอสไอดี  $N$  ตัวแรกที่ใช้ค่าความแรงสัญญาณสูงที่สุด ซึ่งจะถูกนำมาใช้สร้างลายนิ้วมือของสัญญาณ ความซับซ้อนของอัลกอริทึมคือ  $O(Ns)$  เมื่อใช้ top- $N$  BSSIDs เป็นข้อมูลขาเข้าของอัลกอริทึมและ  $s$  คือจำนวนลักษณะสัญญาณที่ถูกเลือกมาสร้างลายนิ้วมือของสัญญาณ

### อัลกอริทึมที่ 3 อัลกอริทึมสร้างลายนิ้วมือของสัญญาณสำหรับ ExtHit

---

#### Algorithm 3 Fingerprinting algorithm for ExtHit

---

**Input:** top-N BSSIDs  $N$ ,  
collection of Wi-Fi scanning results  $sc\_list$ ,  
building name  $building\_name$

**Output:** fingerprint  $fingerprint$

```

1: Initialize  $bssid\_set$ 
2: for each  $scanning\_result$  in  $sc\_list$  do:
3:   Initialize  $count_N$  to 0
4:   for each  $bssid$  in  $scanning\_result$  do:
5:     if  $count_N$  is less than  $N$  then
6:       Add 1 to  $count_N$ 
7:       if  $bssid$  not in  $bssid\_set$ :
8:         Add  $bssid$  to  $bssid\_set$ 
9:       end if
10:    end if
11:  end for
12: end for
13:  $fingerprint \leftarrow \{building\_name, bssid\_set\}$ 
14: return  $fingerprint$ 

```

---

เมื่อสร้างลายนิ้วมือของสัญญาณแล้ว จะสามารถจัดรูปแบบได้ดังนี้

$$FP = \{F_1, F_2, F_3, \dots, F_b\} \quad (7)$$

เมื่อ  $FP$  คือชุด (Set) ของลายนิ้วมือของสัญญาณของแต่ละอาคาร,  $b$  คือจำนวนอาคารภายในระบบครอบคลุมและ  $F_i$  คือลายนิ้วมือของสัญญาณของอาคารที่  $i$  ซึ่งสามารถถูกกำหนดได้ดังนี้

$$F_i = \{f_i^1, f_i^2, f_i^3, \dots, f_i^m\} \quad (8)$$

เมื่อ  $f_i^j$  คือบีเอสเอสไอทีของตัวกระจายสัญญาณที่  $j$  ใน  $F_i$ ,  $m$  คือจำนวนบีเอสเอสไอทีใน  $F_i$

หลังการสร้างลายนิ้วมือของสัญญาณ โมดูลจะคำนวณค่า *adaptive threshold score* โดยจะทำการแปลงลักษณะสัญญาณโดยใช้ 2 ค่าคงที่ ซึ่งประกอบไปด้วยค่า top-N BSSIDs ที่ถูกใช้ในการสร้างลายนิ้วมือของสัญญาณและค่า *pre-defined RSSI* ซึ่งบีเอสเอสไอทีใดที่มีค่าความแรงสัญญาณน้อยกว่าค่า *pre-defined RSSI* จะถูกนำออกในกระบวนการแปลงนี้ ดังนั้นผลลัพธ์จากการแปลงนี้จะได้

$$T_i = [s_i^1, s_i^2, s_i^3, \dots, s_i^l] \quad (9)$$

เมื่อ  $T_i$  คือผลลัพธ์การแปลงของ  $S_i$ ,  $s_i^k$  จะมีค่าความแรงสัญญาณมากกว่าหรือเท่ากับ *pre-defined RSSI* และ  $l$  คือจำนวนบีเอสเอสไอดีใน  $T_i$  หลังจากนั้นโมดูลจะคำนวณคะแนน  $H_i$  ของ  $T_i$  โดยใช้สมการที่ 10

$$H_i = \max_{0 \leq j \leq b} \sum_{k=1}^l \begin{cases} 1, & s_i^k \in F_j \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

อัลกอริทึมสำหรับการคำนวณคะแนนดังกล่าวถูกแสดงไว้ที่อัลกอริทึมที่ 4 ความซับซ้อนของอัลกอริทึมคือ  $O(bN)$  เมื่อใช้ top- $N$  BSSIDs เป็นข้อมูลขาเข้าของอัลกอริทึม และ  $b$  คือจำนวนลายนิ้วมือของสัญญาณของอาคาร ในขั้นตอนสุดท้ายโมดูลจะเลือกคะแนนที่น้อยที่สุดเป็นค่า *adaptive threshold score* เพื่อไว้สำหรับใช้ใน ช่วงต่อไป

อัลกอริทึมที่ 4 อัลกอริทึมคำนวณคะแนนสำหรับโมดูลจำแนกในอาคาร/นอกอาคาร

---

**Algorithm 4** Indoor/outdoor identification algorithm

---

**Input:** top- $N$  BSSIDs  $N$ ,  
 minimum RSSI  $min\_rssi$ ,  
 Wi-Fi scanning results from query  $sc$ ,  
 list of fingerprints from database  $fp\_list$

**Output:** fingerprint score  $score$

```

1: Initialize  $count_N$  to 0
2: Initialize  $score$  to 0
3: for each  $bssid$  in  $sc$  do:
4:   if  $count_N$  is less than  $N$  then
5:     Add 1 to  $count_N$ 
6:     if  $bssid.rssi$  is more than or equal to  $min\_rssi$ :
7:       for each  $fp$  in  $fp\_list$  do:
8:         if  $bssid$  in  $fp$ :
9:            $score \leftarrow score + 1$ 
10:        end if
11:       end for
12:     end if
13:   end if
14: end for
15: return  $score$ 

```

---

### 3.3.2 ช่วงของการระบุตำแหน่ง

ในช่วงของการระบุตำแหน่งนี้ โมดูลจะทำการจำแนกตำแหน่งของผู้ใช้งานว่าอยู่ภายในหรือภายนอกอาคาร โดยขั้นตอนแรกโมดูลจะจัดรูปแบบคำร้องขอจากผู้ใช้งานตั้งสมการที่ (6) ต่อมาในขั้นตอนต่อไปจึงทำการแปลงคำร้องขอจากผู้ใช้งาน โดยการใช้วิธีการที่ได้ระบุไว้ในสมการที่ (9) หลังจากนั้นจึงคำนวณคะแนนของคำร้องขอจากผู้ใช้งานดังกล่าวตามสมการที่ (10) และขั้นตอนสุดท้าย ถ้าหากคะแนนที่ได้จากการคำนวณนั้นมากกว่าหรือเท่ากับค่า *adaptive threshold score* แล้วนั้นหมายความว่าลักษณะสัญญาณของพื้นที่ที่ผู้ใช้งานอยู่นั้น มีความคล้ายคลึงกับลักษณะสัญญาณภายในอาคาร ดังนั้นอัลกอริทึมจึงจะให้ผลลัพธ์เป็น “ภายในอาคาร” ไม่เช่นนั้นแล้ว อัลกอริทึมจะให้ผลลัพธ์เป็น “ภายนอกอาคาร” เนื่องจากไม่มีปีเอสเอสไอดีที่ตรงกันระหว่างคำร้องขอจากผู้ใช้งานและลายนิ้วมือของสัญญาณของอาคารมากเพียงพอ

### 3.4) โมดูลจำแนกอาคาร

จากจุดประสงค์ในการลดจำนวนข้อมูลที่ต้องค้นหาภายในฐานข้อมูล เพื่อให้ใช้เวลาในการประมวลผลน้อยลง จึงได้ออกแบบอัลกอริทึมที่มีความซับซ้อนต่ำในการจำแนกว่าผู้ใช้งานอยู่ในอาคารใด ซึ่งการใช้อาคารในการจำแนกพื้นที่นั้นนับว่าเป็นการแบ่งขนาดพื้นที่ที่เหมาะสม เนื่องจากอัลกอริทึมที่เกี่ยวข้องกับระบุตำแหน่งภายในอาคารต่าง ๆ มักจะอ้างอิงและพิจารณาตำแหน่งจากชั้นหรืออาคาร

โมดูลนี้ออกแบบจากสมมติฐานที่ว่าอาคารแต่ละอาคารจะมีลายนิ้วมือของสัญญาณวายฟายที่เป็นเอกลักษณ์ของตัวเอง ซึ่งลายนิ้วมือของสัญญาณดังกล่าวจะประกอบไปด้วยปีเอสเอสไอดีที่สามารถให้ค่าความแรงสัญญาณที่สูง เมื่อเก็บข้อมูลจากภายในอาคารนั้น ๆ โดยโมดูลจำแนกอาคารจะประกอบไปด้วยกันสองช่วง ช่วงแรกคือช่วงของการฝึก โดยลักษณะสัญญาณจากตำแหน่งต่าง ๆ ภายในอาคารจะถูกเก็บในขั้นตอนนี้ หลังจากนั้นจะถูกนำไปสร้างเป็นลายนิ้วมือของสัญญาณวายฟายของอาคารแต่ละอาคาร ช่วงที่สองคือช่วงของการระบุตำแหน่ง โดยจะจำแนกอาคารที่ผู้ใช้งานอยู่ขณะที่ส่งคำร้องขอมาที่ระบบในขั้นตอนนี้

#### 3.4.1) ช่วงของการฝึก

กระบวนการของอัลกอริทึมสร้างลายนิ้วมือของสัญญาณสำหรับโมดูลนี้นั้น เหมือนกับอัลกอริทึมสร้างลายนิ้วมือของสัญญาณสำหรับโมดูลจำแนกในอาคาร/นอกอาคาร โดยอัลกอริทึมดังกล่าวได้ถูกแสดงไว้ในอัลกอริทึมที่ 3

แต่อย่างไรก็ตามค่าคงที่  $N$  สำหรับ top- $N$  BSSIDs สำหรับโมดูลจำแนกอาคารนั้นสามารถใช้ค่าที่แตกต่างกับโมดูลจำแนกภายใน/ภายนอกอาคาร เนื่องจากอัลกอริทึมสร้างลายนิ้วมือของสัญญาณทั้งสองนั้นทำงานแยกจากกันอย่างอิสระ ดังนั้นจึงสามารถเลือกค่าคงที่  $N$  ที่ให้ผลลัพธ์ที่ดีที่สุดในการทำงานสำหรับแต่ละอัลกอริทึมได้

### 3.4.2) ช่วงของการระบุตำแหน่ง

อัลกอริทึมสำหรับการจำแนกอาคารจะเปรียบเทียบคำร้องขอจากผู้ใช้งานกับแต่ละลายนิ้วมือของสัญญาณของแต่ละอาคารภายในฐานข้อมูล โดยขั้นตอนแรกอัลกอริทึมจะจัดรูปแบบคำร้องขอจากผู้ใช้งานดังสมการที่ (6) และแปลงคำร้องขอดังกล่าวด้วยการใช้ค่าคงที่ top- $N$  BSSIDs ที่ถูกใช้ในช่องของการฝึก ดังนั้นผลลัพธ์จากการแปลงคือ

$$Q = [q^1, q^2, q^3, \dots, q^p] \quad (11)$$

โดย  $Q$  คือผลลัพธ์จากการแปลงคำร้องขอจากผู้ใช้งานและ  $p$  คือจำนวนปีเอสเอสไอทีใน  $Q$  หลังจากนั้นอัลกอริทึมจะคำนวณค่า  $\alpha$  ซึ่งเป็นค่าที่แสดงถึงความคล้ายคลึงกันระหว่างคำร้องขอจากผู้ใช้งานกับแต่ละลายนิ้วมือของสัญญาณของอาคาร โดยค่า  $\alpha$  ของแต่ละลายนิ้วมือของสัญญาณของอาคาร  $F_i$  จะสามารถคำนวณได้ดังนี้

$$\alpha = \sum_{j=1}^p \begin{cases} 1, & q^j \in F_i \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

เมื่อกำหนดเสร็จเรียบร้อยแล้ว อัลกอริทึมจะให้คำตอบเป็นอาคารที่สามารถให้ค่า  $\alpha$  ได้สูงที่สุด โดยอัลกอริทึมจำแนกอาคารถูกแสดงไว้ในอัลกอริทึมที่ 5 ความซับซ้อนของอัลกอริทึมดังกล่าวคือ  $O(bN)$  เมื่อใช้ top- $N$  BSSIDs เป็นข้อมูลขาเข้าของอัลกอริทึมและ  $b$  คือจำนวนลายนิ้วมือของสัญญาณของอาคารทั้งหมดภายในฐานข้อมูล



### อัลกอริทึมที่ 5 อัลกอริทึมจำแนกอาคาร

---

#### Algorithm 5 Building identification algorithm

---

**Input:** top-N BSSIDs  $N$ ,

Wi-Fi scanning result from query  $sc$ ,  
list of fingerprints from database  $fp\_list$

**Output:** building ID  $building\_id$

```

1: Initialize  $count_N$  to 0
2: for each  $fp$  in  $fp\_list$  do:
3:   Initialize  $fp.\alpha$  to 0
4: end for
5: for each  $bssid$  in  $sc$  do:
6:   if  $count_N$  is less than  $N$  then
7:     Add 1 to  $count_N$ 
8:     for each  $fp$  in  $fp\_list$  do:
9:       if  $bssid$  in  $fp$ :
10:        Add 1 to  $fp.\alpha$ 
11:       end if
12:     end for
13:   end if
14: end for
15: Initialize  $\alpha$  to 0
16: for each  $fp$  in  $fp\_list$  do:
17:   if  $fp.\alpha$  is more than  $\alpha$  then
18:      $\alpha \leftarrow fp.\alpha$ 
19:      $building\_id \leftarrow fp.building\_id$ 
20:   end if
21: end for
22: return  $building\_id$ 

```

CHULALONGKORN UNIVERSITY

### 3.5) อัลกอริทึม InHit

InHit เป็นอัลกอริทึมสำหรับระบุตำแหน่งภายในอาคารโดยใช้เทคนิคลายนิ้วมือของสัญญาณ ซึ่งสามารถระบุได้ว่าผู้ใช้งานอยู่ชั้นใดของอาคาร และอยู่ตำแหน่งใดของชั้นนั้น ๆ อัลกอริทึมนี้ ประกอบไปด้วยกันสองช่วงการทำงาน โดยช่วงแรกคือช่วงของการฝึก ซึ่งจะเก็บข้อมูลลักษณะสัญญาณวายเป็นจากแต่พื้นที่ที่สนใจและนำข้อมูลเหล่านั้นมาสร้างลายนิ้วมือของสัญญาณเพื่อเก็บในฐานข้อมูลต่อไป ในช่วงที่สองของการทำงาน จะเป็นช่วงของการระบุตำแหน่งผู้ใช้งาน โดยจะระบุทั้งชั้นและตำแหน่งปัจจุบันของผู้ใช้งานอย่างละเอียด

### 3.5.1) ช่วงของการฝึก

ในช่วงของการฝึกสำหรับอัลกอริทึม InHit นั้นจะแตกต่างจากอัลกอริทึม ExtHit โดยอัลกอริทึม InHit นั้นจะสร้างลายนิ้วมือของสัญญาณวายพายสำหรับทุก ๆ ตำแหน่งที่สนใจ โดยแต่ละลักษณะสัญญาณของแต่ละตำแหน่ง จะถูกจัดรูปแบบดังสมการที่ (6) หลังจากนั้นอัลกอริทึมจะเลือกปีเอสเอสไอดี  $N$  ลำดับแรกที่สามารถให้ค่าความแรงสัญญาณสูงที่สุดจากลักษณะสัญญาณนั้น ๆ เพื่อนำมาสร้างลายนิ้วมือของสัญญาณของตำแหน่งนั้น ๆ ดังนั้นเมื่อสิ้นสุดขั้นตอนนี้ จะได้ชุดของลายนิ้วมือของสัญญาณวายพาย  $GP$  ซึ่งสามารถเขียนตามข้อกำหนดได้ดังนี้

$$GP = \{G_1, G_2, G_3, \dots, G_r\} \quad (13)$$

เมื่อ  $r$  คือจำนวนลายนิ้วมือของสัญญาณทั้งหมดภายในฐานข้อมูลของอัลกอริทึมนี้,  $G_i$  คือลายนิ้วมือของสัญญาณที่  $i$  ซึ่งสามารถเขียนตามข้อกำหนดได้ดังนี้

$$G_i = [s_i^1, s_i^2, s_i^3, \dots, s_i^t] \quad (14)$$

เมื่อ  $t$  คือจำนวนของปีเอสเอสไอดีใน  $G_i$  โดยอัลกอริทึมสำหรับการสร้างลายนิ้วมือของสัญญาณนี้ถูกแสดงไว้ที่อัลกอริทึมที่ 6 ความซับซ้อนของอัลกอริทึมคือ  $O(N)$  เมื่อใช้ top- $N$  BSSIDs เป็นข้อมูลขาเข้า ซึ่งค่าคงที่  $N$  สำหรับ top- $N$  BSSIDs ของอัลกอริทึม InHit นั้นสามารถแตกต่างกับของอัลกอริทึม ExtHit ได้ เนื่องจากแต่ละอัลกอริทึมนั้นทำงานเป็นอิสระจากกัน ดังนั้นจึงสามารถเลือกค่า  $N$  ที่ดีที่สุดสำหรับแต่ละอัลกอริทึมได้

### อัลกอริทึมที่ 6 อัลกอริทึมสร้างลายนิ้วมือของสัญญาณสำหรับ InHit

---

#### Algorithm 6 Fingerprinting algorithm for InHit

---

**Input:** top-N BSSIDs  $N$ ,  
 collection of Wi-Fi scanning results  $sc\_list$ ,  
 floor level  $floor\_level$ ,  
 position  $(x,y)$

**Output:** fingerprint  $fingerprint$

```

1: Initialize  $count_N$  to 0
2: Initialize  $bssid\_list$ 
3: for each  $bssid$  in  $sc\_list$  do:
4:   if  $count_N$  is less than  $N$  then
5:     Add 1 to  $count_N$ 
6:     Add  $bssid$  to  $bssid\_list$ 
7:   end if
8: end for
9:  $fingerprint \leftarrow \{floor\_level, (x, y), bssid\_set\}$ 
10: return  $fingerprint$ 

```

---

#### 3.5.2) ช่วงของการระบุตำแหน่ง

เมื่อมีคำร้องขอจากผู้ใช้งานเข้ามาที่โมดูล คำร้องขอเหล่านั้นจะถูกแปลงตั้งสมการที่ (11) ด้วยการใช้ค่าคงที่เดียวกับช่วงของการฝึกของโมดูลนี้ หลังจากนั้นอัลกอริทึมจะคำนวณค่า  $\beta$  ซึ่งเป็นค่าที่แสดงถึงความคล้ายคลึงกันระหว่างผลลัพธ์การแปลงของคำร้องขอกับแต่ละลายนิ้วมือของสัญญาณในฐานข้อมูล เมื่อจบขั้นตอนการคำนวณแล้ว จะพิจารณาดำแหน่งของผู้ใช้งานจากการเลือกลายนิ้วมือของสัญญาณที่มีค่า  $\beta$  สูงที่สุด  $k$  ลำดับแรกมาหาดำแหน่งผลลัพธ์ตั้งสมการที่ 5 โดยจะใช้ค่า  $\beta$  เป็นค่าสำหรับใช้ถ่วงน้ำหนักในการคำนวณ

โดยระบบคำนวณค่า  $\beta$  ของคำร้องขอจากผู้ใช้งาน  $Q$  และลายนิ้วมือของสัญญาณ  $G_i$  ดังสมการนี้

$$\beta = \sum_{j=1}^p \sum_{k=1}^t \begin{cases} 1, & q^j = s_i^k \text{ and } |RSSI(q^j) - RSSI(s_i^k)| \leq \gamma \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

เมื่อ  $RSSI(s)$  คือค่าความแรงสัญญาณของ  $s$  และ  $\gamma$  คือค่าคงที่ที่ถูุกกำหนดไว้ล่วงหน้าสำหรับอัลกอริทึม อัลกอริทึมการระบุตำแหน่งได้ถูกแสดงไว้ที่อัลกอริทึมที่ 7 ความซับซ้อนของอัลกอริทึมคือ  $O(rN^2)$  เมื่อใช้ top-N BSSIDs เป็นข้อมูลขาเข้าและ  $r$  คือจำนวนลายนิ้วมือของสัญญาณในฐานข้อมูล

### อัลกอริทึมที่ 7 อัลกอริทึมระบุตำแหน่งสำหรับ InHit

---

#### Algorithm 7 Localization algorithm for InHit

---

**Input:** Wi-Fi scanning result  $sc$ ,  
pre-defined value  $\gamma$ ,  
fingerprints from the database  $fp\_list$ ,  
pre-defined value  $k$

**Output:** floor level  $floor\_level$ ,  
position  $position$

```

1:  for each  $fp$  in  $fp\_list$  do:
2:    Initialize  $fp.\beta$  to 0
3:    for each  $bssid_{sc}$  in  $sc$  do:
4:      for each  $bssid_{fp}$  in  $fp$  do:
5:        if  $bssid_{sc}$  equals to  $bssid_{fp}$ :
6:          if  $|RSSI(bssid_{sc}) - RSSI(bssid_{fp})| \leq \gamma$ 
7:            Add 1 to  $fp.\beta$ 
8:          end if
9:        end if
10:       end for
11:      end for
12:    end for
13:
14:  Selects  $k$  fingerprints from  $fp\_list$  which provided the highest  $\beta$  value as  $k\_list$ 
15:  Initialize  $sum\_position$ ,  $sum\_floor$ ,  $sum\_weight$  to 0
16:  for each  $fp$  in  $k\_list$  do:
17:    Add  $fp.\beta * fp.position$  to  $sum\_position$ 
18:    Add  $fp.\beta * fp.floor$  to  $sum\_floor$ 
19:    Add  $fp.\beta$  to  $sum\_weight$ 
20:  end for
21:  Initialize  $floor\_level$  to  $sum\_floor / sum\_weight$ 
22:  Initialize  $position$  to  $sum\_position / sum\_weight$ 
23:  return  $floor\_level$ ,  $position$ 

```

---

### 3.6) อัลกอริทึม MissingHit

ในสถานการณ์การใช้งานจริงนั้น บีเอสเอสไอตีจำนวนมากสามารถหายไป เนื่องจากตัวกระจายสัญญาณอาจถูกเปลี่ยน, ถอนการติดตั้ง หรือเกิดจากการที่สำนักงานปิดชั่วคราว โดยสถานการณ์เหล่านี้ทำให้ประสิทธิภาพโดยรวมของระบบนั้นลดลงได้ เพื่อที่จะแก้ปัญหาเหล่านี้ งานวิจัยชิ้นนี้จึงได้นำเสนอโมดูลสำหรับตรวจจับบีเอสเอสไอตีที่หายไป โดยโมดูลดังกล่าวชื่อว่า MissingHit โมดูลนี้จะตรวจจับบีเอสเอสไอตีที่หายไปจากคำร้องขอจากผู้ใช้งาน และลบบีเอสเอสไอตีเหล่านั้นออกจากฐานข้อมูลลักษณะสัญญาณเพื่อบำรุงรักษาประสิทธิภาพโดยรวมของระบบ

การทำงานของ MissingHit จะเริ่มจากการที่ผู้ใช้งานส่งคำร้องขอมายังระบบเพื่อระบุตำแหน่ง ระบบจะสร้างสำเนาของคำร้องขอเหล่านั้นแล้วส่งมายังโมดูลนี้ หลังจากนั้นโมดูลนี้จะทำการจัดรูปแบบแต่ละคำร้องขอ  $R_i$  ด้วยสมการที่ (6) หลังจากนั้นเมื่อระบบทำการระบุตำแหน่งเสร็จเรียบร้อยแล้ว ระบบจะส่งลายนิ้วมือของสัญญาณที่มีความคล้ายคลึงที่สุดกล่าวคือลายนิ้วมือที่ให้ค่า  $\beta$  สูงที่สุดด้วยการใช้อัลกอริทึม InHit มายังโมดูลนี้เพื่อที่จะทำการตรวจจับปีเอสเอสไอดีที่หายไปต่อไป

สำหรับกระบวนการตรวจจับนั้น ในกรณีของงานวิจัยชิ้นนี้จะทำกระบวนการนี้วันละหนึ่งครั้ง เนื่องจากได้เปิดใช้งานระบบในงานนิทรรศการ จึงทำให้สภาพแวดล้อมสามารถเปลี่ยนได้ทุกวัน ดังนั้นถ้าหากระบบนี้ได้เปิดใช้งานในสถานการณ์อื่น ๆ หรือเงื่อนไขอื่น ๆ แล้ว คาบเวลาของกระบวนการนี้ก็ สามารถปรับเปลี่ยนได้ แต่อย่างไรก็ตาม โดยในทางปฏิบัตินั้น การใช้คำร้องขอจากผู้ใช้งานเพียงเล็กน้อยในการตรวจจับปีเอสเอสไอดีนั้นอาจเกิดความคลาดเคลื่อนได้มาก เนื่องจากปีเอสเอสไอดี บางตัวอาจจะหายไปเพียงชั่วคราวเท่านั้น ดังนั้นคาบของการทำงานจึงควรจะนานเพียงพอต่อการสะสมคำร้องขอจากผู้ใช้งานในระดับที่สมเหตุสมผล

เมื่อถึงกำหนดคาบเวลาในการปรับปรุงฐานข้อมูล อัลกอริทึมตรวจจับปีเอสเอสไอดีที่หายไป จะเริ่มทำงาน โดยสำหรับแต่ละคู่ของคำร้องขอจากผู้ใช้งานและลายนิ้วมือของสัญญาณนั้น อัลกอริทึม จะทำการดึงข้อมูลลักษณะสัญญาณที่ตรงกับลายนิ้วมือของสัญญาณนั้น ๆ ออกมาและจัดรูปแบบ ลักษณะสัญญาณ  $P_i$  โดยการใช้สมการที่ (6) หลังจากนั้นอัลกอริทึมจะสร้าง *possible missing-BSSIDs set* ของ  $R_i$  โดยเขียนชุดดังกล่าวเป็นตัวแปรชื่อ  $MS_i$  ซึ่งสามารถกำหนดได้ด้วยสมการดังนี้

$$MS_i = \{x | x \in P_i, x \notin R_i\} \quad (16)$$

และอัลกอริทึมจะสร้าง *found-BSSIDs set* ของ  $R_i$  โดยเขียนชุดดังกล่าวเป็นตัวแปรชื่อ  $FS_i$  โดยสามารถกำหนดด้วยสมการดังนี้

$$FS_i = \{x | x \in R_i\} \quad (17)$$

ในขั้นตอนต่อไป อัลกอริทึมจะสร้าง *actual missing-BSSIDs set* โดยกำหนดชุดดังกล่าว เป็น  $AMS$  ด้วยการใช้สมการดังนี้

$$AMS = \bigcup_{i=1}^u MS_i - \bigcup_{i=1}^u FS_i \quad (18)$$

เมื่อ  $u$  คือจำนวนคำร้องขอจากผู้ใช้งานที่ถูกนำมาใช้ในการปรับปรุงฐานข้อมูล ในขั้นตอนสุดท้าย โมดูลจะทำการปรับปรุงฐานข้อมูลด้วยการทำสำเนาของฐานข้อมูลลักษณะสัญญาณออกมา และลบปีเอสเอสไอดีที่ปรากฏในชุด  $AMS$  ดังนั้นเมื่อโมดูลอื่นต้องการปรับปรุงฐานข้อมูลลายนิ้วมือของ

สัญญาณของโมดูลนั้น ๆ แล้ว ฐานข้อมูลดังกล่าวจะไม่มีบีเอสเอสไอดีที่หายไปที่ถูกตรวจจับโดยโมดูลนี้ อัลกอริทึมสำหรับการสร้าง *AMS* นั้นถูกแสดงไว้ที่อัลกอริทึมที่ 8 โดยความซับซ้อนของอัลกอริทึมคือ  $O(un)$  เมื่อ  $n$  คือจำนวนบีเอสเอสไอดีในคำร้องขอจากผู้ใช้งานและ  $u$  คือจำนวนคำร้องขอจากผู้ใช้งานที่ถูกนำมาปรับปรุงฐานข้อมูล

#### อัลกอริทึมที่ 8 อัลกอริทึมสร้าง Actual missing-BSSIDs set

---

**Algorithm 8** Creating the set of actual missing-BSSIDs

---

**Input:** list of missing-BSSIDs set *MS*,

list of found-BSSIDs set *FS*,

**Output:** set of actual missing-BSSIDs *AMS*

- 1: Initialize *AMS* to { }
  - 2: Initialize *U\_MS* to { }
  - 3: **for each** *ms\_i* **in** *MS* **do:**
  - 4:     *U\_MS*  $\leftarrow$  *U\_MS* **union** *ms\_i*
  - 5: **end for**
  - 6:
  - 7: Initialize *U\_FS* to { }
  - 8: **for each** *fs\_i* **in** *FS* **do:**
  - 9:     *U\_FS*  $\leftarrow$  *U\_FS* **union** *fs\_i*
  - 10: **end for**
  - 11:
  - 12: *AMS*  $\leftarrow$  *U\_MS* - *U\_FS*
  - 13: return *AMS*
-

## บทที่ 4 การวัดประสิทธิภาพการทำงานของระบบ

### 4.1) การตั้งค่าการทดลอง

#### 4.1.1) การตั้งค่าการทดลองสำหรับอัลกอริทึม ExtHit

อัลกอริทึม ExtHit ถูกวัดผลความแม่นยำในงานนิทรรศการขนาดใหญ่ชื่อว่า “Chula Expo 2017” โดยข้อมูลสำหรับการทดลองถูกเก็บจากอาคารทั้งหมด 37 อาคาร ภายในพื้นที่ของจุฬาลงกรณ์มหาวิทยาลัย แผนที่ที่แสดงพื้นที่ที่งานนิทรรศการดังกล่าว ครอบคลุมได้แสดงไว้ในภาพที่ 7 สำหรับชุดข้อมูลในการทดลองนี้ มีทั้งหมด 5 ชุดข้อมูล โดยชุดข้อมูลที่ 1, 2 และ 3 นั้นถูกเก็บก่อนเริ่มงานนิทรรศการ ชุดข้อมูลที่ 4 นั้นถูกเก็บระหว่างงานนิทรรศการ และชุดข้อมูลที่ 5 นั้นเป็นข้อมูลจากคำร้องขอจากผู้ใช้งานจริงระหว่างวันงานนิทรรศการ

สำหรับการเก็บข้อมูลนั้น จะเก็บโดยการสุมตัวอย่างตลอดทางเดินภายในอาคารอย่างต่อเนื่อง และสำหรับการเก็บข้อมูลภายนอกอาคารนั้นจะเก็บตลอดทางเดินภายนอก รอบ ๆ อาคารในระยะ 2 – 3 เมตรจากกำแพงของอาคาร โดยรายละเอียดสำหรับแต่ละชุดข้อมูลนั้นถูกแสดงไว้ในตารางที่ 1 ซึ่งการเก็บข้อมูลดังกล่าวนั้นจะใช้โทรศัพท์มือถือรุ่น Samsung Galaxy S5 ทั้งหมด 2 เครื่องและ LG Nexus 5X 2 ทั้งหมด เครื่อง

โดยชุดข้อมูลที่ 1 ที่มีการเก็บข้อมูลจากทั้งภายนอกอาคารและภายในอาคารนั้น จะเก็บจากอาคารวิศวกรรมร้อยปีและอาคารเจริญวิศวกรรม เนื่องจากทั้งสองอาคารนี้อยู่ใกล้กัน เป็นอาคารที่มีลักษณะเป็นโถงเปิดขนาดใหญ่ และไม่มีกำแพงกั้นระหว่างภายในอาคาร และภายนอกอาคาร ดังนั้นจึงเป็นสภาพแวดล้อมที่มีความท้าทายอย่างสูงในการวัดผลความแม่นยำในการจำแนกคำร้องว่ามาจากภายในหรือภายนอกอาคาร ภาพถ่ายอาคารวิศวกรรมร้อยปีได้แสดงไว้ในภาพที่ 8



ภาพที่ 7 พื้นที่งานนิทรรศการ

ตารางที่ 1 จำนวนข้อมูลลักษณะสัญญาณวายพายและรายละเอียดของแต่ละชุดข้อมูลสำหรับทดสอบ

ExtHit

ชุดข้อมูล	วันที่เก็บข้อมูล	สภาพแวดล้อม		จำนวนลักษณะสัญญาณ	จำนวนอาคาร
		ในอาคาร	นอกอาคาร		
1	2 สัปดาห์ก่อนงานนิทรรศการ	✓	✓	1010	2
2	2 สัปดาห์ก่อนงานนิทรรศการ	✓	✗	2,239	37
3	2 สัปดาห์ก่อนงานนิทรรศการ	✓	✗	2,260	37
4	ภายในวันนิทรรศการ	✓	✗	1,142	34
5	ภายในวันนิทรรศการ	ไม่ทราบ	ไม่ทราบ	609,277	ไม่ทราบ





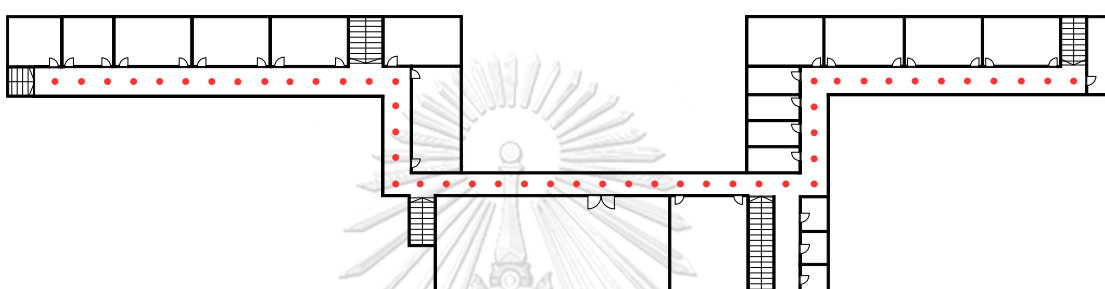
ภาพที่ 8 ภาพถ่ายอาคารวิศวกรรมร้อยปี

#### 4.1.2) การตั้งค่าการทดลองสำหรับอัลกอริทึม InHit

อัลกอริทึม InHit ถูกทดสอบในอาคารวิศวกรรมศาสตร์ 3 เนื่องจากเป็นอาคารเรียนที่มีขนาดใหญ่ ทางเดินยาวและระเบียงเปิด ซึ่งเป็นสภาพแวดล้อมที่มีความท้าทายอย่างมากในการระบุตำแหน่งให้แม่นยำ ภาพถ่ายอาคารวิศวกรรมศาสตร์ 3 แสดงได้ดังภาพที่ 9 โดยการเก็บข้อมูลนั้น จะแบ่งพื้นที่ตามทางเดินออกเป็นช่องขนาดเท่า ๆ กัน และตำแหน่งกึ่งกลางของแต่ละช่องคือพิกัดอ้างอิงสำหรับพื้นที่นั้น ๆ ในการเก็บข้อมูล แผลงผังอาคารได้แสดงไว้ดังภาพที่ 10 โดยจุดสีแดงในภาพคือพิกัดที่ห่างกันทุก ๆ 4 เมตร สำหรับการเก็บข้อมูลในการทดสอบนั้น จะประกอบไปด้วยข้อมูลทั้งหมด 2 ชุดข้อมูล โดยตารางที่ 2 ได้สรุปรายละเอียดสำหรับแต่ละชุดข้อมูล ซึ่งการเก็บข้อมูลชุดที่ 1 ได้ใช้โทรศัพท์มือถือทั้งหมด 3 รุ่น ได้แก่ Samsung Galaxy Note 4, Samsung Galaxy S5 และ LG Nexus 5X ส่วนการเก็บข้อมูลชุดที่ 2 ได้ใช้โทรศัพท์มือถือทั้งหมด 2 รุ่น ได้แก่ Samsung Galaxy S5 และ LG Nexus 5X



ภาพที่ 9 ภาพถ่ายอาคารวิศวกรรมศาสตร์ 3



ภาพที่ 10 แผนผังอาคารวิศวกรรมศาสตร์ 3

ตารางที่ 2 จำนวนข้อมูลลักษณะสัญญาณวายฟายและรายละเอียดของแต่ละชุดข้อมูลสำหรับทดสอบ

InHit

ชุดข้อมูล	จำนวนชั้นที่เก็บข้อมูล	จำนวนตำแหน่งที่เก็บข้อมูล	ขนาดของช่องสำหรับข้อมูลที่ใช้ฝึก (ม.)	ขนาดของช่องสำหรับข้อมูลที่ใช้ทดสอบ (ม.)
1	3	127	4	4
2	1	189	4	1

#### 4.1.3) การจำลอง

เพื่อที่จะวัดผลกระทบจากการเปลี่ยนแปลงของสภาพแวดล้อมต่าง ๆ กล่าวคือการหายไปของบีเอสเอสไอดีและการแปรปรวนของค่าความแรงสัญญาณ งานวิจัยชิ้นนี้จึงได้สร้างการจำลองของสถานการณ์ดังกล่าว

ในกรณีการหายไปของบีเอสเอสไอดีนั้น จะทำการสุ่มเลือกบีเอสเอสไอดีตามจำนวนที่ต้องการจากข้อมูลสำหรับการทดสอบ หลังจากนั้นจะลบบีเอสเอสไอดีที่ถูกเลือกเหล่านั้นออกจากชุดข้อมูลก่อนเริ่มการทดลองวัดผลความแม่นยำ

ในกรณีการแปรปรวนของค่าความแรงสัญญาณนั้น จะทำการสุ่มเลือกบีเอสเอสไอดีตามจำนวนที่ต้องจากข้อมูลสำหรับการทดสอบ หลังจากนั้นจะปรับค่าความแรงสัญญาณของบีเอสเอสไอดีที่ถูกเลือกเหล่านั้นจากแต่ละข้อมูลตามสมการที่ (19)

$$RSSI_{new} = 10 \times \log_{10}(10^{\frac{RSSI_{old}}{10}} \times \omega) \quad (19)$$

เมื่อ  $RSSI_{new}$  คือค่าความแรงสัญญาณที่ถูกเปลี่ยนแปลงแล้ว,  $RSSI_{old}$  คือค่าความแรงสัญญาณดั้งเดิม และ  $\omega$  คืออัตราร้อยละของพลังงานการส่งสัญญาณที่เหลือของตัวกระจายสัญญาณที่ให้กำเนิดบีเอสเอสไอดีนั้น ๆ

นอกจากนี้การจำลองการทดลองดังกล่าวจะมีการทำซ้ำทั้งหมด 10 รอบด้วยการสุ่มจากเมล็ด (seed) ต่าง ๆ กัน แล้วนำผลการทดลองดังกล่าวมาเฉลี่ยค่ากันเพื่อลดผลกระทบจากการแปรปรวนของการสุ่ม

## 4.2) มาตรการวัด

### 4.2.1) ความแม่นยำ

ความแม่นยำนั้นถูกวัดผลในอัตราร้อยละของจำนวนผลลัพธ์ที่ต้องตามสมการที่ (20) โดยในกรณีของโมดูลจำแนกในอาคาร/นอกอาคารนั้น  $n_{correct}$  คือจำนวนผลลัพธ์พื้นที่ที่ต้อง ในกรณีของโมดูลจำแนกอาคารนั้น  $n_{correct}$  คือจำนวนผลลัพธ์อาคารที่ต้อง ในกรณีของอัลกอริทึมระบุตำแหน่งภายในอาคารนั้น  $n_{correct}$  คือจำนวนผลลัพธ์ชั้นที่ต้อง และ  $n_{all}$  คือจำนวนข้อมูลที่นำมาทดสอบ

$$Accuracy = \frac{n_{correct}}{n_{all}} \times 100 \quad (20)$$

### 4.2.2) ระยะทางผิดพลาด

ระยะทางผิดพลาดคือการวัดระยะทางระหว่างตำแหน่งที่เก็บข้อมูลจริงและตำแหน่งของผลลัพธ์การระบุตำแหน่งของอัลกอริทึมโดยใช้ระยะทางยูคลิเดียนดังสมการที่ (21) เมื่อ  $x = \{x_1, x_2\}$  คือตำแหน่งที่เก็บข้อมูลจริง และ  $y = \{y_1, y_2\}$  คือตำแหน่งของผลลัพธ์การระบุตำแหน่งของอัลกอริทึม

$$Error\ distance = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad (21)$$

#### 4.2.3) เวลาสะสมในการประมวลผล

เวลาสะสมในการประมวลผลคือผลรวมเวลาในการประมวลผลข้อมูลของอัลกอริทึม โดยการใช้สมการที่ (22) เมื่อ  $s_i$  คือตัวอย่างที่  $i$ ,  $p(s_i)$  คือเวลาในการประมวลผลของตัวอย่างที่  $i$  ของอัลกอริทึม และ  $n$  คือจำนวนตัวอย่างทั้งหมดที่ถูกนำมาทดสอบ

$$\text{Cumulative processing time} = \sum_{i=1}^n p(s_i) \quad (22)$$

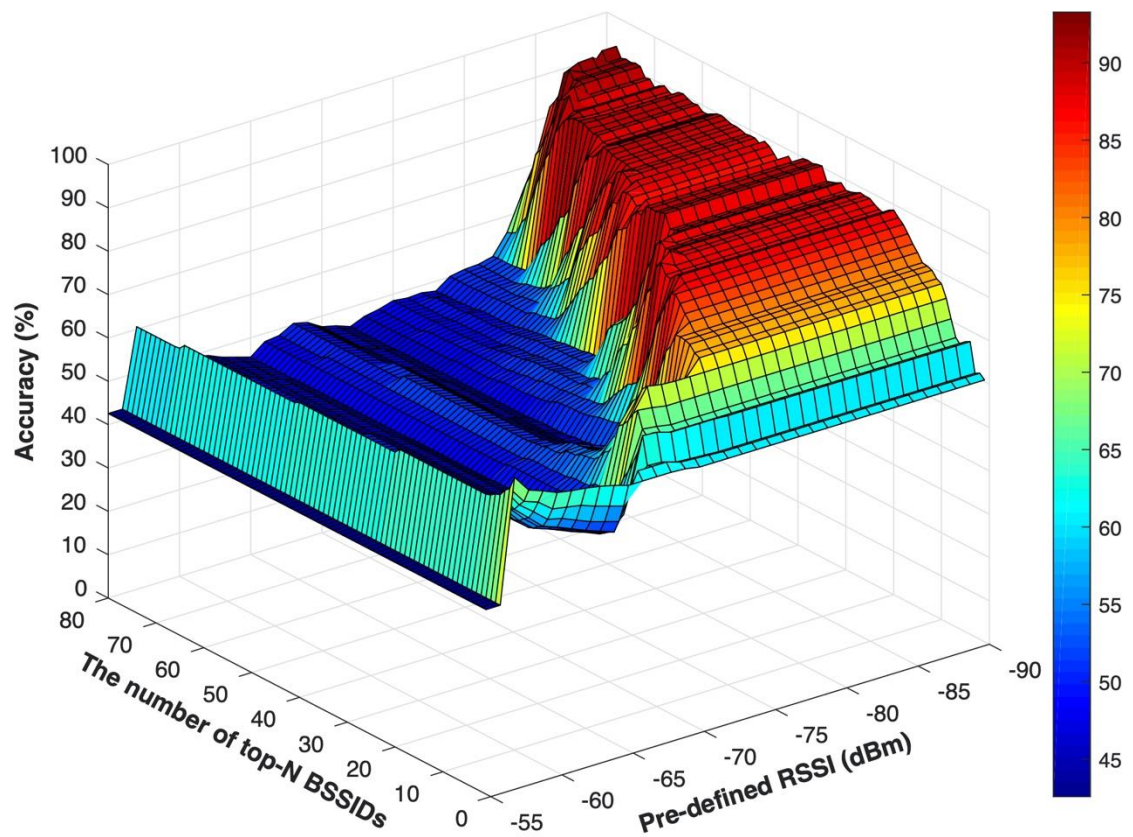
### 4.3) ผลการทดลอง

#### 4.3.1) โมดูลจำแนกในอาคาร/นอกอาคาร

ในการหาค่าคงที่ top- $N$  BSSIDs และ *pre-defined RSSI* ที่เหมาะสม การทดลองนี้จึงได้ปรับเปลี่ยนค่า top- $N$  BSSIDs และ *pre-defined RSSI* ต่าง ๆ เพื่อหาค่าที่ให้ความแม่นยำสูงสุด โดยการทดลองนี้จะใช้ชุดข้อมูลที่ 1 เนื่องจากชุดข้อมูลนี้ประกอบไปด้วยตัวอย่างข้อมูลที่เก็บมาจากภายในอาคารและภายนอกอาคาร ยิ่งไปกว่านั้น การทดลองนี้ได้ใช้โทรศัพท์มือถือหลายเครื่องในการทดสอบ เพื่อวัดผลกระทบจากความหลากหลายของโทรศัพท์มือถือ

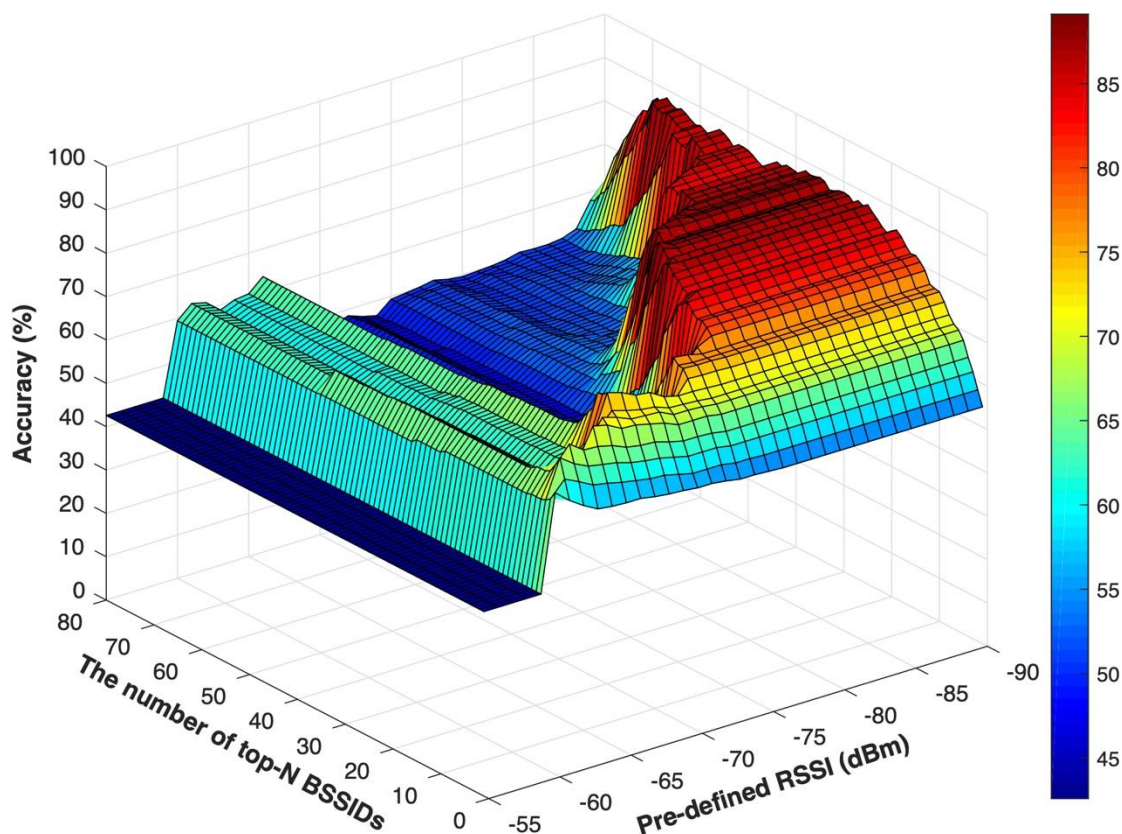
การทดลองนี้เริ่มด้วยการตั้งค่าคงที่ top-1 BSSID และ *pre-defined RSSI* เป็น -90 dBm หลังจากนั้นจึงเพิ่มค่า top- $N$  และ *pre-defined RSSI* เป็น top-80 และ -55 dBm ตามลำดับ โดยผลลัพธ์ความแม่นยำของโมดูลจำแนกในอาคาร/นอกอาคารถูกแสดงในภาพที่ 11 และความแม่นยำสูงสุดสำหรับแต่ละกรณีถูกแสดงไว้ในตารางที่ 3

จากตารางที่ 3 โมดูลจะได้ความแม่นยำสูงสุดเมื่อทดสอบด้วยเครื่อง LG Nexus 5X เพราะส่วนเชื่อมต่อไร้สายของ LG Nexus 5X นั้นเป็นเทคโนโลยีที่ใหม่กว่า จึงมีความอ่อนไหวต่อสัญญาณมากกว่า Samsung Galaxy S5 แต่อย่างไรก็ตาม เมื่อดูจากภาพรวมทั้งหมดความแม่นยำสูงสุดในทุก ๆ กรณีที่โมดูลนี้สามารถทำได้ในโทรศัพท์มือถือทุกเครื่องนั้นแตกต่างกันเพียงเล็กน้อย



(a) ทดสอบด้วย LG Nexus 5X





(b) ทดสอบด้วย Samsung Galaxy S5

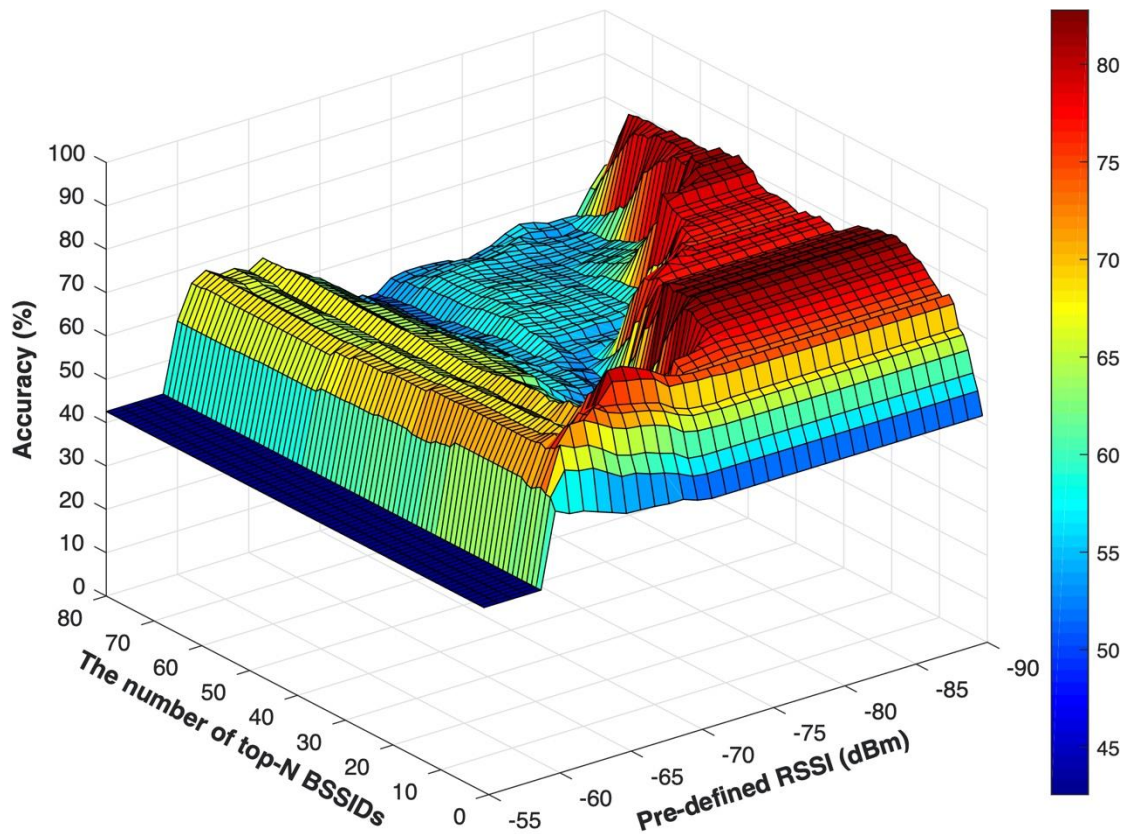
ภาพที่ 11 ความแม่นยำของโมดูลจำแนกในอาคาร/นอกอาคาร

ตารางที่ 3 ความแม่นยำสูงสุดและค่าคงที่ที่ใช้ของโมดูลจำแนกในอาคาร/นอกอาคาร

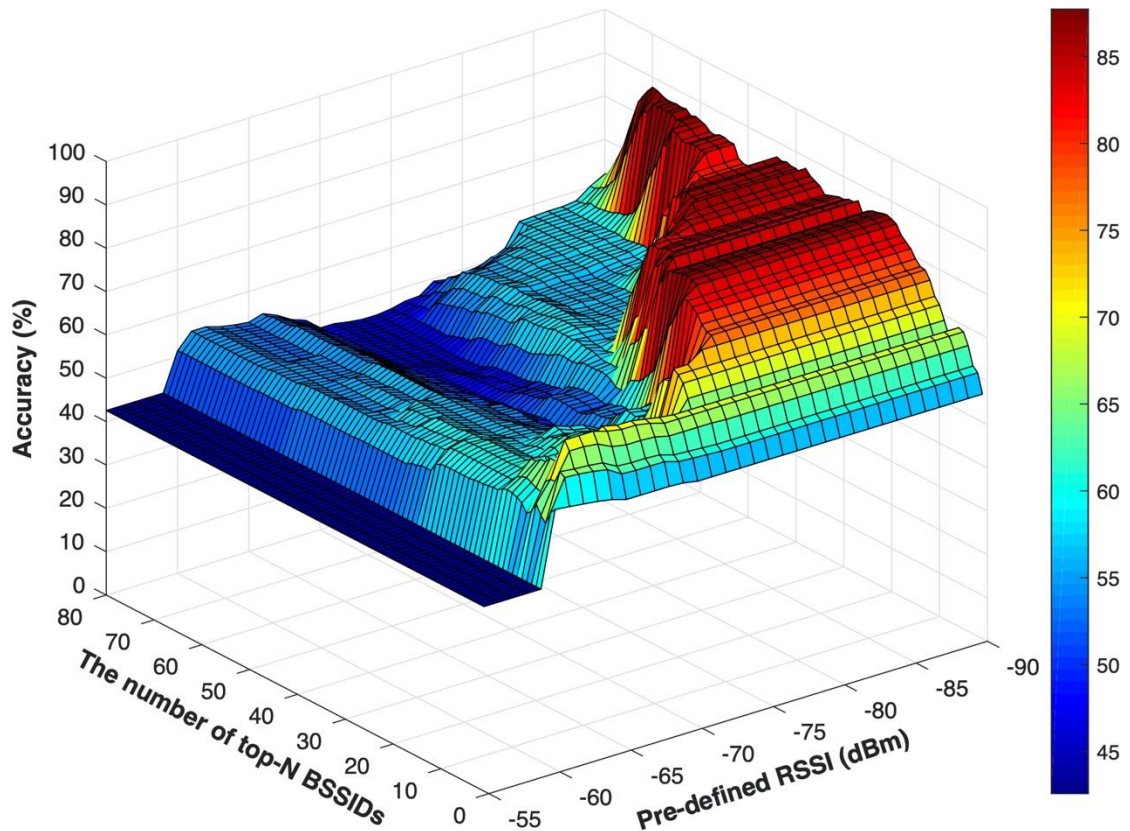
ทดสอบด้วย	ความแม่นยำ (%)	<i>Pre-defined RSSI</i>	top-N
LG Nexus 5X	93.27	-88	77
Samsung Galaxy S5	89.11	-77	30

เพื่อวัดผลกระทบจากการเก็บข้อมูลและทดสอบด้วยโทรศัพท์คนละเครื่องนั้น ในการทดลองนี้จึงใช้ข้อมูลจาก Samsung Galaxy S5 ในการสร้างลายนิ้วมือของสัญญาณ และใช้ Samsung Galaxy S5 อีกเครื่องหนึ่งและ LG Nexus 5X ในการทดสอบหาความแม่นยำ ผลลัพธ์ความแม่นยำถูกแสดงไว้ในภาพที่ 12 และความแม่นยำสูงสุดสำหรับแต่ละกรณีถูกแสดงไว้ที่ตารางที่ 4 จากตารางดังกล่าวจะเห็นว่า เมื่อทดสอบด้วยเครื่องที่แตกต่างกับเครื่องที่ใช้เก็บข้อมูลเพื่อสร้างลายนิ้วมือของสัญญาณนั้น ความแม่นยำจะลดลงเล็กน้อยเนื่องจากส่วนเชื่อมต่อสัญญาณไร้สายของแต่ละเครื่องนั้นมีความแตกต่างกัน จึงทำให้เกิดลักษณะสัญญาณที่แตกต่างกัน แต่อย่างไรก็ตาม ความแม่นยำที่โมดูลทำได้ในแต่ละกรณี ก็

ยังคงสูงอยู่ ดังนั้นโมดูลจำแนกในอาคาร/นอกอาคารจึงสามารถใช้งานบนโทรศัพท์ที่  
หลากหลายได้



(a) ทดสอบด้วย Samsung Galaxy S5



(b) ทดสอบด้วย LG Nexus 5X

ภาพที่ 12 ความแม่นยำของโมดูลจำแนกในอาคาร/นอกอาคารเมื่อสร้างลายนิ้วมือของสัญญาณและทดสอบด้วยเครื่องที่แตกต่างกัน

ตารางที่ 4 ความแม่นยำสูงสุดและค่าคงที่ที่ใช้ของโมดูลจำแนกในอาคาร/นอกอาคารเมื่อสร้างลายนิ้วมือของสัญญาณและทดสอบด้วยเครื่องที่แตกต่างกัน

ทดสอบด้วย	ความแม่นยำ (%)	Pre-defined RSSI	top-N
Samsung Galaxy S5	82.77	-76	19
LG Nexus 5X	87.72	-90	70

เพื่อวัดผลกระทบจากการเปลี่ยนแปลงของสภาพแวดล้อม การทดลองต่อไปนี้จะใช้การจำลองในการเปลี่ยนแปลงสภาพแวดล้อมก่อนการทดสอบ เนื่องจากในสถานการณ์จริงจะไม่สามารถควบคุมสภาพแวดล้อมได้และไม่สามารถระบุได้อย่างถูกต้องแน่นอนว่าสภาพแวดล้อมได้เปลี่ยนไปอย่างไรบ้าง ดังนั้นเพื่อวัดผลดังกล่าวจึงจำเป็นต้องใช้การจำลอง เพราะการจำลองนั้นสามารถกำหนดสภาพแวดล้อมที่ต้องการได้ สำหรับการเลือกค่าคงที่ที่เหมาะสมที่สุดสำหรับการจำลองนั้น อ้างอิงจากภาพที่ 11 อัลกอริทึมจะได้ผลลัพธ์ความ



แม่นยำที่สูงสำหรับทั้งสองเครื่องเมื่อใช้ *pre-defined RSSI* เป็น -86 ถึง -90 dBm และ top-30 ถึง top-40 BSSIDs ดังนั้นจึงเลือกค่าเฉลี่ยจากค่าคงที่ดังกล่าวก็คือ top-35 BSSIDs และ *pre-defined RSSI* เป็น -88 dBm เพื่อเป็นค่าคงที่สำหรับอัลกอริทึมสำหรับการจำลองนี้

ในกรณีของบีเอสเอสไอดีที่หายไป การทดลองเริ่มจากการลบบีเอสเอสไอดีร้อยละ 10 ของข้อมูลทดสอบ หลังจากนั้นจึงเพิ่มจำนวนบีเอสเอสไอดีที่ถูกลบเป็นร้อยละ 70 ผลลัพธ์ความแม่นยำของการจำลองนี้ถูกแสดงไว้ในตารางที่ 5 เมื่อดูจากตารางดังกล่าวแล้ว ความแม่นยำลดลงอย่างเห็นได้ชัดเมื่อจำนวนบีเอสเอสไอดีที่หายไปนั้นมากกว่าร้อยละ 10 เนื่องจากเมื่อมีบีเอสเอสไอดีหายไปมากกว่าร้อยละ 10 แล้ว คะแนนจากการคำนวณของสมการที่ (10) นั้นลดลงจนน้อยกว่าค่า *adaptive threshold score* ดังนั้นอัลกอริทึมจึงจำแนกคำร้องขอดังกล่าวเป็นคำร้องที่มาจากภายนอกอาคารทั้ง ๆ ที่ถูกส่งมาจากภายในอาคาร จึงทำให้ประสิทธิภาพการทำงานโดยรวมของระบบลดลง ซึ่งในเนื้อหาในบทที่ 4.3.4 จะแสดงให้เห็นว่า MissingHit นั้นสามารถลดผลกระทบจากการหายไปของบีเอสเอสไอดีได้ และเพิ่มความแม่นยำในกรณีนี้ได้

ตารางที่ 5 ความแม่นยำของโมดูลจำแนกในอาคาร/นอกอาคารในสถานการณ์ที่บีเอสเอสไอดีหายไป

บีเอสเอสไอดี ที่หายไป	0%	10%	20%	30%	40%	50%	60%	70%
ความแม่นยำ (%)	90.30	88.67	84.38	80.02	74.50	70.20	64.89	58.79

ในกรณีของค่าความแรงสัญญาณที่แปรปรวน การทดลองเริ่มจากการเลือกร้อยละ 25 ของบีเอสเอสไอดีของข้อมูลทดสอบและลดพลังงานในการส่งของบีเอสเอสไอดีที่ถูกเลือกนั้นเหลือร้อยละ 90 ของพลังงานในการส่งดั้งเดิม หลังจากนั้นจึงเพิ่มจำนวนบีเอสเอสไอดีที่ถูกเลือกเป็นร้อยละ 75 และลดพลังงานในการส่งเป็นร้อยละ 40 ความแม่นยำของสถานการณ์ต่าง ๆ ดังกล่าวถูกแสดงไว้ในตารางที่ 6 จากภาพรวมทั้งหมด ความแม่นยำในแต่ละกรณีนั้นแตกต่างกันเพียงเล็กน้อย เนื่องจากสมการที่ (19) ซึ่งอธิบายความสัมพันธ์ระหว่างค่าความแรงสัญญาณและพลังงานในการส่งนั้น ถ้าหากจำนวน top-N ที่ใช้มีค่าสูงมากเพียงพอแล้ว อัลกอริทึมก็สามารถทนต่อการเปลี่ยนแปลงของค่าความแรงสัญญาณได้ ดังนั้นจึงสรุปได้ว่าโมดูลจำแนกในอาคาร/นอกอาคารนั้นทนทานต่อการเปลี่ยนแปลงของค่าความแรงสัญญาณ

ตารางที่ 6 ความแม่นยำของโมดูลจำแนกในอาคาร/นอกอาคารในสถานการณ์ที่ค่าความแรงสัญญาณแปรปรวน

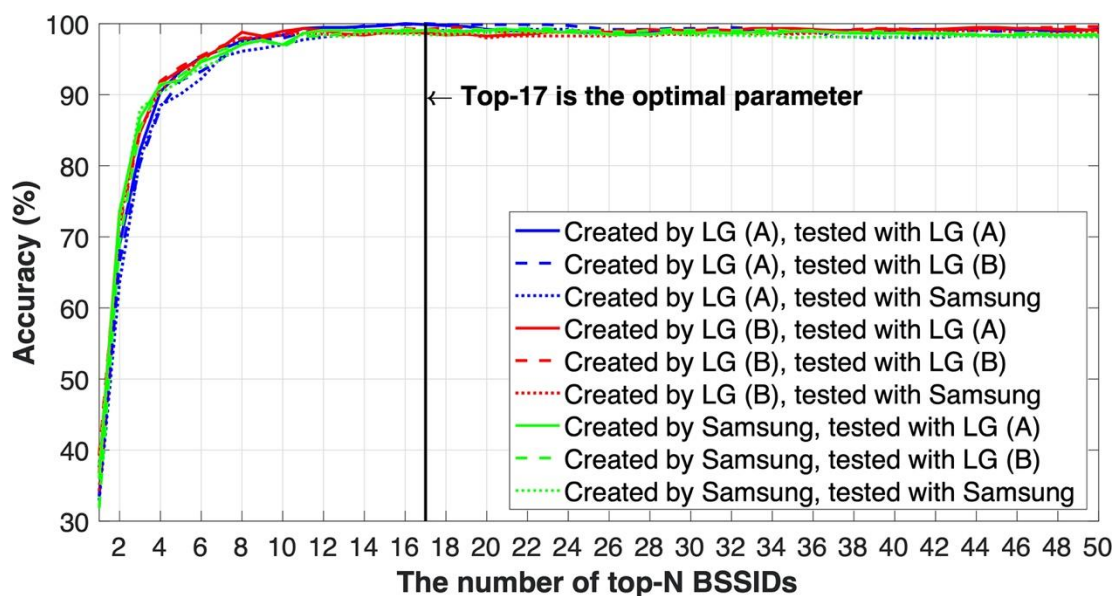
ปีเอสเอสไอดี ที่ถูกเลือก	พลังงานในการส่งที่เหลือ						
	100%	90%	80%	70%	60%	50%	40%
25%	90.30	90.30	90.30	90.32	90.77	90.99	90.99
50%	90.30	90.30	90.30	90.57	91.12	90.43	90.44
75%	90.30	90.30	90.30	90.79	91.01	90.25	90.26

#### 4.3.2) โมดูลจำแนกอาคาร

ในการทดลองเพื่อวัดผลความแม่นยำของโมดูลนี้ จะใช้ข้อมูลชุดที่ 2 และ 3 เนื่องจากชุดข้อมูลทั้งสองนั้นถูกเก็บมาจากหลายอาคารภายในงานนิทรรศการ ซึ่งในการทดลองนี้จะใช้ค่าคงที่ top-N BSSIDs หลายค่าเพื่อหาค่าที่ดีที่สุดสำหรับโมดูลนี้ในการจำแนกอาคาร และโทรศัพท์มือถือหลายเครื่องในการทดสอบ เพื่อใช้วัดผลกระทบจากการใช้โทรศัพท์ต่างเครื่องในการจำแนกอาคาร

ภาพที่ 13 แสดงความแม่นยำของโมดูลจำแนกอาคาร โดยเริ่มการทดสอบจากค่า top-1 BSSID และเพิ่มค่าคงที่นั้นจนถึง top-50 BSSIDs สำหรับทุก ๆ กรณีของโทรศัพท์มือถือ โดยตารางที่ 7 ได้สรุปค่าแม่นยำที่สูงที่สุดสำหรับแต่ละกรณี

จากตารางที่ 7 เมื่อทดสอบด้วย LG Nexus 5X แล้ว โมดูลสามารถจำแนกอาคารได้แม่นยำกว่าในทุกกรณี เมื่อเทียบกับการทดสอบด้วย Samsung Galaxy S5 เนื่องจากส่วนเชื่อมต่อสัญญาณไร้สายของ LG Nexus 5X นั้นเป็นเทคโนโลยีที่ใหม่กว่า Samsung Galaxy S5 ดังนั้น LG Nexus 5X จึงมีความอ่อนไหวต่อสัญญาณมากกว่า แต่อย่างไรก็ตาม จากภาพรวมทั้งหมดนั้นความแม่นยำสูงสุดในแต่ละกรณีนั้นแตกต่างกันเพียงเล็กน้อย จึงสามารถสรุปได้ว่าโมดูลจำแนกอาคารนั้นทนทานต่อความแตกต่างของโทรศัพท์มือถือ



ภาพที่ 13 ความแม่นยำของโมดูลจำแนกอาคาร

ตารางที่ 7 ความแม่นยำสูงสุดและค่าคงที่ที่ใช้ของโมดูลจำแนกอาคาร

สร้างลายนิ้วมือของ สัญญาณด้วย	ทดสอบด้วย	ความแม่นยำ (%)	top-N
Nexus 5X (A)	Nexus 5X (A)	100.00	16
	Nexus 5X (B)	100.00	17
	Samsung S5	99.20	22
Nexus 5X (B)	Nexus 5X (A)	99.46	44
	Nexus 5X (B)	99.59	49
	Samsung S5	99.20	45
Samsung S5	Nexus 5X (A)	99.19	21
	Nexus 5X (B)	99.31	21
	Samsung S5	98.80	21

สำหรับการเลือกค่าคงที่ที่ดีที่สุดสำหรับโมดูลนี้ จะพิจารณาจากผลการทดลองข้างต้น จึงสรุปได้ว่าการใช้ LG Nexus 5X และใช้ top-17 จะเป็นค่าที่เหมาะสมที่สุด เรื่องจากเงื่อนไขดังกล่าวสามารถให้ความแม่นยำร้อยละ 100 เมื่อทดสอบกับโทรศัพท์รุ่นเดียวกัน และได้ความแม่นยำร้อยละ 98.93 เมื่อทดสอบกับโทรศัพท์ต่างรุ่น แต่อย่างไรก็ตาม เมื่อพิจารณาภาพที่ 13 แล้ว ความแม่นยำมีแนวโน้มที่จะเสถียรเมื่ออัลกอริทึมใช้ค่า top-10 ขึ้น

ไป ดังนั้นหากการใช้งานในสถานการณ์อื่นที่ต้องพิจารณาเวลาในการประมวลผลเป็นหลัก การทดลองนี้สามารถชี้ให้เห็นว่าการใช้ค่าคงที่ top-11 ก็สามารถให้ความแม่นยำที่สูงมาก และแทบจะไม่แตกต่างกับกรณีที่ใช้ top-N สูง ในขณะที่ใช้เวลาประมวลผลน้อยกว่า

เพื่อวัดการเปลี่ยนแปลงของสภาพแวดล้อมที่มีบีเอสเอสไอดีที่หายไปและมีการแปรปรวนของค่าความแรงสัญญาณนั้น การทดลองต่อไปนี้จะใช้ LG Nexus 5X เป็นเครื่องมือในการทดสอบ ในกรณีของบีเอสเอสไอดีที่หายไปนั้น ได้เริ่มการทดลองโดยการลบ บีเอสเอสไอดีร้อยละ 10 และเพิ่มไปจนถึงร้อยละ 70 โดยใช้ top-17 เป็นค่าคงที่สำหรับ อัลกอริทึม เนื่องจากให้ค่าความแม่นยำสูงสุด ผลลัพธ์การทดลองได้แสดงไว้ในตารางที่ 8 จากการพิจารณาตารางดังกล่าว จะสังเกตเห็นได้ว่าความแม่นยำลดลงเล็กน้อยเมื่อจำนวนบีเอสเอสไอดีที่หายไปเพิ่มขึ้น ซึ่งในบทที่ 4.3.4 จะแสดงให้เห็นว่าการใช้ MissingHit นั้นสามารถเพิ่มความแม่นยำในสถานการณ์เหล่านี้ได้

ในกรณีของค่าความแรงสัญญาณที่แปรปรวน เริ่มการทดลองจากการเลือกบีเอสเอสไอดีร้อยละ 25 และลดพลังงานในการส่งของบีเอสเอสไอดีที่ถูกเลือกนั้นเหลือร้อยละ 90 ของพลังงานในการส่งดั้งเดิม หลังจากนั้นจึงเพิ่มจำนวนบีเอสเอสไอดีที่ถูกเลือกเป็นร้อยละ 75 และลดพลังงานในการส่งเป็นร้อยละ 40 ความแม่นยำของสถานการณ์ต่าง ๆ ดังกล่าวถูกแสดงไว้ในตารางที่ 9 จากตารางดังกล่าว ความแม่นยำในแต่ละกรณีนั้นแตกต่างกันเพียงเล็กน้อย ดังนั้นจึงสรุปได้ว่าโมดูลจำแนกอาคารนั้นทนทานต่อความแปรปรวนของค่าความแรงสัญญาณ

ตารางที่ 8 ความแม่นยำของโมดูลจำแนกอาคารในสถานการณ์ที่บีเอสเอสไอดีหายไป

บีเอสเอสไอดี ที่หายไป	0%	10%	20%	30%	40%	50%	60%	70%
ความแม่นยำ (%)	100	97.39	97.14	97.02	96.88	96.53	96.1	95.22

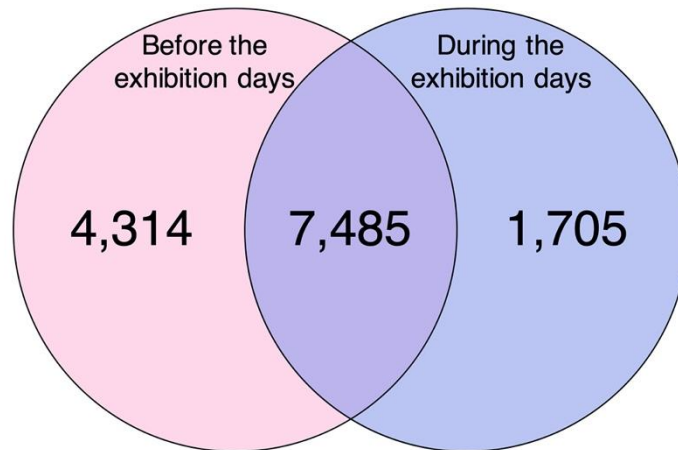
ตารางที่ 9 ความแม่นยำของโมดูลจำแนกอาคารในสถานการณ์ที่ค่าความแรงสัญญาณแปรปรวน

ปีเอสเอสไอดีที่ ถูกเลือก	พลังงานในการส่งที่เหลือ						
	100%	90%	80%	70%	60%	50%	40%
25%	100	99.73	99.73	99.65	99.61	99.62	99.62
50%	100	99.73	99.73	99.64	99.61	99.62	99.62
75%	100	99.73	99.73	99.64	99.61	99.64	99.64

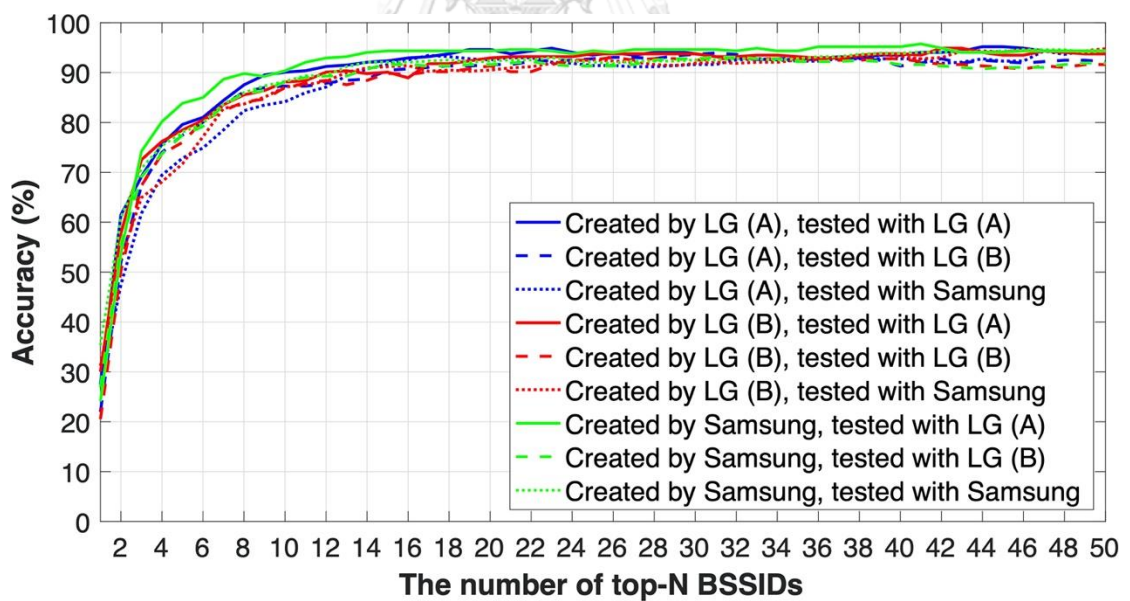
นอกจากนี้ยังได้ทดสอบโมดูลจำแนกอาคารในสถานการณ์จริงที่มีปีเอสเอสไอดีหายไป โดยภาพที่ 14 ได้สรุปข้อมูลปีเอสเอสไอดีระหว่างก่อนวันงานนิทรรศการและระหว่างวันงานนิทรรศการด้วยการใช้ข้อมูลชุดที่ 2 และ 4 จำนวนปีเอสเอสไอดีที่พบจากทั้งก่อนวันงานนิทรรศการและระหว่างวันงานนิทรรศการนั้น มีทั้งหมด 7,485 ปีเอสเอสไอดี จำนวนปีเอสเอสไอดีที่พบเฉพาะก่อนวันงานนิทรรศการนั้น มีทั้งหมด 4,314 ปีเอสเอสไอดี โดยปีเอสเอสไอดีกลุ่มนี้คือปีเอสเอสไอดีที่หายไป ซึ่งคิดเป็นร้อยละ 36 ของปีเอสเอสไอดีที่พบก่อนวันงานนิทรรศการ และจำนวนปีเอสเอสไอดีที่พบเฉพาะระหว่างวันงานนิทรรศการ มีทั้งหมด 1,750 ปีเอสเอสไอดี โดยปีเอสเอสไอดีกลุ่มนี้คือปีเอสเอสไอดีที่ไม่รู้จัก ดังนั้นจากข้อมูลเหล่านี้ จะพบว่าจำนวนปีเอสเอสไอดีนั้นเปลี่ยนไปอย่างมาก

เพื่อวัดผลกระทบจากการเปลี่ยนแปลงดังกล่าว จึงได้ใช้ข้อมูลชุดที่ 2 และ 4 ในการทดลองต่อไปนี้ เนื่องจากชุดข้อมูลที่ 2 ถูกเก็บก่อนวันงานนิทรรศการ และข้อมูลชุดที่ 4 ถูกเก็บระหว่างวันงานนิทรรศการ ภาพที่ 15 ได้แสดงผลลัพธ์ความแม่นยำของโมดูลนี้ ซึ่งเริ่มการทดสอบด้วย top-1 BSSID และเพิ่มไปจนถึง top-50 BSSIDs สำหรับทุก ๆ กรณีของโทรศัพท์มือถือ ตารางที่ 10 ได้สรุปความแม่นยำสูงสุดสำหรับแต่ละกรณี จากตารางดังกล่าว จะพบว่าความแม่นยำโดยรวมในการทดลองนี้ น้อยกว่าการทดลองก่อนหน้า และมีแนวโน้มที่จะต้องใช้ top-N ที่สูงขึ้นเพื่อให้ได้ความแม่นยำที่ดีขึ้น เนื่องจากผลกระทบจากการเปลี่ยนแปลงของสภาพแวดล้อมที่มีจำนวนปีเอสเอสไอดีแตกต่างกันอย่างชัดเจน แต่อย่างไรก็ตามความแม่นยำสูงสุดในแต่ละกรณีนั้นลดลงเพียงเล็กน้อย หรือประมาณร้อยละ 3 ถึง 7 ดังนั้นจึงสามารถสรุปได้ว่าโมดูลจำแนกอาคารนั้นทนทานต่อการเปลี่ยนแปลงของสภาพแวดล้อม ยิ่งไปกว่านั้นผลลัพธ์จากการทดลองด้วยข้อมูลจริงนี้ยังสอดคล้องกับผลลัพธ์

จากการจำลองก่อนหน้านี้ด้วย กล่าวคือความแม่นยำลดลงประมาณร้อยละ 3 เมื่อมีบีเอสเอสไอดีหายไปประมาณร้อยละ 30 ถึง 40 ในกรณีของการใช้ LG Nexus 5X



ภาพที่ 14 การเปลี่ยนแปลงของจำนวนบีเอสเอสไอดีระหว่างก่อนวันงานนิทรรศการและระหว่างวันงานนิทรรศการ



ภาพที่ 15 ความแม่นยำของโมดูลจำแนกอาคารในสถานการณ์จริงที่สภาพแวดล้อมเปลี่ยนแปลง

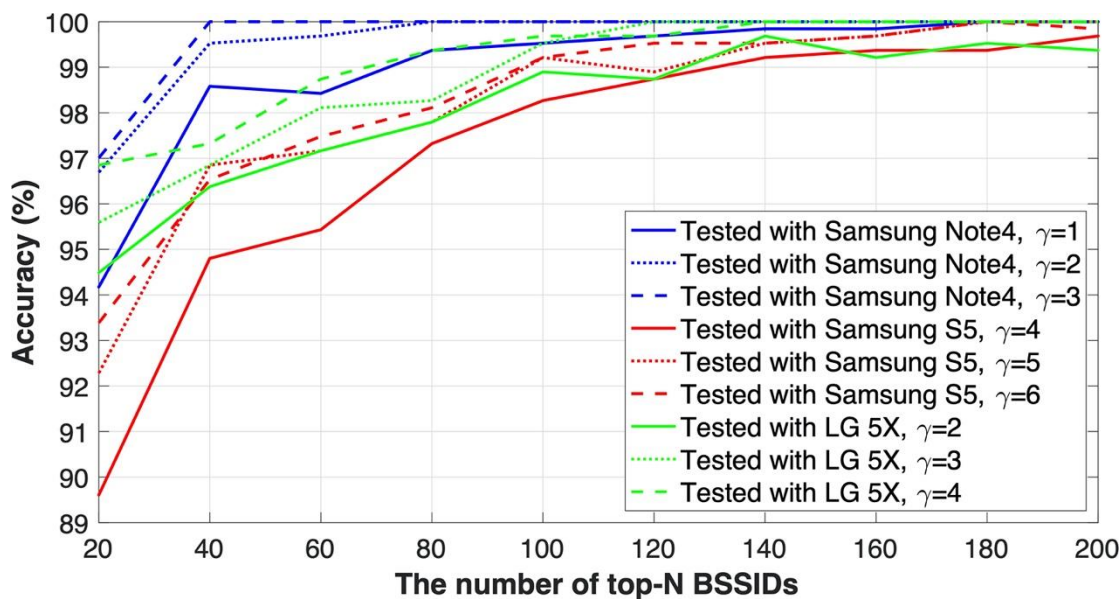
ตารางที่ 10 ความแม่นยำสูงสุดและค่าคงที่ที่ใช้ของโมดูลจำแนกอาคารในสถานการณ์ที่สภาพแวดล้อมเปลี่ยนไป

สร้างลายนิ้วมือของสัญญาณด้วย	ทดสอบด้วย	ความแม่นยำ (%)	top-N
Nexus 5X (A)	Nexus 5X (A)	95.18	44
	Nexus 5X (B)	93.93	31
	Samsung S5	94.34	47
Nexus 5X (B)	Nexus 5X (A)	94.90	42
	Nexus 5X (B)	93.06	26
	Samsung S5	94.80	50
Samsung S5	Nexus 5X (A)	95.75	41
	Nexus 5X (B)	93.06	31
	Samsung S5	94.57	46

#### 4.3.3) อัลกอริทึม InHit

การวัดผลความแม่นยำของอัลกอริทึม InHit นั้นจะแบ่งออกเป็น 2 ส่วนหลัก คือ ความแม่นยำในการระบุชั้นและความแม่นยำในการระบุตำแหน่ง

การวัดผลความแม่นยำในการระบุชั้น จะใช้ชุดข้อมูลที่ 1 ที่ได้อธิบายไว้ในบทที่ 4.1.2 เนื่องจากเป็นชุดข้อมูลที่เก็บมาจากหลายชั้นของอาคาร ซึ่งในการทดลองนี้จะกำหนดค่า  $k$  เป็น 1 และกำหนดค่าต่าง ๆ ให้กับค่าคงที่ top-N BSSIDs และค่า  $\gamma$  เพื่อหาค่าที่เหมาะสมที่สุดสำหรับอัลกอริทึม นอกจากนี้ยังใช้โทรศัพท์มือถือหลายเครื่องในการทดสอบด้วย เพื่อหาผลกระทบจากปัจจัยดังกล่าว โดยการทดลองจะใช้ Samsung Galaxy Note 4 ในการสร้างลายนิ้วมือของสัญญาณ เริ่มจากการตั้งค่า top-20 BSSIDs และ  $\gamma$  เป็น 0 dBm หลังจากนั้นจึงได้เพิ่มค่าเป็น top-200 และ 10 dBm สำหรับค่า  $\gamma$  ผลลัพธ์ความแม่นยำในการระบุชั้นได้แสดงไว้ในภาพที่ 16 และตารางที่ 11 ได้สรุปผลลัพธ์ความแม่นยำสูงสุดในการระบุชั้นและค่าคงที่ที่ใช้ในกรณีต่าง ๆ จากภาพดังกล่าว จะพบว่าความแม่นยำของ InHit มีแนวโน้มที่จะใช้ top-N ที่สูงขึ้นเพื่อให้มีความแม่นยำที่สูงขึ้น เมื่อทดสอบกับโทรศัพท์มือถือรุ่นอื่น โดยผลลัพธ์สุดท้ายแล้ว InHit สามารถให้ความแม่นยำในการระบุชั้นที่ร้อยละ 100



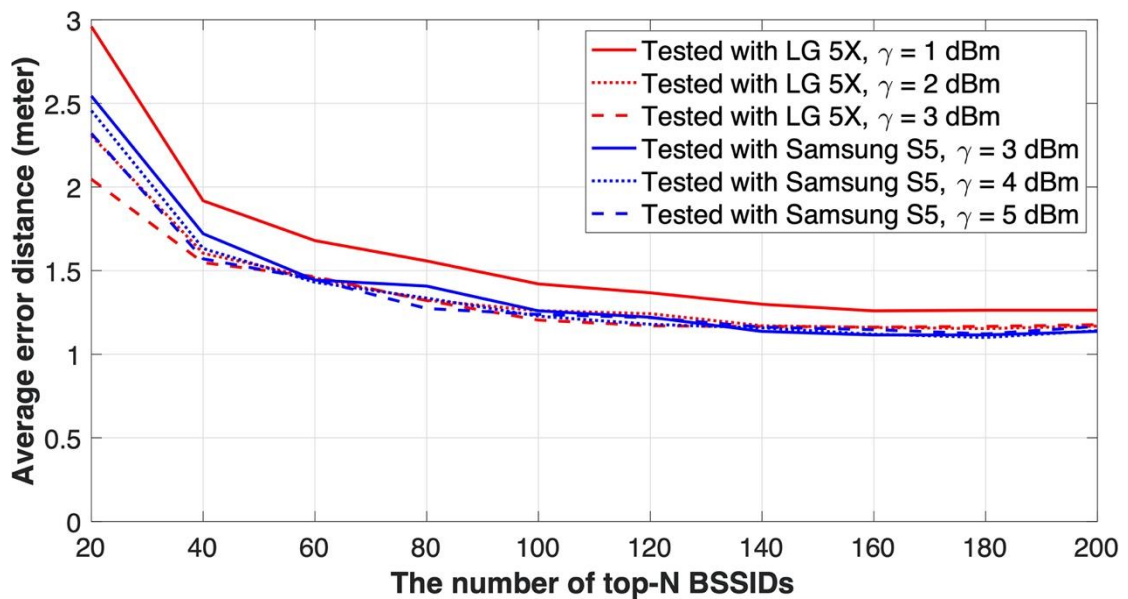
ภาพที่ 16 ความแม่นยำในการระบุชั้นของอัลกอริทึม InHit

ตารางที่ 11 ความแม่นยำสูงสุดและค่าคงที่ที่ใช้ของอัลกอริทึม InHit ในการระบุชั้น

ทดสอบด้วย	ความแม่นยำในการระบุชั้น (%)	ค่าคงที่ที่ใช้
Samsung Note 4	100	top-80, $\gamma = 2$
Samsung S5	100	top-180, $\gamma = 5$
Nexus 5X	100	top-120, $\gamma = 3$

การวัดผลความแม่นยำในการระบุตำแหน่งนั้น จะใช้ข้อมูลชุดที่ 2 ที่ได้อธิบายไว้ในบทที่ 4.1.2 ในการวัดผล เนื่องจากเป็นข้อมูลที่ระยะห่างในการเก็บข้อมูลที่ใช้ฝึกมากกว่าข้อมูลที่ใช้ทดสอบ ข้อมูลนี้จึงสามารถทำให้เกิดปัญหาตำแหน่งที่ไม่ตรงกันได้ โดยการทดลองนี้จะเริ่มกำหนดค่า  $k$  เป็น 1 และกำหนดค่าต่าง ๆ ให้กับค่าคงที่ top-N BSSIDs และค่า  $\gamma$  เพื่อหาค่าที่เหมาะสมที่สุดสำหรับอัลกอริทึม หลังจากนั้นจึงจะใช้ค่าคงที่ top-N BSSIDs และค่า  $\gamma$  ที่เหมาะสมที่สุดนี้ในการหาค่า  $k$  ที่เหมาะสมที่สุดต่อไป ผลลัพธ์ในการระบุตำแหน่งได้แสดงไว้ในภาพที่ 17 และตารางที่ 12 ได้สรุปผลลัพธ์ระยะทางผิดพลาดเฉลี่ยน้อยที่สุดและค่าคงที่ที่ใช้ในกรณีต่าง ๆ เมื่อพิจารณาจากภาพ จะพบว่าระยะทางผิดพลาดเฉลี่ยนั้นมีแนวโน้มลดลงอย่างมากจาก top-20 ถึง top-100 และลดลงเพียงเล็กน้อยตั้งแต่ top-100 ขึ้นไป



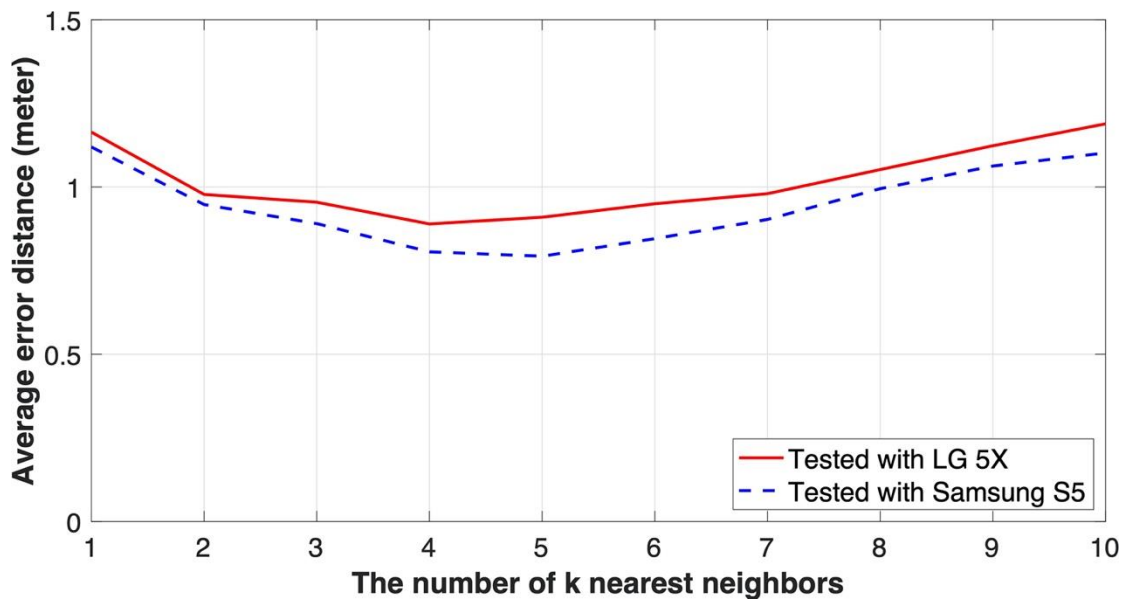


ภาพที่ 17 ระยะทางผิดพลาดเฉลี่ยของอัลกอริทึม InHit เมื่อค่า  $k$  เป็น 1

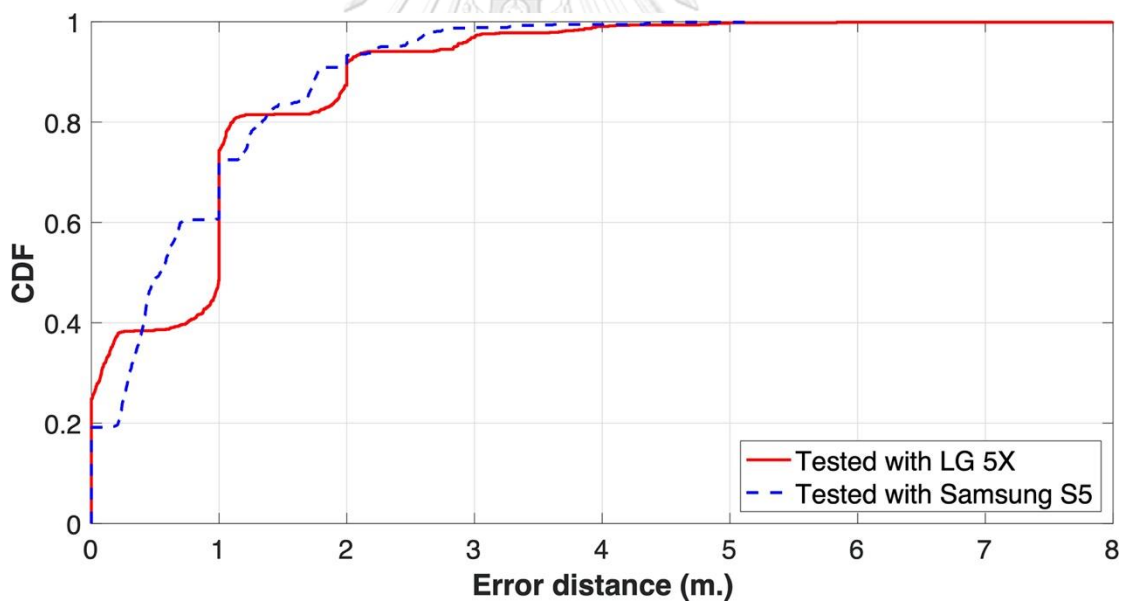
ตารางที่ 12 ระยะทางผิดพลาดเฉลี่ยน้อยสุดของอัลกอริทึม InHit เมื่อค่า  $k$  เป็น 1

ทดสอบด้วย	ระยะทางผิดพลาดเฉลี่ยน้อยสุด (ม.)	ค่าคงที่ที่ใช้
Nexus 5X	1.15	top-180, $\gamma = 2$
Samsung S5	1.1	top-180, $\gamma = 4$

ในการทดลองต่อไป จะเป็นการทดลองเพื่อหาค่า  $k$  ที่เหมาะสมที่สุด โดยค่าคงที่ที่ใช้จะใช้ค่าคงที่ในตารางที่ 12 สำหรับโทรศัพท์มือถือรุ่นนั้น ๆ ผลลัพธ์การทดลองได้แสดงไว้ในภาพที่ 18 ซึ่งระยะทางผิดพลาดเฉลี่ยน้อยสุดในกรณีของ LG Nexus 5X นั้นคือ 0.89 เมตร เมื่อ  $k$  เป็น 4 ส่วนในกรณีของ Samsung Galaxy S5 คือ 0.79 เมตร เมื่อ  $k$  เป็น 5 ดังนั้นจึงได้นำผลการทดลองของกรณีดังกล่าวมาแสดงในรูปแบบของฟังก์ชันการแจกแจงความน่าจะเป็นแบบสะสม (Cumulative Distribution Function) ไว้ในภาพที่ 19 เพื่อพิจารณาจากภาพดังกล่าวจะพบว่าอัลกอริทึมสามารถระบุตำแหน่งได้ระยะทางผิดพลาดน้อยกว่าหรือเท่ากับ 1 เมตรเป็นจำนวนร้อยละ 74.39 และ 72.38 ในกรณีของ LG Nexus 5X และ Samsung Galaxy S5 ตามลำดับ ดังนั้นจากผลลัพธ์ดังกล่าวจึงสามารถแสดงให้เห็นว่าการพิจารณาลายนิ้วมือของสัญญาณวิทยุหลาย ๆ ลายนิ้วมือของอัลกอริทึม InHit นั้นสามารถลดผลกระทบจากปัญหาตำแหน่งที่ไม่ตรงกันได้



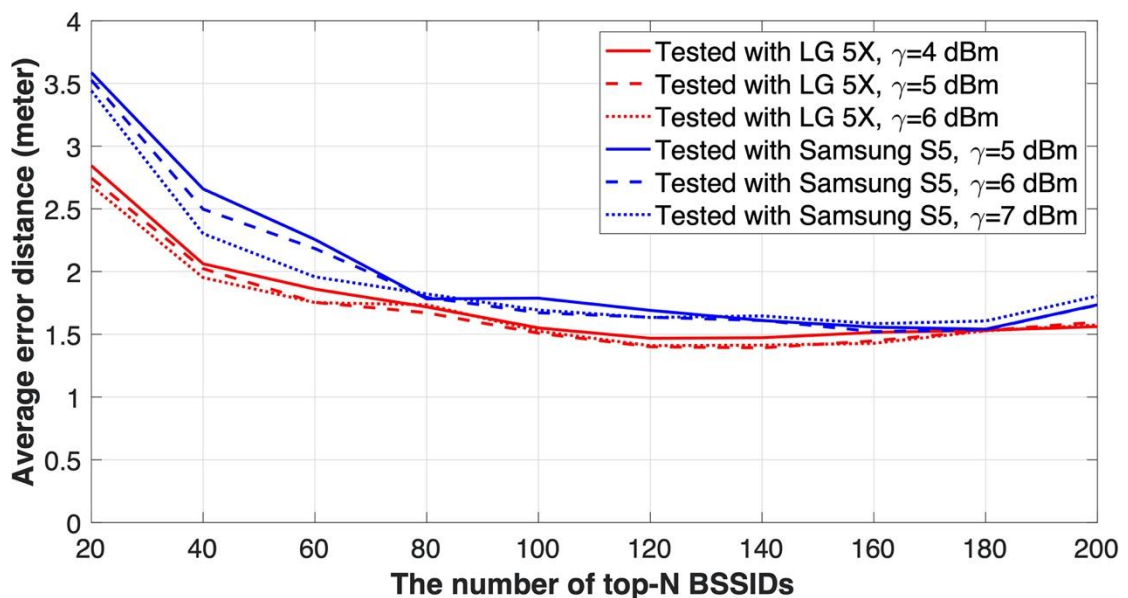
ภาพที่ 18 ระยะทางผิดพลาดเฉลี่ยของอัลกอริทึม InHit ในกรณีค่า k ต่าง ๆ



ภาพที่ 19 ฟังก์ชันการแจกแจงความน่าจะเป็นแบบสะสมของระยะทางผิดพลาด

เพื่อวัดผลกระทบจากความหลากหลายของโทรศัพท์มือถือ การทดลองนี้จึงใช้ข้อมูลจาก LG Nexus 5X ในการสร้างลายนิ้วมือของสัญญาณและตั้งค่า  $k$  เป็น 1 ส่วนการทดสอบจะใช้ข้อมูลจาก LG Nexus 5X อีกเครื่องหนึ่งและ Samsung Galaxy S5 ภาพที่ 20 ได้แสดงผลลัพธ์ความแม่นยำของการทดลองดังกล่าว ตารางที่ 13 ได้สรุประยะทางผิดพลาดเฉลี่ยน้อยสุดของแต่ละกรณีและค่าคงที่ที่ใช้ในกรณีนั้น ๆ เมื่อพิจารณาจากภาพดังกล่าว จะ

พบว่าระยะทางผิดพลาดเฉลี่ยของทั้งสองกรณีของโทรศัพท์มือถือนั้นมีค่าแตกต่างกันเพียงเล็กน้อยเมื่อใช้ค่าคงที่ที่เหมาะสม โดยในการทดลองนี้จะพิจารณาค่า top-160 และ  $\gamma$  เป็น 6 dBm เป็นค่าคงที่ที่เหมาะสม เนื่องจากเป็นค่าที่ทำให้เครื่อง Samsung Galaxy S5 ได้รับระยะทางผิดพลาดเฉลี่ยเหมาะสมน้อยสุด และทำให้เครื่อง LG Nexus 5X ได้ระยะทางผิดพลาดเฉลี่ยเหมาะสมใกล้เคียงกับค่าน้อยสุด ดังนั้นจึงจะใช้ค่าคงที่นี้ในการพิจารณาหาค่า  $k$  ที่เหมาะสมต่อไป



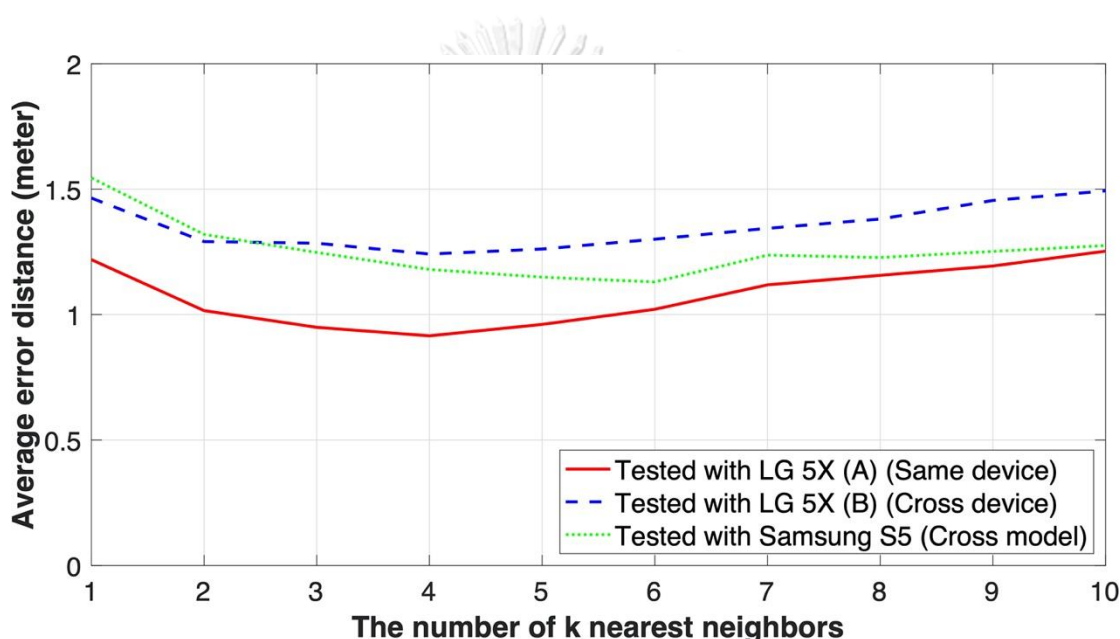
ภาพที่ 20 ระยะทางผิดพลาดเฉลี่ยของอัลกอริทึม InHit เมื่อสร้างลายนิ้วมือของสัญญาณและทดสอบด้วยเครื่องที่แตกต่างกัน

ตารางที่ 13 ระยะทางผิดพลาดเฉลี่ยน้อยสุดของอัลกอริทึม InHit เมื่อสร้างลายนิ้วมือของสัญญาณและทดสอบด้วยเครื่องที่แตกต่างกัน

ทดสอบด้วย	ระยะทางผิดพลาดเฉลี่ยน้อยสุด (ม.)	ค่าคงที่ที่ใช้
Nexus 5X	1.39	top-140, $\gamma = 5$
Samsung S5	1.1	top-160, $\gamma = 6$

ภาพที่ 21 ได้แสดงผลพธีในการทดลองเพื่อหาค่า  $k$  ที่เหมาะสมที่สุด สำหรับการใส่โทรศัพท์มือถือรุ่นต่าง ๆ เมื่อใช้ค่าคงที่ top-160 และ  $\gamma$  เป็น 6 dBm โดยในกรณีของการทดสอบด้วยโทรศัพท์เครื่องเดียวกันกับที่ใช้ในการสร้างลายนิ้วมือของสัญญาณนั้น ระยะทาง

ผิดพลาดเฉลี่ยสะสมน้อยที่สุดคือ 0.92 เมตร เมื่อใช้  $k$  เป็น 4, กรณีของการทดสอบด้วย โทรศัพท์ต่างเครื่องแต่รุ่นเดียวกันจะได้ระยะทางผิดพลาดเฉลี่ยน้อยที่สุดคือ 1.24 เมตรเมื่อใช้  $k$  เป็น 4 และกรณีของการทดสอบด้วยโทรศัพท์ต่างเครื่องและต่างรุ่นกันจะได้ระยะทางผิดพลาดเฉลี่ยน้อยที่สุดคือ 1.13 เมตร เมื่อใช้  $k$  เป็น 6 จากผลลัพธ์ดังกล่าวจะพบว่าระยะทางผิดพลาดเฉลี่ยในกรณีของการใช้โทรศัพท์ต่างเครื่องในการทดสอบนั้น ส่งผลกระทบต่อ ระยะทางผิดพลาดเฉลี่ยเพียงเล็กน้อย นอกจากนี้ค่าคงที่  $k$  ที่เหมาะสมที่สุดในแต่ละกรณีนั้นก็ใกล้เคียงกัน ดังนั้นจึงสามารถสรุปได้ว่าอัลกอริทึม InHit นั้นทนทานต่อความแตกต่างของ โทรศัพท์มือถือ



ภาพที่ 21 ระยะทางผิดพลาดเฉลี่ยของอัลกอริทึม InHit เมื่อสร้างลายนิ้วมือของสัญญาณและทดสอบ ด้วยเครื่องที่แตกต่างกัน ในกรณีค่า  $k$  ต่าง ๆ

นอกจากนี้ในงานวิจัยชิ้นนี้ ได้ทดสอบ InHit ในสถานการณ์ที่มีการเปลี่ยนแปลงของ สภาพแวดล้อมด้วย กล่าวคือ กรณีของการหายไปของบีเอสเอสไอดีและค่าความแรงสัญญาณ ที่แปรปรวน โดยจะใช้ชุดข้อมูลที่ 2 ที่ถูกเก็บด้วย Samsung Galaxy S5 สำหรับการจำลอง สถานการณ์ต่าง ๆ ดังกล่าว โดยในการจำลองนี้จะใช้ค่าคงที่ top-180,  $\gamma = 4$  และ  $k$  เป็น 5 เนื่องจากเป็นค่าคงที่ที่ให้ค่าระยะทางผิดพลาดเฉลี่ยน้อยที่สุด

ในกรณีของการหายไปของบีเอสเอสไอดีนั้น ได้เริ่มการทดสอบจากการลบบีเอสเอส ไอดีร้อยละ 10 ของจำนวนบีเอสเอสไอดีในข้อมูลทดสอบ หลังจากนั้นจึงเพิ่มขึ้นเป็นร้อยละ 70 ตารางที่ 14 ได้สรุประยะทางผิดพลาดเฉลี่ยของแต่ละกรณี เมื่อพิจารณาตารางดังกล่าว

ระยะทางผิดพลาดเฉลี่ยในแต่ละกรณีนั้นเพิ่มขึ้นเพียงเล็กน้อยเมื่อจำนวนปีเอสเอสไอตีที่หายไปเพิ่มขึ้น เนื่องจากการใช้ค่าคงที่ top-N ที่สูงเพียงพอสำหรับอัลกอริทึม จึงทำให้อัลกอริทึมจึงสามารถทนต่อการเปลี่ยนแปลงนี้ได้ ดังนั้นจึงสามารถสรุปได้ว่า InHit ทนทานต่อการหายไปของปีเอสเอสไอตี

ในกรณีของค่าความแรงสัญญาณที่แปรปรวน ได้เริ่มการทดลองด้วยการเลือกปีเอสเอสไอตีร้อยละ 25 ของจำนวนปีเอสเอสไอตีในข้อมูลทดสอบและลดพลังงานในการส่งสัญญาณของปีเอสเอสไอตีเหล่านั้นเป็นร้อยละ 90 ของพลังงานดั้งเดิม หลังจากนั้นจึงเพิ่มจำนวนปีเอสเอสไอตีเป็นร้อยละ 75 และลดพลังงานในการส่งเป็นร้อยละ 40 ตารางที่ 15 ได้สรุประยะทางผิดพลาดเฉลี่ยของการทดลองนี้ จากตารางดังกล่าว ระยะทางผิดพลาดเฉลี่ยในแต่ละกรณีนั้นลดลงเพียงเล็กน้อย โดยเหตุผลได้อธิบายไว้ในหัวข้อที่ 4.3.1 แล้ว จึงสามารถสรุปได้ว่าอัลกอริทึม InHit นั้นทนทานต่อความแปรปรวนของค่าความแรงสัญญาณ

ตารางที่ 14 ระยะทางผิดพลาดเฉลี่ยของอัลกอริทึม InHit ในสถานการณ์ที่ปีเอสเอสไอตีหายไป

ปีเอสเอสไอตีที่หายไป	0%	10%	20%	30%	40%	50%	60%	70%
ระยะทางผิดพลาดเฉลี่ย (ม.)	0.79	0.87	0.87	0.88	0.90	0.94	0.99	1.08

ตารางที่ 15 ระยะทางผิดพลาดเฉลี่ยของอัลกอริทึม InHit ในสถานการณ์ที่ค่าความแรงของสัญญาณแปรปรวน

ปีเอสเอสไอตีที่ถูกเลือก	พลังงานในการส่งที่เหลือ						
	100%	90%	80%	70%	60%	50%	40%
25%	0.79	0.85	0.85	0.85	0.84	0.86	0.86
50%	0.79	0.85	0.85	0.85	0.88	0.93	0.93
75%	0.79	0.85	0.85	0.88	0.97	1.13	1.13

#### 4.3.4) อัลกอริทึม MissingHit

ในการทดลองเพื่อหาความแม่นยำในการตรวจจับบีเอสเอสไอดีที่หายไปของอัลกอริทึม MissingHit จะใช้วิธีการจำลองที่ได้กล่าวไปแล้วในบทที่ 4.1.3 และจะใช้ชุดข้อมูลที่ 2 ที่ได้กล่าวถึงในบทที่ 4.1.2 โดยการทดลองจะเริ่มจากการลบบีเอสเอสไอดีร้อยละ 10 และเพิ่มขึ้นเป็นร้อยละ 70 ซึ่งจากผลการทดลองของอัลกอริทึม InHit ในบทที่ 4.3.3 นั้น อัลกอริทึมให้ความแม่นยำที่สูงทั้งในกรณีของสถานการณ์ปกติและในสถานการณ์ที่มีบีเอสเอสไอดีที่หายไปเมื่อใช้ค่าคงที่เป็น top-180,  $\gamma = 2$  สำหรับ LG Nexus 5X และ top-180,  $\gamma = 4$  สำหรับ Samsung Galaxy S5 ดังนั้นการทดลองนี้จึงจะใช้ค่าคงที่เหล่านี้เป็นข้อมูลขาเข้าสำหรับอัลกอริทึม

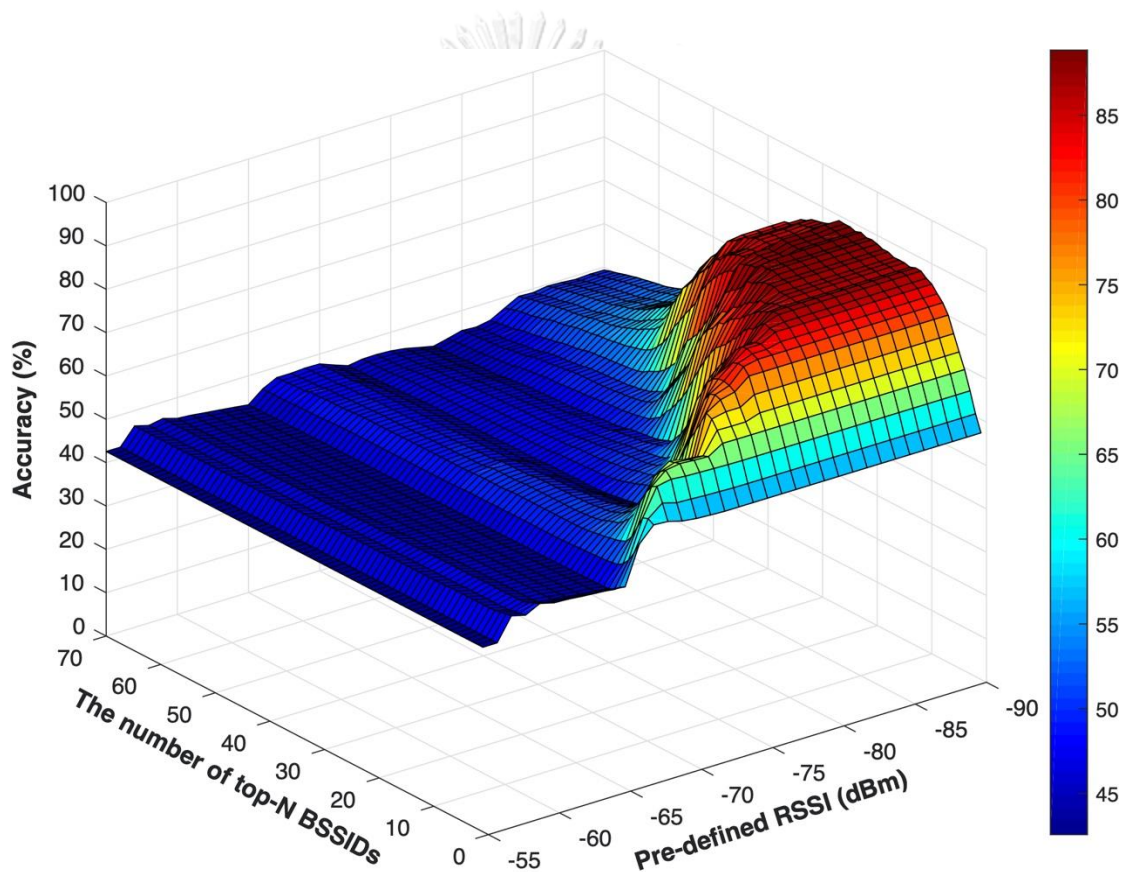
ผลลัพธ์ของอัตราการตรวจจับบีเอสเอสไอดีที่หายไปได้สรุปไว้ในตารางที่ 16 จากตารางดังกล่าว โมดูลนี้ให้ผลลัพธ์ที่สูงในอัตราการตรวจจับบีเอสเอสไอดีที่หายไปสำหรับทุก ๆ กรณี ดังนั้นโมดูลนี้สามารถใช้ในการตรวจจับบีเอสเอสไอดีที่หายไปได้อย่างมีประสิทธิภาพ ซึ่งระบบจะสามารถปรับปรุงฐานข้อมูลของลักษณะสัญญาณเพื่อรับมือกับปัญหาการเปลี่ยนแปลงไปของสภาพแวดล้อมได้

ตารางที่ 16 อัตราการตรวจจับบีเอสเอสไอดีที่หายไปของอัลกอริทึม MissingHit

บีเอสเอสไอดีที่หายไป	10%	20%	30%	40%	50%	60%	70%
อัตราการตรวจจับ (%)	ทดสอบด้วย LG Nexus 5X						
	98.80	98.98	98.55	98.55	98.52	98.30	98.36
	ทดสอบด้วย Samsung Galaxy S5						
	98.75	98.79	99.14	99.27	99.37	99.39	99.35

เนื่องจากโมดูลจำแนกในอาคาร/นอกอาคารนั้นไม่ทนทานต่อการหายไปของบีเอสเอสไอดี ดังนั้นงานวิจัยชิ้นนี้จึงใช้ MissingHit ในการปรับปรุงฐานข้อมูลลักษณะสัญญาณก่อนที่โมดูลจำแนกในอาคาร/นอกอาคารจะสร้างฐานข้อมูลลายนิ้วมือของสัญญาณ โดยจะใช้ชุดข้อมูลที่ 1 ซึ่งได้อธิบายไว้ในบทที่ 4.1.1 ที่ถูกเก็บด้วย LG Nexus 5X ในการฝึกและทดสอบสำหรับการทดลองนี้

ภาพที่ 22 ได้แสดงผลลัพธ์ความแม่นยำสำหรับโมดูลจำแนกในอาคาร/นอกอาคารที่ทำงานร่วมกับ MissingHit ในสถานการณ์ที่มีบีเอสเอสไอดีหายไปร้อยละ 50 ซึ่งความแม่นยำสูงสุดของโมดูลจำแนกในอาคาร/นอกอาคารคือร้อยละ 88.83 เมื่อใช้ top-27 และ *pre-defined RSSI* เป็น -89 dBm จะพบว่าความแม่นยำสูงสุดมีแนวโน้มที่จะใช้ top-N ที่น้อยลงเมื่อเทียบกับสถานการณ์ปกติในภาพที่ 11 ดังนั้นจึงสามารถเลือกค่าคงที่ล่วงหน้าที่เหมาะสมที่สุดสำหรับแต่ละสถานการณ์ด้วยการสังเกตจากผลลัพธ์จากการจำลองได้ ซึ่งในสถานการณ์จริงนั้น โมดูลจำแนกในอาคาร/นอกอาคารสามารถปรับเปลี่ยนค่าคงที่ตามจำนวนบีเอสเอสไอดีที่หายไปเพื่อให้ได้ผลลัพธ์ความแม่นยำที่สูงที่สุด

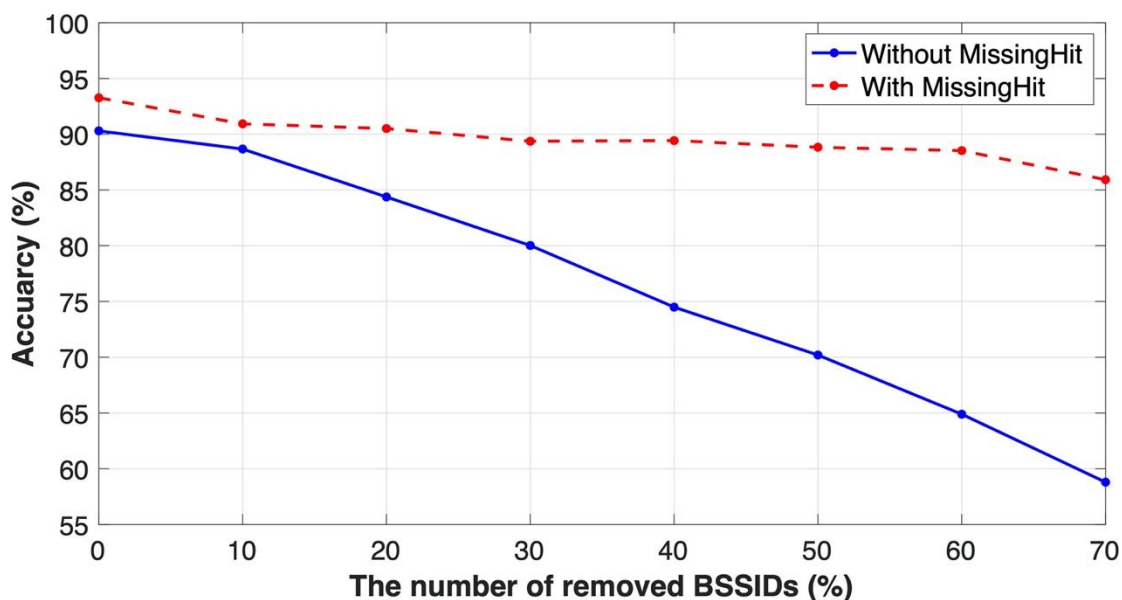


ภาพที่ 22 ความแม่นยำของโมดูลจำแนกในอาคาร/นอกอาคารที่ทำงานร่วมกับ MissingHit ในกรณีที่ มีบีเอสเอสไอดีหายไปร้อยละ 50

ภาพที่ 23 ได้เปรียบเทียบผลลัพธ์ความแม่นยำของโมดูลจำแนกในอาคาร/นอกอาคารที่ไม่ได้ทำงานร่วมกับ MissingHit และโมดูลจำแนกในอาคาร/นอกอาคารที่ทำงานร่วมกับ MissingHit เมื่อพิจารณาภาพดังกล่าว ในกรณีที่บีเอสเอสไอดีหายไปร้อยละ 70



ความแม่นยำเพิ่มขึ้นจากร้อยละ 58.79 เป็นร้อยละ 85.92 ดังนั้นจะเห็นได้ว่า MissingHit สามารถเพิ่มความแม่นยำของโมดูลจำแนกในอาคาร/นอกอาคารได้อย่างมาก



ภาพที่ 23 การเปรียบเทียบความแม่นยำระหว่างโมดูลจำแนกในอาคาร/นอกอาคารที่ไม่ได้ทำงานร่วมกับ MissingHit และโมดูลจำแนกในอาคาร/นอกอาคารที่ทำงานร่วมกับ MissingHit

เพื่อเพิ่มความแม่นยำของโมดูลจำแนกอาคารนั้น งานวิจัยชิ้นนี้จึงได้ใช้ MissingHit ในการปรับปรุงฐานข้อมูลลักษณะสัญญาณก่อนที่โมดูลจำแนกอาคารจะสร้างฐานข้อมูลลายนิ้วมือของสัญญาณ โดยจะใช้ชุดข้อมูลที่ 2 และ 3 ซึ่งได้อธิบายไว้ในบทที่ 4.1.1 ที่ถูกเก็บด้วย LG Nexus 5X สำหรับการทดลองนี้

ตารางที่ 17 ได้เปรียบเทียบความแม่นยำของโมดูลจำแนกอาคารที่ไม่ได้ทำงานร่วมกับ MissingHit และโมดูลจำแนกอาคารที่ทำงานร่วมกับ MissingHit จากตารางจะเห็นว่า MissingHit สามารถเพิ่มความแม่นยำให้กับโมดูลจำแนกอาคารได้ในทุก ๆ กรณี

โดยรวมแล้ว อัลกอริทึมต่าง ๆ จะมีความทนทานที่สูงขึ้นกับสภาพแวดล้อมที่เปลี่ยนไปเมื่อทำงานร่วมกับ MissingHit เนื่องจาก MissingHit จะปรับปรุงฐานข้อมูลลักษณะสัญญาณก่อนที่โมดูลจำแนกในอาคาร/นอกอาคาร และโมดูลจำแนกอาคารจะสร้างฐานข้อมูลลายนิ้วมือของสัญญาณของตัวเอง



ตารางที่ 17 การเปรียบเทียบความแม่นยำระหว่างโมดูลจำแนกอาคารที่ไม่ได้ทำงานร่วมกับ MissingHit และโมดูลจำแนกอาคารที่ทำงานร่วมกับ MissingHit

ปีเอสเอสไอดีที่หายไป	10%	20%	30%	40%	50%	60%	70%
ไม่ร่วมกับ MissingHit	97.39	97.14	97.02	96.88	96.53	96.10	95.22
ร่วมกับ MissingHit	99.85	99.80	99.67	99.43	99.07	98.74	98.27

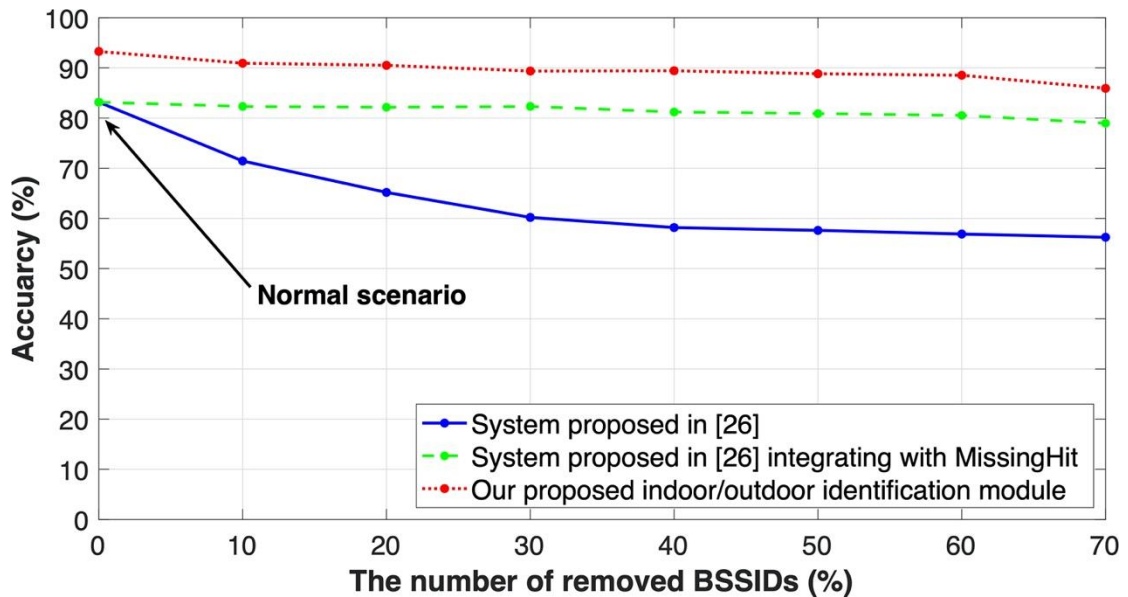
#### 4.3.5) การเปรียบเทียบกับระบบจำแนกพื้นที่อื่น ๆ

งานวิจัยชิ้นนี้ได้เปรียบเทียบประสิทธิภาพระหว่าง ExtHit กับระบบจำแนกพื้นที่ที่ถูกลำเสนอใน [26] ด้วย เนื่องจากระบบใน [26] ให้ความแม่นยำสูงและใช้สัญญาณวายฟายเป็นข้อมูลขาเข้า ระบบใน [26] ประกอบไปด้วยกันทั้งหมด 3 โมดูล ดังนี้ 1) โมดูลระบุใน/นอกพื้นที่ จะจำแนกคำร้องขอจากผู้ใช้งานว่าถูกส่งมาจากภายในอาคารหรือภายนอกอาคาร 2) โมดูลจำแนกพื้นที่ จะระบุอาคารที่ผู้ใช้งานส่งคำร้องขอเข้ามาที่ระบบ 3) โมดูลปรับปรุงผลลัพธ์การอ่านค่าสัญญาณของแต่ละอุปกรณ์ โดยการเปรียบเทียบจะทำการทดลองดังนี้

1) การทดลองเพื่อเปรียบเทียบระหว่างโมดูลจำแนกในอาคาร/นอกอาคารที่งานวิจัยชิ้นนี้นำเสนอและโมดูลระบุใน/นอกพื้นที่ที่ระบบใน [26] นำเสนอ โดยการทดลองนี้จะใช้ข้อมูลชุดที่ 1 จากตารางที่ 1 และเนื่องจากการวิเคราะห์องค์ประกอบหลัก (Principal Component Analysis: PCA) ที่ถูกรายงานใน [26] นั้นให้ประสิทธิภาพที่ดีที่สุด ดังนั้นจึงจะถูกนำมาใช้ในโมดูลระบุใน/นอกพื้นที่

นอกจากนี้ การทดลองนี้ได้วัดผลโมดูลดังกล่าวในสถานการณ์ที่ปีเอสเอสไอดีหายไป ด้วยการทำการจำลองที่ถูกกล่าวไว้ในบทที่ 4.1.3 เนื่องจาก MissingHit สามารถเพิ่มความแม่นยำในสภาพแวดล้อมที่เปลี่ยนไป จึงได้ประสานโมดูลทั้งสองเข้าด้วยกัน ภาพที่ 24 ได้แสดงผลการทดลองดังกล่าว ในสถานการณ์ปกติที่ไม่มีปีเอสเอสไอดีที่หายไป ระบบใน [26] ให้ผลลัพธ์ความแม่นยำที่ร้อยละ 83.17 ในขณะที่ระบบที่งานวิจัยชิ้นนี้นำเสนอให้ผลลัพธ์ความแม่นยำที่ร้อยละ 93.27 ยิ่งไปกว่านั้นความแม่นยำของระบบใน [26] นั้นลดลงอย่างมากในสถานการณ์ที่มีปีเอสเอสไอดีหายไป เนื่องจากปีเอสเอสไอดีที่หายไปนั้นส่งผลกับค่า ความผิดพลาดในการสร้างใหม่ (Reconstruction error) ที่ PCA ใช้ในการพิจารณาพื้นที่

ของคำร้องขอจากผู้ใช้งาน แต่อย่างไรก็ตามเมื่อประสานระบบใน [26] กับ MissingHit แล้ว ความแม่นยำของระบบดังกล่าวเพิ่มขึ้นอย่างมาก โดยเพิ่มขึ้นอยู่ที่ประมาณร้อยละ 23.64



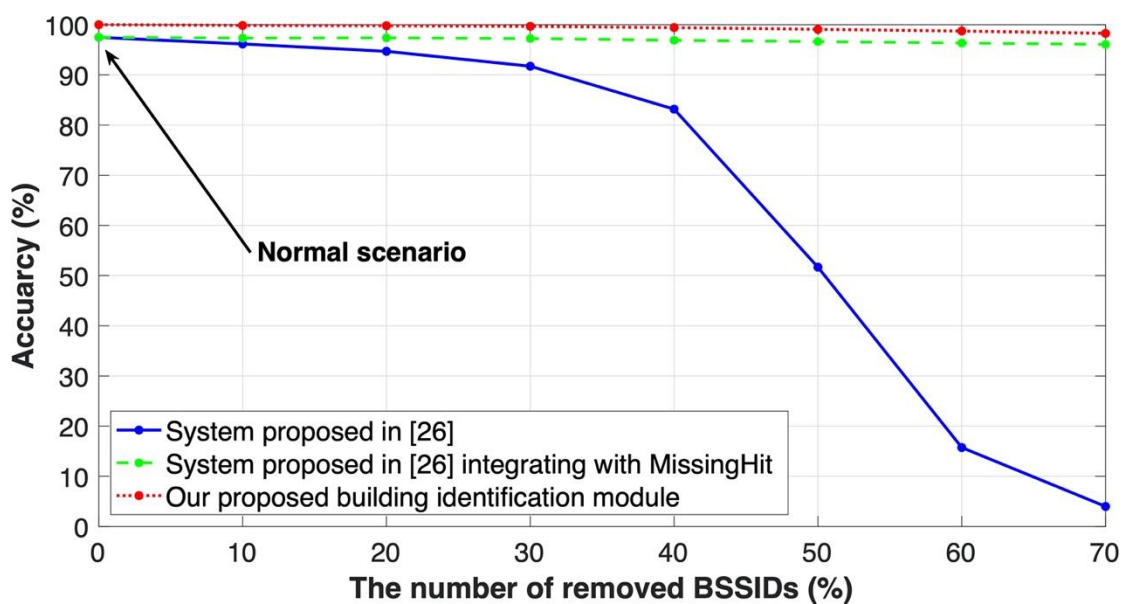
ภาพที่ 24 ความแม่นยำของโมดูลระบุใน/นอกพื้นที่ของระบบที่นำเสนอใน [26] ในสถานการณ์ที่มีบีเอสเอสไอดีหายไป

2) การทดลองเพื่อเปรียบเทียบระหว่างโมดูลจำแนกอาคารที่งานวิจัยชิ้นนี้นำเสนอ และโมดูลจำแนกพื้นที่ที่ระบบใน [26] นำเสนอ โดยการทดลองนี้จะใช้ข้อมูลชุดที่ 2 และ 3 จากตารางที่ 1 เป็นข้อมูลในการฝึกและการทดสอบตามลำดับ และเนื่องจาก Probabilistic Support Vector Machine (SVM) ที่ถูกรายงานใน [26] นั้นให้ประสิทธิภาพที่ดีที่สุด ดังนั้นจึงจะถูกนำมาใช้งานโมดูลจำแนกพื้นที่

นอกจากนี้ การทดลองนี้ได้วัดผลโมดูลดังกล่าวในสถานการณ์ที่บีเอสเอสไอดีหายไป ด้วย โดยการใช้การจำลองที่ถูกกล่าวไว้ในบทที่ 4.1.3 และได้ทดลองประสาน MissingHit เข้ากับโมดูลดังกล่าวเข้าด้วยกัน เพื่อเพิ่มความแม่นยำในสถานการณ์ที่สภาพแวดล้อมเปลี่ยนแปลงไป ภาพที่ 25 ได้แสดงผลผลความแม่นยำของการทดลองนี้ ในสถานการณ์ปกติที่ไม่มีบีเอสเอสไอดีที่หายไปนั้น ระบบใน [26] ได้ความแม่นยำร้อยละ 97.49 ในขณะที่ระบบที่งานวิจัยชิ้นนี้นำเสนอได้ความแม่นยำร้อยละ 100 ในสถานการณ์ที่มีบีเอสเอสไอดีที่หายไปมากกว่าร้อยละ 40 นั้น ความแม่นยำของระบบใน [26] ลดลงอย่างวิกฤต และเมื่อจำนวนบีเอสเอสไอดีที่หายไปร้อยละ 70 ความแม่นยำของระบบเหลือเพียงร้อยละ 3.99 เนื่องจากยังมีบีเอสเอสไอดีที่หายไปมากนั้น เวกเตอร์ก็จะยิ่งเบาบาง ซึ่งทำให้เกิดความไม่

แม่นยำในการจำแนกของ SVM แต่อย่างไรก็ตามเมื่อระบบดังกล่าวได้ประสานกับ MissingHit ความแม่นยำได้เพิ่มขึ้นอย่างมหาศาลจากร้อยละ 3.99 เป็นร้อยละ 96.08 จากผลการทดลองนี้แสดงให้เห็นว่า MissingHit นั้นเป็นประโยชน์มากในสภาพแวดล้อมที่เปลี่ยนไป

จากผลการทดลองทั้งสอง ExtHit ที่งานวิจัยชิ้นนี้นำเสนอสามารถทำงานได้ดีกว่าระบบที่นำเสนอใน [26] ยิ่งไปกว่านั้น MissingHit สามารถเพิ่มความแม่นยำให้กับระบบใน [26] ในสถานการณ์ที่สภาพแวดล้อมเปลี่ยนไป



ภาพที่ 25 ความแม่นยำของโมดูลจำแนกพื้นที่ของระบบที่นำเสนอใน [26] ในสถานการณ์ที่มีบีเอสเอสไอดีหายไป

#### 4.3.6) การเปรียบเทียบกับระบบระบุตำแหน่งภายในอาคารอื่น ๆ

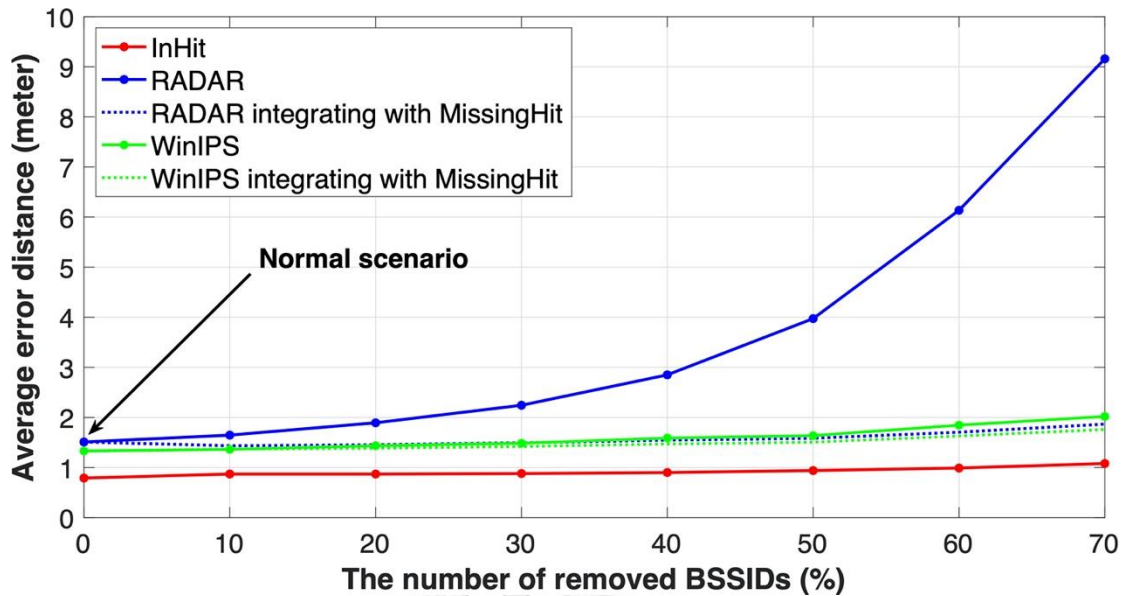
งานวิจัยชิ้นนี้ได้เปรียบเทียบ InHit กับระบบระบุตำแหน่งภายในอาคารที่ชื่อว่า RADAR [25] และ WinIPS [27] โดยจะใช้ชุดข้อมูลที่ 1 และ 2 ที่ได้อธิบายไว้ในบทที่ 4.1.2 ในการทดสอบความแม่นยำในการระบุชั้นและความแม่นยำในการระบุตำแหน่ง ตามลำดับ ซึ่งจะทดสอบระบบเหล่านี้ในกรณีที่มีบีเอสเอสไอดีที่หายไป ด้วยการทำการจำลองด้วย นอกจากนี้ในการทดลองจะประสาน MissingHit เพื่อเพิ่มความแม่นยำให้กับระบบดังกล่าว ตารางที่ 18 ได้แสดงผลลัพธ์ความแม่นยำ และภาพที่ 26 ได้แสดงผลลัพธ์ระยะทางผิดพลาดเฉลี่ยของการทดลองดังกล่าว

ในสถานการณ์ปกติที่ไม่มีการหายไปของบีเอสเอสไอตินั้น RADAR และ WinIPS ให้ความแม่นยำในการระบุชั้นที่ร้อยละ 100 RADAR และ WinIPS ได้ระยะทางผิดพลาดเฉลี่ยน้อยสุดสำหรับการระบุตำแหน่งที่ 1.51 เมตรและ 1.33 เมตร ตามลำดับ แต่อย่างไรก็ตาม InHit ให้ระยะทางผิดพลาดเฉลี่ยที่ 0.79 เมตร ซึ่งน้อยกว่า RADAR และ WinIPS

ในสถานการณ์ที่มีบีเอสเอสไอตีหายไป ความแม่นยำของ RADAR ลดลงอย่างมาก เนื่องจากบีเอสเอสไอตีที่หายไปส่งผลให้เวกเตอร์ของคำร้องขอจากผู้ใช้งานนั้นเบาบางลง ซึ่งส่งผลกับระยะทางยูคลิเดียนที่ RADAR ใช้ในการระบุตำแหน่ง ส่วนความแม่นยำของ WinIPS นั้นลดลงเล็กน้อยสำหรับการระบุตำแหน่งชั้นและตำแหน่งภายในอาคาร เนื่องจากกระบวนการปรับขนาดนั้นสามารถลดผลกระทบจากบีเอสเอสไอตีที่หายไปได้ เมื่อได้ทดลองประสาน MissingHit ในกรณีที่บีเอสเอสไอตีหายไปร้อยละ 70 แล้ว ความแม่นยำในการระบุชั้นของ RADAR เพิ่มขึ้นอย่างมหาศาลจากร้อยละ 66.69 เป็นร้อยละ 99.80 และระยะทางผิดพลาดเฉลี่ยลดลงอย่างมาก จาก 9.16 เมตรเป็น 1.87 เมตร โดยในกรณีของ WinIPS ความแม่นยำในการระบุชั้นเพิ่มขึ้นจากร้อยละ 99.89 เป็นร้อยละ 99.92 และระยะทางผิดพลาดเฉลี่ยลดลงจาก 2.02 เมตร เป็น 1.76 เมตร จากภาพรวมทั้งหมด MissingHit สามารถเพิ่มความแม่นยำให้กับระบบอื่น ๆ ได้ทั้งในการระบุชั้นและการระบุตำแหน่ง

ตารางที่ 18 ความแม่นยำในการระบุชั้นของ RADAR และ WinIPS เปรียบเทียบกับ InHit ในสถานการณ์ที่บีเอสเอสไอตีหายไป

บีเอสเอสไอตี ที่หายไป	0%	10%	20%	30%	40%	50%	60%	70%
<i>InHit</i>	100	100	100	100	100	100	100	100
RADAR	100	99.95	99.86	99.69	99	94.94	82.67	66.69
RADAR ร่วมกับ <i>MissingHit</i>	100	100	100	99.98	99.94	99.97	99.97	99.80
WinIPS	100	100	100	100	99.95	99.95	99.86	99.90
WinIPS ร่วมกับ <i>MissingHit</i>	100	100	100	100	99.98	99.97	99.97	99.92



ภาพที่ 26 ระยะทางผิดพลาดเฉลี่ยในการระบุตำแหน่งของ RADAR และ WinIPS เปรียบเทียบกับ InHit ในสถานการณ์ที่บีเอสเอสไอดีหายไป

#### 4.3.7) เวลาในการประมวลผลของระบบ

ในบทนี้จะวัดเวลาในการประมวลผลของสถาปัตยกรรมระบบระบุตำแหน่งภายในอาคารทั้งหมด 3 สถาปัตยกรรม ในการระบุตำแหน่งของคำร้องขอจากผู้ใช้งานจริง โดยแต่ละสถาปัตยกรรมมีรายละเอียดดังนี้

สถาปัตยกรรมที่ 1 จะมีเพียงอัลกอริทึม InHit เท่านั้น ซึ่งเป็นอัลกอริทึมสำหรับระบุตำแหน่งภายในอาคาร จะเปรียบเทียบแต่ละคำร้องขอจากผู้ใช้งานกับทุก ๆ ลายนิ้วมือของสัญญาณภายในฐานข้อมูล ดังนั้นระบบจะใช้ทรัพยากรและเวลาอย่างมหาศาลในการประมวลผลคำร้องขอจากผู้ใช้งาน

สถาปัตยกรรมที่ 2 จะใช้ ExtHit และ InHit ทำงานร่วมกัน ซึ่ง ExtHit จะทำการระบุว่าคำร้องขอจากผู้ใช้งานนั้นถูกส่งมาจากภายนอกอาคารหรือภายในอาคารใด หลังจากนั้น InHit จะใช้ข้อมูลนี้ในการลดปริมาณการค้นหา ดังนั้นระบบจะใช้ทรัพยากรและเวลาน้อยกว่าในการประมวลผลคำร้องขอจากผู้ใช้งาน เมื่อเทียบกับสถาปัตยกรรมที่ 1

สถาปัตยกรรมที่ 3 จะใช้ระบบใน [26] และ InHit ทำงานร่วมกัน เนื่องจากระบบใน [26] นั้นมีเพียงอัลกอริทึมจำแนกพื้นที่ ดังนั้นจึงได้ประสานเข้ากับ InHit ซึ่งเป็นอัลกอริทึมในการระบุตำแหน่ง ดังนั้นระบบใน [26] จะถูกใช้งานในการลดปริมาณการค้นหาให้กับ InHit

โดยในการทดลองเพื่อวัดเวลาในการประมวลผลของระบบนั้น จะใช้ชุดข้อมูลที่ 2 และ 5 ที่ได้อธิบายไว้ในบทที่ 4.1.1 ในการสร้างลายนิ้วมือของสัญญาณและใช้ทดสอบ

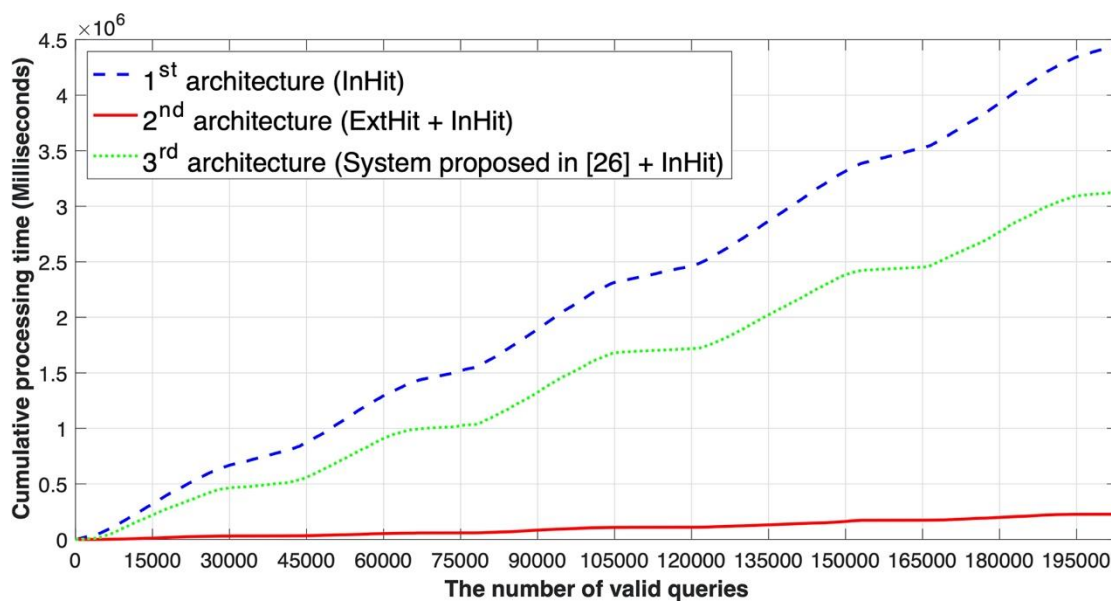
ตามลำดับ โดยชุดข้อมูลที่ 5 นั้นประกอบไปด้วยรายการคำร้องขอจากผู้ใช้งานทั้งหมด 609,277 รายการ ซึ่ง 405,969 รายการนั้นไม่สามารถใช้งานได้เพราะไม่มีบีเอสเอสไอดี เนื่องจากผู้ใช้งานไม่ได้เปิดใช้งานส่วนเชื่อมต่อวายฟาย ซึ่งในการทดลองนี้จะใช้ 203,308 รายการที่เหลือในการทดสอบ

เวลาประมวลผลสะสมสำหรับแต่ละสถาปัตยกรรมได้ถูกแสดงไว้ในภาพที่ 27 ตารางที่ 19 ได้สรุปเวลาในการประมวลผลสะสมและจำนวนคำร้องขอที่ระบบจำแนกกว่าเป็นคำร้องขอจากภายในอาคาร จากภาพดังกล่าวจะเห็นว่า การใช้ ExtHit ในการจำกัดปริมาณในการค้นหาให้ InHit นั้นสามารถลดเวลาในการประมวลผลได้อย่างมาก จาก 4,446 วินาที เป็น 227 วินาที แต่อย่างไรก็ตามผลลัพธ์ที่ได้แสดงให้เห็นว่าการใช้ระบบใน [26] นั้นไม่สามารถลดเวลาในการประมวลผลได้มากนัก เนื่องจากในพื้นที่ขนาดใหญ่จะมีตัวกระจายสัญญาณจำนวนมาก ทำให้มิติของข้อมูลมีขนาดใหญ่ ส่งผลให้กระบวนการสกัดลักษณะเฉพาะ (Feature extraction) ใช้เวลานาน

ยิ่งไปกว่านั้น จากตารางที่กล่าวไปข้างต้น จำนวนคำร้องขอที่สถาปัตยกรรมที่ 1 จำแนกกว่าเป็นภายในอาคารมีประมาณ 40,000 คำร้องขอ ซึ่งคำร้องขอส่วนใหญ่นั้นถูกส่งมาจากภายนอกอาคารแต่ InHit คืบคั้นค่าในอาคารกลับไปเป็นผลลัพธ์ ซึ่งการจำแนกที่ผิดพลาดนี้ทำให้เกิดการใช้งานทรัพยากรและเวลาอย่างเปล่าประโยชน์ ดังนั้นจึงแสดงให้เห็นว่าการใช้การจำแนกพื้นที่นั้นสำคัญอย่างมาก

ในภาพที่ 27 นั้นจะสังเกตเห็นได้ว่าความชันของเวลาประมวลผลสะสมของสถาปัตยกรรมทั้งหมดนั้นเปลี่ยนเป็นระยะ ๆ เช่น ความชันลดลงที่คำร้องขอที่ 30,000 และเพิ่มขึ้นที่คำร้องขอที่ 45,000 เนื่องจากช่วงที่มีความชันสูงนั้นเป็นคำร้องขอที่ถูกส่งมาจากเวลากลางวันทำงานนิทรรศการกำลังเปิด ส่วนช่วงที่มีความชันต่ำนั้นเป็นคำร้องขอที่ถูกส่งมาจากเวลากลางคืนที่งานนิทรรศการกำลังปิด ดังนั้นจำนวนบีเอสเอสไอดีในคำร้องขอจากผู้ใช้งานจึงขึ้นอยู่กับช่วงเวลา ในเวลากลางวัน จำนวนบีเอสเอสไอดีโดยเฉลี่ยในคำร้องขอนั้นมีอยู่ประมาณ 44 บีเอสเอสไอดี ในเวลากลางคืน จำนวนบีเอสเอสไอดีโดยเฉลี่ยในคำร้องขอนั้นมีอยู่เพียง 7 บีเอสเอสไอดี ดังนั้นเวลาในการประมวลผลของอัลกอริทึมจึงแตกต่างกันอย่างชัดเจนระหว่างสองช่วงเวลา

เพื่อที่จะสนับสนุนสมมติฐานดังกล่าว จึงได้แสดงภาพข้อมูลจีทีเอสของผู้ใช้งานโดยใช้แผนที่ความร้อน (Heat map) ภาพที่ 28 (a) และ ภาพที่ 28 (b) แสดงแผนที่ความร้อนของคำร้องขอจากผู้ใช้งานระหว่างเวลากลางคืนและกลางวัน ตามลำดับ

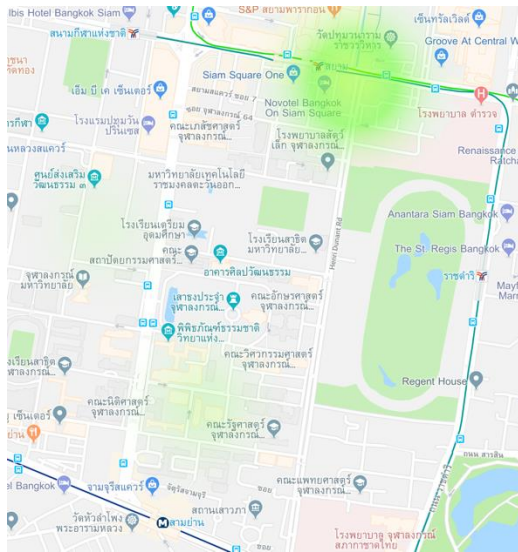


ภาพที่ 27 เวลาในการประมวลผลผลสะสมของแต่ละสถาปัตยกรรม

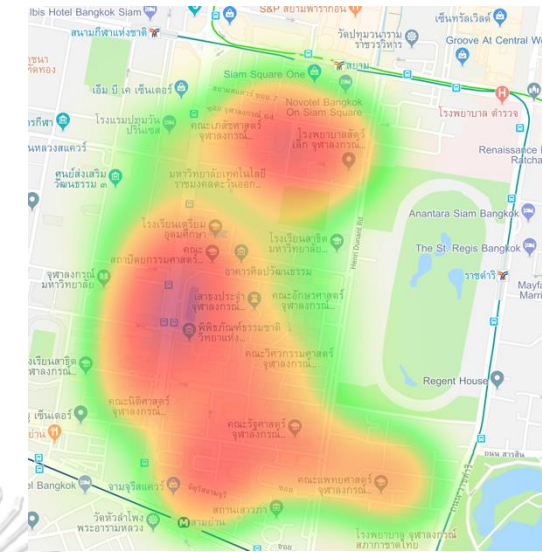
ตารางที่ 19 เวลาในการประมวลผลผลสะสมและผลลัพธ์การระบุตำแหน่งของแต่ละสถาปัตยกรรม

สถาปัตยกรรม	เวลาในการประมวลผล สะสม (วินาที)	จำนวนคำร้องขอที่ ระบบจำแนกว่าเป็นคำ ร้องขอจากในอาคาร
สถาปัตยกรรมที่ 1 (InHit)	4,446	39,573
สถาปัตยกรรมที่ 2 (ExtHit และ InHit)	227	27,438
สถาปัตยกรรมที่ 3 (ระบบใน [26] และ InHit)	3,135	12,634





(a) ช่วงเวลากลางคืน



(b) ช่วงเวลากลางวัน

ภาพที่ 28 แผนที่ความร้อนแสดงคำร้องขอจากผู้ใช้งานในช่วงเวลาที่ต่างกัน





## บทที่ 5 บทสรุปของงานวิจัยและอภิปรายผลการวิจัย

### 5.1 สรุปผลการวิจัย

ในงานวิจัยครั้งนี้ได้นำเสนอระบบระบุตำแหน่งภายในอาคารสำหรับสมาร์ทโฟนด้วยการใช้เทคนิคลายนิ้วมือของสัญญาณเวฟวิทยุเชิงกำหนด ระบบดังกล่าวประกอบไปด้วยทั้งหมด 3 ส่วน ดังนี้ 1) อัลกอริทึมสำหรับจัดหมวดหมู่พื้นที่ ชื่อว่า “ExtHit” อัลกอริทึมนี้ ประกอบไปด้วยทั้งหมด 3 โมดูล โดยโมดูลที่หนึ่งคือโมดูลกรองบีเอสเอสไอทีที่ไม่รู้จัก จะกรองบีเอสเอสไอทีที่ไม่รู้จักออกจากคำร้องขอจากผู้ใช้งาน โมดูลที่สองคือโมดูลจำแนกในอาคาร/นอกอาคาร จะจำแนกว่าผู้ใช้งานนั้นส่งคำร้องขอจากพื้นที่ภายในอาคารหรือภายนอกอาคาร ซึ่งได้ความแม่นยำสูงสุดที่ร้อยละ 93.27 โดยโมดูลนี้ไม่จำเป็นต้องใช้ลายนิ้วมือของสัญญาณจากภายนอกอาคาร ดังนั้นจึงไม่จำเป็นต้องใช้แรงงานในการเก็บลักษณะสัญญาณเพิ่ม และโมดูลที่สามคือโมดูลจำแนกอาคาร จะจำแนกว่าผู้ใช้งานอยู่ในอาคารใด ซึ่งสามารถให้ความแม่นยำสูงสุดที่ร้อยละ 100 โดยอัลกอริทึมนี้ทนทานต่อความหลากหลายของโทรศัพท์มือถือและสภาพแวดล้อมที่เปลี่ยนแปลงไป ยิ่งไปกว่านั้นอัลกอริทึม ExtHit นี้สามารถทำงานร่วมกับอัลกอริทึมระบุตำแหน่งภายในอาคารอื่น ๆ ที่ใช้เทคนิคลายนิ้วมือของสัญญาณได้ 2) อัลกอริทึมระบุตำแหน่งภายในอาคาร ชื่อว่า “InHit” ได้ผลลัพธ์ความแม่นยำสูงสุดในการระบุชั้นที่ร้อยละ 100 และได้ระยะทางผิดพลาดเฉลี่ยน้อยสุดที่ 0.79 เมตร 3) อัลกอริทึมตรวจจับบีเอสเอสไอทีที่หายไป ชื่อว่า “MissingHit” จะตรวจจับบีเอสเอสไอทีที่หายไปจากคำร้องขอจากผู้ใช้งานและปรับปรุงฐานข้อมูลลักษณะสัญญาณเพื่อเพิ่มความแม่นยำให้กับส่วนอื่น ๆ โดยอัตราการตรวจจับสูงสุดคือร้อยละ 99.39 เมื่อเปรียบเทียบกับงานวิจัยอื่นจะพบว่า ระบบที่นำเสนอสามารถให้ความแม่นยำทั้งในการจำแนกพื้นที่และการระบุตำแหน่งภายในอาคารสูงกว่างานวิจัยอื่น นอกจากนี้ผลลัพธ์ได้แสดงให้เห็นว่าการใช้ ExtHit ในการจำกัดปริมาณค้นหาสำหรับ InHit แล้ว สามารถลดเวลาในการประมวลผลลงมากกว่างานอื่นด้วยเช่นกัน เนื่องจากความซับซ้อนของระบบที่ต่ำกว่า ยิ่งไปกว่านั้น MissingHit สามารถทำงานร่วมกับ ExtHit และ InHit ได้อย่างมีประสิทธิภาพ และยังสามารถทำงานร่วมกับอัลกอริทึมจำแนกพื้นที่อื่นและอัลกอริทึมระบุตำแหน่งภายในอาคารอื่นเพื่อเพิ่มความแม่นยำได้ในสภาพแวดล้อมที่เปลี่ยนแปลงไป

### 5.2 ข้อเสนอแนะ

#### 5.2.1 การใช้ทรัพยากรของระบบ

สำหรับระบบระบุตำแหน่งภายในอาคารแล้ว การใช้ทรัพยากรของระบบเป็นเรื่องที่ต้องพิจารณาและให้ความสำคัญเช่นเดียวกับความแม่นยำในการระบุตำแหน่ง เนื่องจากการเพิ่มความแม่นยำในการระบุตำแหน่งเพียงเล็กน้อยแต่ต้องแลกมาด้วยการใช้ทรัพยากรที่เพิ่มขึ้นมหาศาลนับว่า

เป็นการใช้ทรัพยากรที่ไม่คุ้มค่า โดยการพิจารณาทรัพยากรที่ระบบต้องใช้ในการประมวลผล สามารถพิจารณาได้จากความซับซ้อนของอัลกอริทึม เนื่องจากถ้าอัลกอริทึมมีความซับซ้อนสูง ก็จะมีแนวโน้มที่จะต้องใช้เวลาในการประมวลผลมากตามไปด้วย และมีแนวโน้มที่จะใช้ทรัพยากรอื่นๆ สูงตาม เช่น พลังงานไฟฟ้าหรือหน่วยความจำ เป็นต้น

การเปรียบเทียบการใช้ทรัพยากรของระบบจะเปรียบเทียบแยกส่วนในแต่ละโมดูล โดยความซับซ้อนของแต่ละโมดูลได้สรุปไว้ในตารางที่ 20 ความซับซ้อนของอัลกอริทึมสำหรับแต่ละโมดูล โดย  $b$  คือจำนวนอาคารที่ระบบครอบคลุม  $N$  คือจำนวนปีเอสเอสไอดีที่ใช้เป็นข้อมูลขาเข้าของอัลกอริทึม  $r$  คือจำนวนลายนิ้วมือของสัญญาณภายในอาคารหนึ่งๆ และ  $U$  คือจำนวนปีเอสเอสไอดีทั้งหมดในฐานข้อมูลของลักษณะสัญญาณวายพาย ซึ่งจากการประมาณค่าของตัวแปรดังกล่าวโดยอ้างอิงจากข้อมูลที่ให้ทดสอบในงานวิจัยชิ้นนี้แล้ว จะได้  $b$  มีค่าเท่ากับ 37,  $N$  มีค่าเท่ากับ 80,  $n$  มีค่าเท่ากับ 100 และ  $U$  มีค่าเท่ากับ 11,800 เมื่อพิจารณาค่าของตัวแปรเหล่านี้แล้ว จะพบว่าระบบใน [26] มีความซับซ้อนสูงกว่าระบบที่นำเสนออย่างเห็นได้ชัด เนื่องจากค่า  $U$  มีปริมาณสูงกว่าค่า  $N$  หลายเท่าตัว นอกจากนี้ RADAR และ WinIPS มีความซับซ้อนที่สูงกว่าระบบที่นำเสนอเช่นกัน เนื่องจากความซับซ้อนของทั้งสองระบบดังกล่าวขึ้นกับค่า  $U$  แต่อย่างไรก็ตาม ถ้าหากพิจารณาในกรณีที่ใช้งานระบบให้ครอบคลุมพื้นที่ที่มีขนาดเล็กลง ซึ่งส่งผลให้ค่า  $U$  และค่า  $N$  มีแนวโน้มที่จะมีค่าใกล้เคียงกันและค่า  $b$  มีค่าเข้าใกล้ 1 ทำให้ความซับซ้อนแต่ละระบบแตกต่างกันน้อยลงและมีค่าใกล้เคียงกันในโมดูลระบุตำแหน่งภายในอาคาร

เมื่อพิจารณาในสถานการณ์ต่างๆ แล้วระบบที่นำเสนอจึงมีความคุ้มค่ามากกว่าระบบใน [26], RADAR และ WinIPS เนื่องจากมีความซับซ้อนที่ต่ำกว่า และผลลัพธ์ความแม่นยำที่ได้นำเสนอในบทที่ 4.3) ผลการทดลอง นั้นมีความแม่นยำที่สูงกว่า

ตารางที่ 20 ความซับซ้อนของอัลกอริทึมสำหรับแต่ละโมดูล

โมดูล	ระบบที่นำเสนอ	ระบบใน [26]	RADAR	WinIPS
จำแนกในอาคาร/ นอกอาคาร	$O(bN)$	$O(nU)$	-	-
จำแนกอาคาร	$O(bN)$	$O(bU)$	-	-
ระบุตำแหน่ง ภายในอาคาร	$O(rN^2)$	-	$O(rUn)$	$O(rUn)$

### 5.2.2) ความทนทานของระบบต่อสัญญาณรบกวน

ในบางสถานการณ์ เมื่อเปิดใช้งานระบบไปแล้วอาจมีสัญญาณรบกวนต่างๆ มารบกวนการทำงานของวายุพายได้จากหลายสาเหตุ เช่น การเปิดใช้งานตัวกระจายสัญญาณเคลื่อนที่ หรือการติดตั้งตัวกระจายสัญญาณใหม่ ซึ่งในสถานการณ์เหล่านี้ อาจทำให้ความแม่นยำของระบบลดลงได้ จึงจำเป็นที่จะต้องพิจารณาผลกระทบดังกล่าว โดยการรบกวนกันของสัญญาณข้างต้นเป็นสาเหตุของการทำให้ข้อมูลสูญหายเนื่องจากเกิดการชนกันของข้อมูล ซึ่งข้อมูลเหล่านั้นรวมไปถึงข้อมูลที่ตัวกระจายสัญญาณใช้ในการบ่งบอกถึงการมีตัวตนของตัวเองด้วย หรือที่เรียกว่า Beacon ดังนั้นเมื่อข้อมูลดังกล่าวหายไป ส่งผลให้ตัวรับสัญญาณไม่สามารถรับรู้ถึงการมีอยู่ของตัวกระจายสัญญาณนั้นๆ เปรียบเสมือนว่าตัวกระจายสัญญาณนั้นๆ หายไปจากระบบ

ถ้าหากพิจารณาในกรณีที่มีความยืดหยุ่นน้อยที่สุด คือระบบครอบคลุมวายุพายเฉพาะย่านความถี่ 2.4 GHz เท่านั้น โดยในความเป็นจริงระบบจะครอบคลุมวายุพายทั้งย่านความถี่ 2.4 GHz และ 5 GHz ซึ่งจะทำให้มีความยืดหยุ่นมากขึ้น นอกจากนี้ถ้าหากมีสัญญาณรบกวนเกิดขึ้นในช่องสัญญาณใด จะกำหนดให้ให้ช่องสัญญาณข้างเคียงอีกฝั่งละ 2 ช่องสัญญาณถูกรบกวนไปด้วย เนื่องจากความกว้างของช่องสัญญาณไปรบกวนช่องสัญญาณข้างเคียง กล่าวคือจะถูกรบกวนไปทั้งสิ้น 5 ช่องสัญญาณจากทั้งหมด 11 ช่องสัญญาณ คิดเป็นร้อยละ 45 โดยถ้าหากตัวกระจายสัญญาณกระจายทำงานอยู่ในช่องสัญญาณละเท่าๆ กัน จะสามารถเปรียบเทียบได้ว่ามีตัวกระจายสัญญาณหายไปจากระบบร้อยละ 45 ซึ่งเมื่อพิจารณาจากผลการทดลองในบทที่ 4.3) ผลการทดลอง นั้น การหายไปของตัวกระจายสัญญาณร้อยละ 50 นั้น อัลกอริทึมระบุตำแหน่งสามารถระบุตำแหน่งโดยมีความผิดพลาดเฉลี่ยที่ 0.94 เมตร ซึ่งเพิ่มขึ้นเพียง 0.15 เมตรจากกรณีปกติ

### 5.2.3) การใช้งานระบบในอาคารอื่นๆ

เนื่องจากระบบที่นำเสนอเป็นระบบที่ใช้แนวคิดของเทคนิคคลายนิ้วมือของสัญญาณ ที่จะใช้การจดจำลักษณะของสัญญาณวายุพายในแต่ละพื้นที่ และเปรียบเทียบลักษณะสัญญาณดังกล่าวในการระบุตำแหน่ง โดยการประยุกต์ใช้ระบบที่นำเสนอกับอาคารอื่นๆ ที่มีสภาพแวดล้อมที่แตกต่างกัน ให้ได้ความแม่นยำที่สูงนั้น จำเป็นที่จะต้องพิจารณาถึงความหนาแน่นของตัวกระจายสัญญาณ เนื่องจากความแม่นยำของระบบขึ้นอยู่กับจำนวนบีเอสเอสไอดีที่อัลกอริทึมใช้ในการเปรียบเทียบ เมื่อพิจารณาจากผลการทดลองในบทที่ 4.3) ผลการทดลอง แล้ว จะพบว่าจำนวนของบีเอสเอสไอดีที่เหมาะสมคือควรจะมีอย่างน้อย 40 บีเอสเอสไอดีขึ้นไป ระบบจึงจะสามารถระบุตำแหน่งด้วยความแม่นยำที่ดีที่สุดได้

#### 5.2.4) ขนาดของข้อมูลที่ถูกลง

ในการประยุกต์ใช้ระบบในการสถานการณ์จริง จำเป็นที่จะต้องพิจารณาถึงขนาดของข้อมูลที่ถูกลง เนื่องจากถ้าหากข้อมูลที่ถูกลงมีขนาดที่ใหญ่เกินไป จะทำให้ผู้ใช้งานมีโอกาสส่งข้อมูลแล้วสูญหายได้มาก ซึ่งจะทำให้เกิดความล่าช้าในการทำงาน โดยเมื่อพิจารณาถึงข้อมูลที่ระบบจะใช้จะประกอบไปด้วย 1) รายการของตัวกระจายสัญญาณที่สามารถอ่านค่าได้ ณ ตำแหน่งนั้น โดยจำนวนสูงสุดคือ 200 รายการ ซึ่งแต่ละรายการจะประกอบไปด้วยบีเอสเอสไอดี, เอสเอสไอดี, ค่าความแรงสัญญาณ และเวลาที่ตัวกระจายสัญญาณนั้นใช้ 2) รหัสของรุ่นโทรศัพท์ที่ใช้อ่านค่าสัญญาณ 3) ระยะเวลาที่อ่านค่าสัญญาณ ซึ่งเมื่อวัดขนาดของข้อมูลดังกล่าวแล้วสูงสุดจะมีค่าไม่เกิน 20 KB โดยเมื่อใช้งานกับความเร็วในการส่งข้อมูลปัจจุบัน จะใช้เวลาส่งเพียงเล็กน้อยเท่านั้น แต่อย่างไรก็ตามถ้าหากมีผู้ใช้งานมากจนเกิดการชนกันของข้อมูลแล้ว การประยุกต์ใช้วิธีการส่งข้อมูลที่มีความน่าเชื่อถือ (Reliable) ก็สามารถแก้ไขปัญหาคาการสูญหายของข้อมูลได้เช่นกัน

#### 5.2.5) ตำแหน่งอ้างอิงในการระบุตำแหน่ง

ในการเก็บข้อมูลเพื่อระบุตำแหน่งนั้น จะแบ่งพื้นที่ออกเป็นช่อง โดยแต่ละช่องจะมีขนาดเท่ากัน โดยตำแหน่งกึ่งกลางของแต่ละช่องคือพิกัดอ้างอิงสำหรับพื้นที่นั้น ๆ ในการเก็บข้อมูล ซึ่งลายนิ้วมือของสัญญาณที่ถูกสร้างจากข้อมูลที่ได้จากในช่องใด ก็จะมีตำแหน่งอ้างอิงเดียวกันกับช่องนั้นๆ โดยเมื่ออัลกอริทึมทำการระบุตำแหน่งเสร็จเรียบร้อยแล้ว จะคืนค่าพิกัดสุดท้ายเป็นผลลัพธ์จากการระบุตำแหน่งกลับไปหาผู้ใช้งาน โดยการวัดผลระยะทางผิดพลาดคือการวัดระยะทางระหว่างตำแหน่งที่เก็บข้อมูลจริงที่ใช้สำหรับการทดสอบ และตำแหน่งของผลลัพธ์จากการระบุตำแหน่งของอัลกอริทึม โดยการใช้ระยะทางยูคลิดีียนดังสมการที่ (21)

#### 5.2.6) แนวทางพัฒนาระบบในอนาคต

ปัจจุบันระบบสามารถตรวจจับบีเอสเอสไอดีที่หายไปและนำไปปรับปรุงฐานข้อมูลเพื่อลดผลกระทบจากการเปลี่ยนแปลงของสภาพแวดล้อมได้ แต่อย่างไรก็ตามในกรณีของบีเอสเอสไอดีที่ไม่รู้จัก ระบบจะกรองออกจากคำร้องขอจากผู้ใช้งานก่อนจะนำมาประมวลผล เพื่อลดผลกระทบจากบีเอสเอสไอดีดังกล่าว แต่หากบีเอสเอสไอดีที่ไม่รู้จักนั้น มาจากการติดตั้งตัวกระจายสัญญาณวายพายใหม่ในพื้นที่ ระบบควรที่จะเพิ่มลายนิ้วมือของสัญญาณใหม่เข้าไปในฐานข้อมูล แทนการกรองออกแบบในปัจจุบัน เนื่องจากเป็นบีเอสเอสไอดีที่มีแนวโน้มว่าจะอยู่อย่างถาวร ด้วยแนวคิดนี้จะสามารถทำให้ระบบสามารถปรับปรุงฐานข้อมูลได้เหมาะสมกับสถานการณ์มากขึ้น

## บรรณานุกรม

- [1] M. S. Bargh, and Robert de Groote, "Indoor localization based on response rate of bluetooth inquiries," *In Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, ACM, 2008.
- [2] R. Faragher, and Robert Harle, "Location fingerprinting with bluetooth low energy beacons.," *IEEE journal on Selected Areas in Communications* 33, no. 11 (2015).
- [3] Y. Gu, and Fuji Ren, "Energy-efficient indoor localization of smart hand-held devices using Bluetooth," *IEEE Access* 3, 2015.
- [4] G. C. C. R. Giuliano, C. Cesarini, L. D. Nunzio, F. Fallucchi, R. Fazzolari, F. Mazzenga, M. Re, and A. Vizzarri, "Indoor localization system based on Bluetooth low energy for museum applications," *Electronics*, vol. 9, 2020.
- [5] S. Holm, "Hybrid ultrasound-RFID indoor positioning: Combining the best of both worlds.," *In 2009 IEEE International Conference on RFID*, IEEE 2009.
- [6] Y. B. Bai, Suqin Wu, Hong Ren Wu, and Kefei Zhang, "Overview of RFID-Based Indoor Positioning Technology," *In GSR*, 2012.
- [7] A. A. A. M. El-Absi, A. Abuelhaija, F. Zheng, K. Solbach, and T. Kaiser, "High-accuracy indoor localization based on chipless RFID systems at THz band," *IEEE Access*, vol. 6, p. 14.
- [8] B. W. Y. Ma, S. Pei, Y. Zhang, S. Zhang, and J. Yu, "An indoor localization method based on AOA and PDOA using virtual stations in multipath and NLOS environments for passive UHF RFID," *IEEE Access*, vol. 6, p. 11, 2018.
- [9] S. A.-C. E. Hatem, E. Colin, J.-M. Laheurte, and B. El-Hassan, "Performance, accuracy and generalization capability of RFID tags' constellation for indoor localization," *Sensors*, vol. 20, 2020.
- [10] K. Liu, Xinxin Liu, and Xiaolin Li, "Guoguo: Enabling fine-grained indoor localization via smartphone," *In Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, 2013.
- [11] K. Khaoampai, Kulit Na Nakorn, and Kultida Rojviboonchai, "FloorLoc-SL: floor localization system with fingerprint self-learning mechanism," *International*

*Journal of Distributed Sensor Networks* 11, 2015.

- [12] G. T. Ye H, Zhu X, Xu J, Tao X, Lu J, Jin N, "FTrack: Infrastructure-free floor localization via mobile phone sensing," *In Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on 2012*, 2012.
- [13] S. Beauregard, and Harald Haas, "Pedestrian dead reckoning: A basis for personal positioning," *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*, 2006.
- [14] M. Zhang, Yingyou Wen, Jian Chen, Xiaotao Yang, Rui Gao, and Hong Zhao, "Pedestrian dead-reckoning indoor localization based on OS-ELM," *IEEE Access* 6, 2018.
- [15] M. Zhang, Ling Pei, and Xiaotie Deng, "GraphSLAM-based crowdsourcing framework for indoor Wi-Fi fingerprinting," *2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS)*, 2016.
- [16] R. Górak, Marcin Luckner, Michał Okulewicz, Joanna Porter-Sobieraj, and Piotr Wawrzyniak, "Indoor localisation based on GSM signals: Multistorey building study," *Mobile Information Systems 2016*, 2016.
- [17] A. Varshavsky, Anthony LaMarca, Jeffrey Hightower, and Eyal De Lara, "The skyloc floor localization system," *In Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'07)*, 2007.
- [18] J. Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, and Micaela Wiseman, "Indoor location sensing using geo-magnetism," *In Proceedings of the 9th international conference on Mobile systems, applications, and services*, 2011.
- [19] K. P. Subbu, Brandon Gozick, and Ram Dantu, "LocateMe: Magnetic-fields-based indoor localization using smartphones," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2013.
- [20] T. Vongsuteera, Kulit Na Nakorn, and Kultida Rojviboonchai, "Floor localization algorithm utilizing different order of access point from Wi-Fi signal fingerprint," *In 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2016.

- [21] S. Yang, et al, "FreeLoc: Calibration-free crowdsourced indoor localization," *2013 Proceedings IEEE INFOCOM*, 2013.
- [22] M. Youssef, and Ashok Agrawala, "The Horus WLAN location determination system," *In Proceedings of the 3rd international conference on Mobile systems, applications, and services*, 2005.
- [23] K. Khaoampai, Kulit Na Nakorn, and Kultida Rojviboonchai, "Low complexity floor localization algorithm for mobile phone," *In 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2014.
- [24] J. Zhang, Guangjie Han, Ning Sun, and Lei Shu, "Path-loss-based fingerprint localization approach for location-based services in indoor environments," *IEEE Access* 5, 2017.
- [25] P. Bahl, et al, "RADAR: An in-building RF-based user location and tracking system," 2000.
- [26] S. He, Jiajie Tan, and S-H. Gary Chan, "Towards area classification for large-scale fingerprint-based system," *In Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*, 2016.
- [27] H. Zou, et al., "WinIPS: WiFi-based non-intrusive indoor positioning system with online radio map construction and adaptation," *IEEE Transactions on Wireless Communications* 16.12 (2017), 2017.
- [28] F. Zafari, Athanasios Gkelias, and Kin Leung, "A survey of indoor localization systems and technologies," *arXiv preprint arXiv:1709.01015*, 2017.
- [29] W. Q. X. Zhu, T. Qiu, L. Zhao, M. Atiquzzaman, and D. O. Wu, "Indoor intelligent fingerprint-based localization: Principles, approaches and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, p. 24, 2020.
- [30] X. H. F. Gu, M. Ramezani, D. Acharya, K. Khoshelham, S. Valaee, and J. Shang, "Indoor localization improved by spatial context - A survey," *ACM Comput. Surv.*, vol. 52, p. 35, 2019.
- [31] S. S. a. P. Spachos, "RSSI-based indoor localization with the Internet of Things," *IEEE Access*, vol. 6, p. 13, 2018.
- [32] B. Wang, et al., "Indoor smartphone localization via fingerprint crowdsourcing:

- Challenges and approaches," *IEEE Wireless Communications* 23.3, 2016.
- [33] A. Khalajmehrabadi, Nikolaos Gatsis, and David Akopian, "Modern WLAN fingerprinting indoor positioning methods and deployment challenges," *IEEE Communications Surveys & Tutorials* 19.3, 2017.
- [34] L. Pei, et al., "A survey of crowd sensing opportunistic signals for indoor localization," *Mobile Information Systems* 2016, 2016.
- [35] S. He, and S-H. Gary Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Communications Surveys & Tutorials* 18.1, 2015.
- [36] T. V. a. K. Rojviboonchai, "Adaptive Indoor Localization System for Large-Scale Area," *IEEE Access*, vol. 9, p. 19, 2021.
- [37] P. Kumar, Lohith Reddy, and Shirshu Varma, "Distance measurement and error estimation scheme for RSSI based localization in Wireless Sensor Networks," *Fifth international conference on wireless communication and sensor networks (WCSN). IEEE*, 2009.
- [38] P. Zhou, Yuanqing Zheng, Zhenjiang Li, Mo Li, and Guobin Shen, "Iodetector: A generic service for indoor outdoor detection," *In Proceedings of the 10th acm conference on embedded network sensor systems*, 2012.
- [39] V. Radu, Panagiota Katsikouli, Rik Sarkar, and Mahesh K. Marina, "A semi-supervised learning approach for robust indoor-outdoor detection with smartphones," *In Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, 2014.
- [40] M. Azizyan, Ionut Constandache, and Romit Roy Choudhury, "SurroundSense: mobile phone localization via ambience fingerprinting," *In Proceedings of the 15th annual international conference on Mobile computing and networking*, 2009.





จุฬาลงกรณ์มหาวิทยาลัย  
**CHULALONGKORN UNIVERSITY**

## ประวัติผู้เขียน

ชื่อ-สกุล	นายธีรภัทร์ วงศ์สุธีรา
วัน เดือน ปี เกิด	14 กุมภาพันธ์ 2537
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษา	ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ที่อยู่ปัจจุบัน	256/62 ซ.ประชาสงเคราะห์ 47 เขตดินแดง แขวงดินแดง กรุงเทพมหานคร 10400
ผลงานตีพิมพ์	Adaptive Indoor Localization System for Large-Scale Area ตีพิมพ์ใน IEEE Access.;9:8847-65
รางวัลที่ได้รับ	Best paper award จากงานวิจัยชื่อ Floor localization algorithm utilizing different order of access point from Wi-Fi signal fingerprint ตีพิมพ์ใน 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) 2016 Jun 28 (pp. 1-6). IEEE.