ระบบตรวจจับโดรนด้วยไวไฟฟิงเกอร์พริ้นท์

Drone Detection by WiFi Fingerprint

พงษ์ดนัย พระสว่าง
สุริยวุฒิ สุริยาวงค์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

ระบบตรวจจับโดรนด้วยไวไฟฟิงเกอร์พริ้นท์

Drone Detection by WiFi Fingerprint

พงษ์ดนัย พระสว่าง
สุริยวุฒิ สุริยาวงค์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2561

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจจับโดรนด้วยไวไฟฟิงเกอร์พริ้นท์

DRONE DETECTION BY WIFI FINGERPRINT

ผู้จัดทำ

1. นายพงษ์ดนัย พระสว่าง รหัสนักศึกษา 58010819
2. นายสุริยวุฒิ สุริยาวงค์ รหัสนักศึกษา 58011369

_____ อาจารย์ที่ปรึกษา

(ผศ.ดร.สุรินทร์ กิตติธรกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจจับโดรนด้วยไวไฟฟิงเกอร์พริ้นท์

นายพงษ์ดนัย พระสว่าง 58010819
นายสุริยวุฒิ สุริยาวงค์ 58011369
ผศ.ดร.สุรินทร์ กิตติธรกุล อาจารย์ที่ปรึกษา
ปีการศึกษา 2561

บทคัดย่อ

โดรนในปัจจุบันมีความนิยมสูงมากขึ้น เนื่องจากมีขนาดเล็ก ราคาถูก สามารถเข้าถึงและหาซื้อได้ง่าย อีกทั้งมีการพัฒนาให้มีการควบคุมที่ง่ายมากขึ้น เพียงผู้ใช้มีสมาร์ทโฟนก็สามารถควบคุมการบินของโดรนได้ผ่านคลื่นสัญญาณไวไฟ จากที่กล่าวมาอาจทำให้เกิดปัญหาในการป้องกันการบินบุกรุกเข้าไปในพื้นที่ส่วนบุคคลาหรือสถานที่ที่มีความสำคัญ ซึ่งอาจเป็นภัยต่อบุคคลหรือสาธารณะได้ ในบทความนี้เราจึงเสนอวิธีตรวจจับโดรนที่อยู่ในบริเวณใกล้เคียงโดยใช้สัญญาณไวไฟ ซึ่งจะทำการวิเคราะห์ความเป็นเอกลักษณ์ของโดรนที่ส่งและรับข้อมูลกับอุปกรณ์ควบคุมบนเครือข่ายไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Drone Detection by WiFi Fingerprint

Mr. Pongdanai	Prasawang	58010819
Mr. Suriyawut	Suriyawong	58011369
Asst. Prof. Dr. Surin	Kittitornkun	Advisor

Academic Year 2561

ABSTRACT

Nowadays, demands for drone rise dramatically because they are small and low-price. Also, drones have been developed for ease of human controls. Users can remotely control the flights through wifi signal with only a smartphone. From the above mentioned, it may cause problems in preventing flight, invading into areas of people or places that are important. Drones may be a threat to a person or public. In this project, we present a method of detecting nearby drones based on wifi signals. The method analyzes the fingerprint of a drone that transmits/receives data to/from the remote control device.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงด้วยดีได้ด้วยความกรุณาจาก ผศ.ดร.สุรินทร์ กิตติธรกุล อาจารย์ที่ปรึกษาของโครงงานนี้ ที่ช่วยแนะนำแนวทาง ให้ความรู้ ตรวจสอบและแก้ไขข้อบกพร่องของรายงานให้สมบูรณ์ยิ่งขึ้น ซึ่งทำให้การทำงานต่าง ๆ ราบลื่นและเป็นไปด้วยดี

ขอขอบพระคุณอาจารย์และบุคลากรต่าง ๆ ในสาขาวิศวกรรมคอมพิวเตอร์ที่สั่งสอนความรู้ ให้คำแนะนำและช่วยเหลือในการทำโครงงานนี้ ขอขอบคุณรุ่นพี่และเพื่อน ๆ ในสาขาวิศวกรรมคอมพิวเตอร์ทุกคนที่ให้คำปรึกษาและคอยแบ่งปันความรู้ซึ่งกันและกัน

สุดท้ายนี้ขอขอบพระคุณ บิดา มารดา และครอบครัว ที่ได้เลี้ยงดู อบรมสั่งสอน ให้กำลังใจและรับฟังปัญหาต่าง ๆ พร้อมทั้งให้การสนับสนุนในด้านต่าง ๆ เสมอมา

พงษ์ดนัย พระสว่าง

สุริยวุฒิ สุริยาวงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาของปัญหา 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ประโยชน์ที่คาดว่าจะได้รับ 2

1.4 ขอบเขตของโครงงาน 2

1.5 วิธีการดำเนินงาน 2

1.6 ข้อจำกัดโครงงาน 2

1.7 ส่วนประกอบของปริญญานิพนธ์ 3

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 4

2.1 เทคโนโลยี UAV 4

2.2 Raspberry Pi 4

2.3 มาตรฐานระบบเครือข่ายไร้สาย (IEEE 802.11) 6

2.4 Machine Learning 7

2.5 Scikit Learn 8

2.6 อัลกอริทึม Random Forest 9

2.7 Web Framework 9

2.8 งานวิจัยที่เกี่ยวข้อง 10

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและพัฒนา 12

3.1 ภาพรวมโครงสร้างของระบบ 12

3.2 User Requirement 12

3.3 Flowchart ภาพรวมการทำงานของระบบ 13

3.4 การพัฒนาโปรแกรมตรวจจับโดรน 14

บทที่ 4 การทดลองและผลการดำเนินงาน 24

4.1 การทดลอง 24

4.2 สรุปผลการทดลอง 30

บทที่ 5 บทสรุปและข้อเสนอแนะ 31

5.1 บทสรุป 31

5.2 ปัญหาและอุปสรรค 31

5.3 แนวทางการพัฒนาต่อ 31

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง **หน้า**

ตาราง 2.1 มาตรฐานของระบบเครือข่ายไร้สาย 6
ตาราง 4.1 ค่า confusion matrix การตรวจจับโดรนในระยะใกล้ (0 - 10 เมตร) 26
ตาราง 4.2 ค่า confusion matrix การตรวจจับโดรนในระยะกลาง (20 - 30 เมตร) 26
ตาราง 4.3 ค่า confusion matrix การตรวจจับโดรนในระยะไกล (40 - 50 เมตร) 26
ตาราง 4.4 การเปรียบเทียบโมเดลตรวจจับโดรนในระยะทางต่าง ๆ 27
ตาราง 4.5 แสดงผลลัพธ์ที่ได้จากการตรวจจับ 27

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป **หน้า**

รูป 2.1 โดรนที่พบเห็นได้ในปัจจุบัน(ซ้าย) และ โดรนทางการทหาร(ขวา) 4
รูป 2.2 Raspberry Pi 3 Model B 5
รูป 2.3 MAC Frame 6
รูป 2.4 Frame Control 7
รูป 2.5 ภาพรวมของ Machine Learning 7
รูป 2.6 สัญลักษณ์ของ Scikit Learn 8
รูป 2.7 หลักการทำงานของ Random Forest 9
รูป 2.8 สัญลักษณ์ของ Flask 9
รูป 2.9 สัญลักษณ์ของ Bootstrap 10
รูป 3.1 System Diagram 12
รูป 3.2 Flow Chart การทำงานของระบบ 13
รูป 3.3 ขั้นตอนการพัฒนาโปรแกรมตรวจจับโดรน 14
รูป 3.4 ขั้นตอนการดักจับข้อมูล 14
รูป 3.5 Flowchart การดักจับข้อมูล 15
รูป 3.6 ตัวอย่างข้อมูลดิบ 16
รูป 3.7 ตัวอย่างข้อมูลที่ทำ Preprocess 16
รูป 3.8 โครงสร้างข้อมูลการสื่อสารระหว่าง AP กับ STA 16
รูป 3.9 ตัวอย่างข้อมูล Feature Vector 19
รูป 3.10 ตัวอย่าง fingerprint ของโดรนในระยะ 1 เมตร 20
รูป 3.11 ตัวอย่าง fingerprint ของโดรนในระยะ 4 เมตร 20
รูป 3.12 ตัวอย่าง fingerprint ของโดรนในระยะ 6 เมตร 20
รูป 3.13 ตัวอย่าง fingerprint ของโดรนในระยะ 10 เมตร 20
รูป 3.14 ตัวอย่าง fingerprint ของโดรนในระยะ 14 เมตร 20
รูป 3.15 ขั้นตอนการสร้างโมเดลในการตรวจจับโดรน 20
รูป 3.16 หน้าต่างเริ่มต้นของระบบ 21
รูป 3.17 หน้าต่างเมื่อคลิกปุ่ม Start Detect 22
รูป 3.18 หน้าต่างแสดงผลเมื่อตรวจพบเจอโดรน 22
รูป 3.19 หน้าต่างเมื่อคลิกปุ่ม Stop Detect 23

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป **หน้า**

รูป 4.1 แสดงการใช้คำสั่ง iw dev 24
รูป 4.2 แสดงขั้นตอนการเปิดใช้งาน Monitor Mode 24
รูป 4.3 การใช้คำสั่ง tcpdump 25
รูป 4.4 แสดงข้อมูลแพ็คเก็ตของโดรนที่จับมาได้ 25
รูป 4.5 ตัวอย่างข้อมูลจริงที่ดักจับมาได้ 28
รูป 4.6 คอมพิวเตอร์เชื่อมต่อกับ Raspberry Pi 3 28
รูป 4.7 โดรนที่ใช้ในการทดลอง Skyhunter X8 29
รูป 4.8 รูปแจ้งเตือนเมื่อมีโดรนบินเข้ามา 29
รูป 4.9 แสดงระยะทางที่สามารถตรวจจับได้ 30

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

อากาศยานไร้คนขับ หรือ UAV (Unmanned Aerial Vehicle) หรือที่ในปัจจุบันรู้จักและเรียกกันในนามว่า " โดรน " โดรนเป็นเทคโนโลยีอากาศยานที่สามารถควบคุมได้จากระยะไกลอัตโนมัติ ในอดีตถูกพัฒนามาเพื่อใช้ในทางการทหาร เพราะช่วยลดความเสี่ยงต่อนักบินที่ไม่จำเป็นต้องลงไปปฏิบัติการในพื้นที่จริงได้ โดรนได้ถูกพัฒนาขึ้นมาอย่างต่อเนื่องเพื่อใช้สำหรับ สอดแนม หรือ ส่งพัสดุ และอีกมากมายซึ่งในปัจจุบันโดรนไม่ได้ถูกพัฒนาเพื่อสำหรับทางการทหารเพียงอย่างเดียว โดรนนั้นได้ถูกพัฒนาเพื่อใช้ในภาคพลเรือน เพื่อใช้สำหรับสื่อบันเทิง เช่น งานวีดีทัศน์สื่อภาพยนตร์ หรือเป็นอุปกรณ์สำหรับการละเล่นที่พบเจอได้ทั่วไป ทำให้ในปัจจุบันมีการใช้โดรนกันอย่างแพร่หลาย เพราะโดรนมีราคาถูกและสามารถใช้งานได้ง่ายขึ้น ในปัจจุบันการควบคุมโดรนถูกพัฒนาให้ควบคุมผ่านสมาร์ทโฟน โดยใช้สัญญาณไวไฟได้ ซึ่งจากการที่มีการใช้อย่างแพร่หลายนี้อาจก่อให้เกิดปัญหากับหน่วยงานที่ต้องการความปลอดภัยสูง สถานที่ส่วนบุคคล หรืออาจเกิดปัญหาการละเมิดสิทธิส่วนบุคคลได้

จากที่กล่าวมาข้างต้น ทำให้มีการป้องกันและตรวจจับโดรนที่บินลุกล้ำเข้ามาในพื้นที่ โดยในปัจจุบันมีระบบตรวจจับที่ช่วยในการจำแนกและตรวจจับโดรนโดยใช้วิธีการต่าง ๆ มากมาย ตัวอย่างเช่น การตรวจจับด้วยภาพถ่ายจากกล้องที่บันทึกได้ ซึ่งการตรวจจับด้วยวิธีนี้อาจมีความแม่นยำลดลงในเวลากลางคืน การพัฒนาระบบตรวจจับในโครงงานนี้จึงจะเป็นส่วนที่ช่วยเพิ่มประสิทธิภาพในการตรวจจับโดรนในเวลากลางคืนให้แม่นยำยิ่งขึ้น โครงงานนี้จะใช้บอร์ด Raspberry Pi 3 เป็นตัวดักจับคลื่นสัญญาณต่าง ๆ ในบริเวณโดยรอบ จากนั้นทำการการวิเคราะห์ความเป็นเอกลักษณ์ของโดรนที่รับส่งข้อมูลกับอุปกรณ์ควบคุมบนเครือข่ายไร้สาย ว่าสัญญาณนั้นเป็นสัญญาณของโดรนหรือไม่ มีส่วนแสดงผลสำหรับผู้ใช้งาน โดยผู้ใช้งานสามารถควบคุมการทำงานของระบบและรับการแจ้งเตือนได้จากส่วนแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อพัฒนาระบบตรวจจับโดรนที่บินเข้ามาในพื้นที่ โดยใช้เอกลักษณ์ของโดรนที่มีการรับส่งข้อมูลกับอุปกรณ์ควบคุม
2) เพื่อเพิ่มประสิทธิภาพในการตรวจจับโดรนในเวลากลางคืน
3) เพื่อพัฒนาระบบตรวจจับโดรนที่มีราคาถูก และติดตั้งได้ง่าย
4) เพื่อศึกษาและวิเคราะห์คลื่นสัญญาณวิทยุ
5) เพื่อศึกษาและทดลองการใช้บอร์ด Raspberry Pi ในการประมวลผลและตรวจจับ

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถนำระบบตรวจจับมาคาดการณ์เอกลักษณ์ข้อมูลที่รับส่งของโดรนและอุปกรณ์ควบคุมได้
2) เข้าใจระบบการทำงานของการรับส่งของคลื่นสัญญาณไวไฟ
3) เข้าใจหลักการทำงานของ Raspberry Pi
4) สามารถนำระบบตรวจจับโดรนไปพัฒนาต่อได้

1.4 ขอบเขตของโครงงาน

1) จำแนกคลื่นสัญญาณไวไฟ เฉพาะในช่วงความถี่ 2.4 - 2.5 GHz
2) แสดงผลการตรวจจับผ่านส่วนติดต่อผู้ใช้งาน (User Interface)
3) รัศมีในการตรวจจับประมาณ 1 - 15 เมตร
4) สามารถจำแนกและตรวจจับคลื่นสัญญาณไวไฟของโดรนเฉพาะรุ่นที่กำหนดคือ Skyhunter X8

1.5 วิธีการดำเนินงาน

1) กำหนดวัตถุประสงค์และขอบเขตการทำงานของโครงงาน
2) ศึกษาและรวบรวมข้อมูลของเทคโนโลยีที่ใช้สำหรับดักจับข้อมูล
3) ออกแบบและวิเคราะห์ระบบตรวจจับข้อมูล เนื้อหา และเอกสารที่เกี่ยวข้องที่ได้รวบรวม
4) ทดลองการใช้งานเทคโนโลยีที่ใช้สำหรับดักจับข้อมูลบน Raspberry Pi

1.6 ข้อจำกัดโครงงาน

1) สามารถจำแนกและตรวจจับได้เฉพาะโดรนที่ควบคุมผ่านสมาร์ทโฟนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.7 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ ได้ทำการแบ่งเนื้อหาออกเป็น 5 บท ซึ่งมีรายละเอียดดังนี้

1) บทนำ จะกล่างถึง ที่มาและความสำคัญของโครงงาน วัตถุประสงค์ของโครงงาน ประโยชน์ที่คาดว่าจะได้รับ ขอบเขตการทำงาน วิธีการดำเนินงาน และส่วนประกอบของโครงงาน
2) ทฤษฎีที่เกี่ยวข้อง จะกล่าวถึงทฤษฎีและเทคโนโลยีที่นำมาใช้เป็นเครื่องมือในการพัฒนาโครงงานนี้ ได้แก่ Raspberry Pi, IEEE 802.11, Machine Learning, Random Forest, Flask, Bootstrap และงานวิจัยต่าง ๆ ที่เกี่ยวข้อง
3) การออกแบบและพัฒนา จะกล่าวถึง ภาพรวมและโครงสร้างของระบบ, การออกแบบการทำงานของระบบ
4) การทดลองและแผนการดำเนินงาน จะกล่าวถึงการทดลองทดลองการทำงานของ Monitor Mode บน Raspberry Pi 3 และทดลองการดักจับสัญญาณ (Sniffing) บนเครือข่ายไร้สาย
5) บทสรุปและข้อเสนอแนะ จะกล่าวถึง บทสรุป, ปัญหาและอุปสรรค และแนวทางการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 เทคโนโลยี UAV

อากาศยานไร้คนขับ หรือ UAV (Unmanned Aerial Vehicle) หมายถึงอากาศยานไร้คนขับเป็นพาหนะไร้คนขับที่สามารถควบคุมได้จากระยะไกล มีรูปร่างต่าง ขนาด ที่แตกต่างกันออกไป หรือที่เรียกกันในว่า " โดรน " เป็นอากาศยานที่สามารถควบคุมได้จากระยะไกล มีการควบคุม 2 ลักษณะ คือ ระบบควบคุมจากระยะไกล (Remote Controller) และ ระบบควบคุมอัตโนมัติด้วยระบบคอมพิวเตอร์ ในปัจจุบันเทคโนโลยีนี้ถูกนำมาประยุกต์ในหลายรูปแบบ เช่น การจัดส่งพัสดุ การสำรวจพื้นที่ต่าง ๆ หรือสื่อบันเทิง ซึ่งต่างจากในอดีตที่ถูกพัฒนาเพื่อการทหารเท่านั้น

โดยในปัจจุบันโดรนได้พัฒนาให้สามารถควบคุมระยะไกลด้วยสมาร์ทโฟนหรืออุปกรณ์ที่รองรับระบบเครือข่ายไร้สายได้ ซึ่งทำงานในช่วงความถี่ 2.4 GHz และ 5 GHz มาตรฐาน IEEE 802.11

รูป 2.1 โดรนที่พบเห็นได้ในปัจจุบัน(ซ้าย) และ โดรนทางการทหาร(ขวา)

2.2 Raspberry Pi

Raspberry Pi เป็นคอมพิวเตอร์ขนาดเล็กที่สามารถแสดงข้อมูลทางจอแสดงผลที่รองรับ HDMI สามารถใช้งานคีย์บอร์ดและเมาส์ได้ เหมาะสำหรับการพัฒนาและประยุกต์ใช้ในระบบอิเล็กทรอนิกส์ และที่สำคัญมีราคาที่ถูก เมื่อเทียบกับความสามารถที่สามารถทำได้เกือบทุกอย่างเหมือนกับเดสก์ท็อปคอมพิวเตอร์ทั่วไป

Raspberry Pi ถูกพัฒนาโดย Raspberry Pi Foundation ในประเทศอังกฤษ รองรับระบบปฏิบัติการของ Linux ได้หลายระบบ เช่น Raspbian, Ubuntu เป็นต้น ซึ่งในปัจจุบันมีอยู่หลายรุ่น แบ่งออกเป็น 2 โมเดลหลัก ๆ คือ Model A และ B โดย Raspberry Pi 3 Model B เป็นโมเดลที่เลือกใช้สำหรับโครงการนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.2 Raspberry Pi 3 Model B

คุณสมบัติของ Raspberry Pi model B

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 มาตรฐานระบบเครือข่ายไร้สาย (IEEE 802.11)

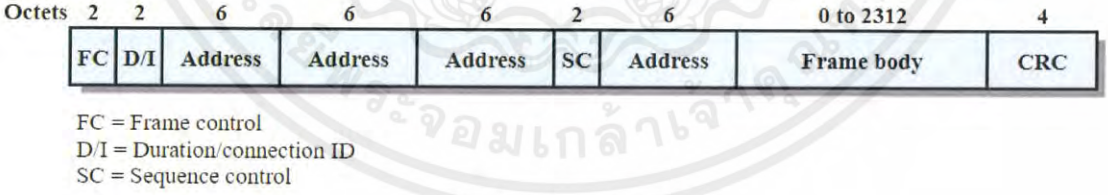
เป็นมาตรฐานการทำงานของระบบเครือข่ายไร้สาย ที่ถูกกำหนดโดยสถาบันวิชาชีพวิศวกรไฟฟ้าและอิเล็กทรอนิกส์ เพื่อเป็นมาตรฐานกลางในการนำไปใช้สร้างระบบสำหรับเครือข่ายไร้สาย โดยปัจจุบันมีมาตรฐานพัฒนาออกมามากมาย เพื่อให้รองรับต่อความเร็วการใช้งานในปัจจุบัน

ช่วงความถี่ที่กำหนดใช้ในการส่งสัญญาณในมาตรฐานที่นิยมใช้ในปัจจุบันมีอยู่ 2 ช่องความถี่ คือ 2.4 GHz และ 5 GHz ตารางที่ 2.1 แสดงรายละเอียดของมาตรฐานแต่ละมาตรฐาน

ตาราง 2.1 มาตรฐานของระบบเครือข่ายไร้สาย

มาตรฐาน	ความถี่ (GHz)	แบนด์วิธ (MHz)	Max Data Rate (Mbps)	โมดูเลชัน
IEEE 802.11	2.4	20	2	DSSS, FHSS
IEEE 802.11a	5	20	54	OFDM
IEEE 802.11b	2.4	20	11	DSSS
IEEE 802.11g	2.4	20	54	OFDM, DSSS
IEEE 802.11n	2.4/5	20/40	72.2/150	OFDM

โดยการทำงานของ IEEE 802.11 มี 2 Layer คือ Physical Layer และ Datalink Layer โดยที่ Datalink Layer มี MAC สำหรับใช้ในการติดต่อ มีลักษณะของข้อมูลที่ใช้สำหรับติดต่อกันนั้นได้ดังรูป 2.3

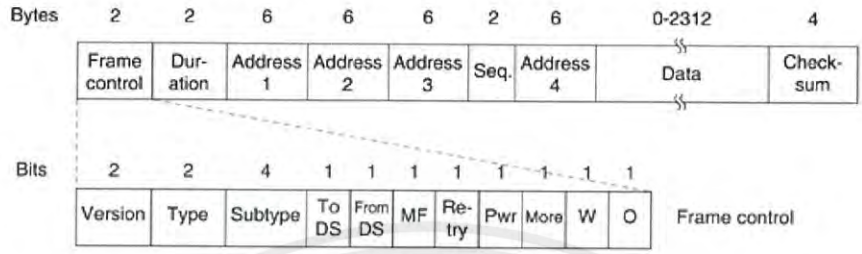


รูป 2.3 MAC Frame

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 Frame Control

ส่วนสำหรับจัดการและควบคุมการรับส่งข้อมูลในแมคเฟรมนั้น ๆ โดยรายละเอียดดังรูป 2.4

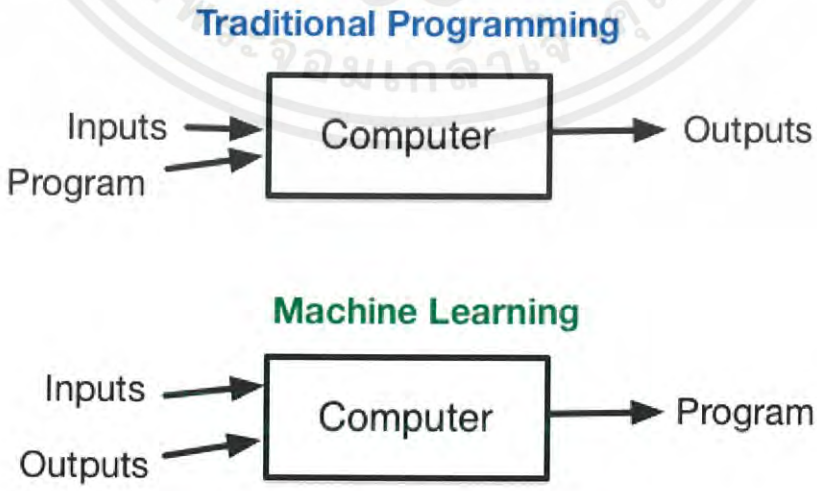


รูป 2.4 Frame Control

โดย Frame Control นั้นจะประกอบไปด้วย 11 ฟิลด์ ซึ่งในฟิลด์ต่าง ๆ นั้นจะมีฟิลด์ To DS และฟิลด์ From DS ที่เป็นส่วนที่มีความสำคัญในการระบุข้อมูลที่รับมานั้น เป็นข้อมูลที่รับเข้ามาหรือส่งออกไป

2.4 Machine Learning

Machine Learning หรือส่วนการเรียนรู้ของเครื่อง เป็นส่วนที่ถูกใช้งานดั่งสมองของ AI (Artificial Intelligence) หรือเป็นปัญญาประดิษฐ์นั้น จะเรียกใช้ Machine Learning ในการสร้างความฉลาดให้กับ AI โดย Machine Learning จะเรียนรู้จากข้อมูลที่เราส่งไปกระตุ้น แล้วจดจำเอาไว้ และจะส่งผลลัพธ์ออกมาเป็นตัวเลขหรือโค้ดไปแสดงผล



รูป 2.5 ภาพรวมของ Machine Learning

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในการสร้าง Machine Learning นั้นต้องอาศัยอัลกอริทึม ที่มีการออกแบบมาเพื่อนำไปใช้ให้เหมาะสมกับข้อมูลที่ต้องการจะนำมาใช้ และ Machine Learning นั้นแยกออกเป็นประเภทต่าง ๆ ตามหน้าที่และขอบเขตได้ คือ

2.4.1 Supervised Learning

คือการเรียนรู้โดยที่เราจะต้องป้อนข้อมูลขาเข้า คือข้อมูลตัวแปรต่าง ๆ และป้อนข้อมูลขาออก หรือก็คือผลลัพธ์ที่ได้มาจากข้อมูลตัวแปรขาเข้า แล้วโปรแกรมจะทำการเรียนรู้และสร้างความเชื่อมโยงของข้อมูลที่ป้อนเข้าไป ทำให้สามารถรู้ได้ว่าข้อมูลที่เข้าไป จะมีข้อมูลที่ออกมาเป็นอย่างไร

2.4.2 Unsupervised Learning

เป็นการเรียนรู้ที่ตรงข้ามกับ Supervised Learning คือจะป้อนข้อมูลขาเข้าอย่างเดียวเท่านั้นเข้าไปให้โปรแกรมเรียนรู้ โดยไม่มีการป้อนคำตอบไปให้ เป็นการให้โปรแกรมหาความเชื่อมโยงเอาเอง

2.5 Scikit Learn

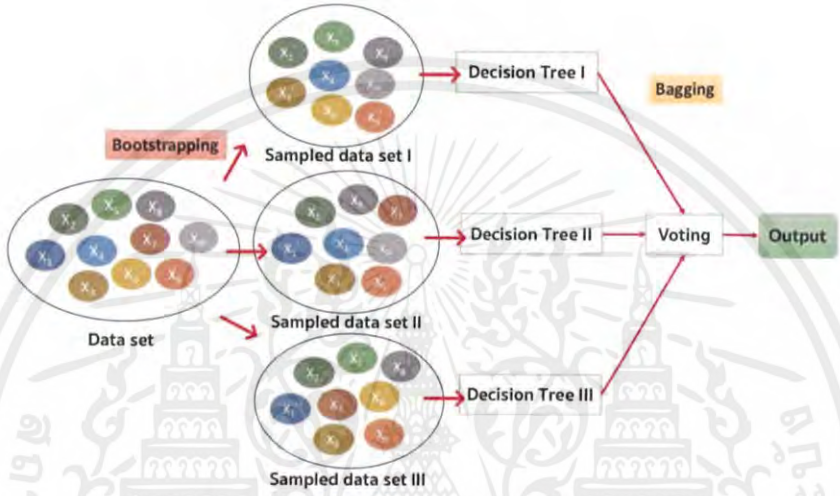
โมดูลสำหรับใช้ในการทำ Machine Learning และ Data Mining ในภาษา Python เป็นโมดูลที่มีมีอัลกอริทึมให้เลือกใช้มากมายมีทั้งแบบ Supervised และ Unsupervised

รูป 2.6 สัญลักษณ์ของ Scikit Learn

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 อัลกอริทึม Random Forest

เป็นอัลกอริทึมสำหรับ Machine Learning ที่มีความยืดหยุ่น และเป็นหนึ่งในอัลกอริทึมที่นิยมในการนำมาใช้มาก โดยมีหลักการคือ จะสร้างโมเดลจาก Decision Tree หลาย ๆ โมเดล โดยแต่ละโมเดลจะได้รับดาต้าเซทที่ไม่เหมือนกัน ซึ่งเป็นสับเซตของดาต้าเซททั้งหมด ซึ่งขั้นตอนการทำนายผลนั้น จะให้แต่ละ Decision Tree ทำการทำนายผลในแต่ละ Decision Tree และคำนวณผลการทำนายด้วยการ Vote Output ที่ ถูกเลือกโดย Decision Tree มากที่สุด



รูป 2.7 หลักการทำงานของ Random Forest

2.7 Web Framework

2.7.1 Flask

Flask คือ Web Framework ที่เขียนขึ้นมาสำหรับ Python เพื่อใช้ร่วมกันกับ Webserver เช่น Apache และได้รับการยอมรับจาก Community web pages ชั้นนำเช่น Pinterest, LinkedIn เป็นต้น โดยน Flask นั้นถูกเรียกว่า Micro Framework เพราะว่าไม่ต้องการเครื่องมือ หรือ Library อะไรมาก อีกทั้งไม่จำเป็นต้องมี Database ด้วย

รูป 2.8 สัญลักษณ์ของ Flask

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 Bootstrap

Bootstrap คือ Web Framework สำหรับพัฒนาหน้า Front-end ที่จะทำให้สามารถพัฒนา Web Application สามารถทำได้สะดวก รวดเร็ว และง่ายขึ้น เป็น Web Framework ที่รวบรวม HTML, CSS และ JS เข้าด้วยกันสำหรับพัฒนา Web ที่รองรับทุก Smart Device หรือ เรียกว่า Responsive Web หรือ Mobile First

รูป 2.9 สัญลักษณ์ของ Bootstrap

2.8 งานวิจัยที่เกี่ยวข้อง

2.8.1 Detecting drones using machine learning [Waylon Dustin Scheller. 2017]

Waylon D. Scheller จากมหาวิทยาลัยไอโอวาสเตต ได้ศึกษาการใช้ Machine Leaning ในการตรวจจับโดรนด้วยการทดลองใช้อุปกรณ์รับสัญญาณวิทยุ HackRF One เป็นเครื่องมือบันทึกความถี่วิทยุในระหว่างที่โดรนทำการบินและลงจอด โดยจะบันทึก Frequency Channel และ RSSI ของโดรนที่สื่อสารกันระหว่างเครื่องควบคุมจากนั้นนำข้อมูลการบินที่บันทึกได้ไปทำนายการตรวจจับโดรนด้วย Machine Learning Models

จากงานวิจัยนี้พบว่าโดรนที่มีการสร้างเครือข่ายไวไฟ (WiFi Access Point) ขึ้นมาเพื่อทำการเชื่อมต่อและสื่อสารกับเครื่องควบคุม เช่น โดรนของ 3DR Solo สามารถตรวจจับได้โดยใช้ Machine Learning Models ที่เทรนมาแล้วและมีความแม่นยำในการตรวจจับประมาณ 75% อย่างไรก็ตามยังมีโดรนอีกมากที่เครื่องส่งสัญญาณไม่มีการสร้างเครือข่ายไวไฟ (WiFi Access Point) ขึ้นมา เช่น โดรนของ DJI เป็นต้น โดยโดรนประเภทนี้อาจจะใช้การสื่อสารด้วยวิธีการ FHSS หรือ DSSS ซึ่งเป็นการจัดสรรคลื่นความถี่ 2.4 GHz ด้วยเทคโนโลยีที่เรียกว่า Spread Spectrum แต่ถึงแม้จะใช้การส่งสัญญาณด้วยเทคโนโลยี Spread Spectrum โดรนรูปแบบนี้ก็ยังสามารถถูกตรวจพบได้ด้วย Machine Learning Models ที่เทรนมาแล้วได้แต่จะมีความแม่นยำน้อยกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 Unauthorized Amateur UAV Detection Based on WiFi Statistical Fingerprint Analysis [Igor Bisio and others. 2018 : 106 - 111]

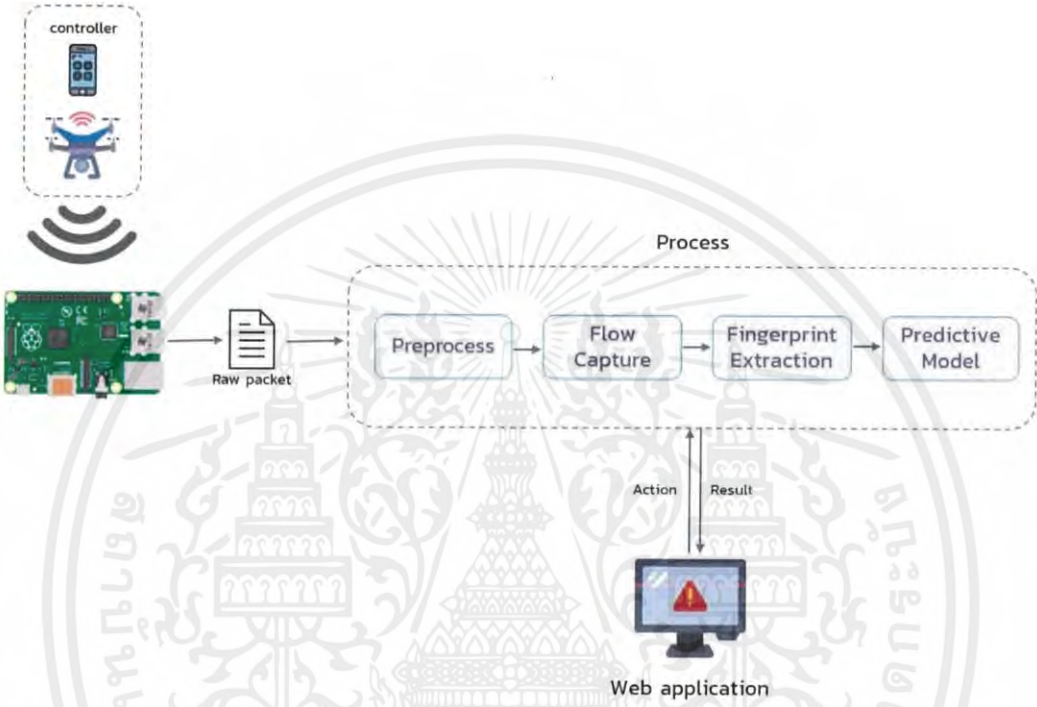
บทความนี้ศึกษาวิธีการตรวจจับโดรนด้วยวิธีต่าง ๆ และสุดท้ายได้นำเสนอวิธีการตรวจจับโดรนโดยใช้ประโยชน์จากข้อมูลการสื่อสารระหว่างอุปกรณ์ควบคุมกับโดรน และนำข้อมูลมาวิเคราะห์เพื่อตรวจจับโดรนโดยใช้อัลกอริทึม Random Forest, SMO, Logit Boost เป็นต้น

ผลจากการวิจัยและทดลองพบว่าโมเดลที่สร้างขึ้นมาเพื่อตรวจจับโดรนนั้นสามารถตรวจจับโดรนได้ในสถานะการต่าง ๆ เช่น มีโดรนหนึ่งตัว หรือมีโดรนมากกว่าหนึ่งตัว และมีความแม่นยำในการตรวจจับประมาณ 87% – 95%

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและพัฒนา

3.1 ภาพรวมโครงสร้างของระบบ



รูป 3.1 System Diagram

โครงงานนี้มีความต้องการที่จะตรวจจับการมีอยู่ของโดรนด้วยการใช้ประโยชน์จากสัญญาณไวไฟ ที่มีอยู่ทั่วไป โดยภาพรวมของการทำงานจะแบ่งเป็น 3 ส่วนหลัก ๆ คือ ส่วนที่ทำการดักจับข้อมูลการสื่อสารของอุปกรณ์เน็ตเวิร์ค, ส่วนที่ทำการวิเคราะห์ข้อมูลที่ได้มา และส่วนสุดท้ายทำหน้าที่ควบคุมการจับข้อมูลและแสดงผลการวิเคราะห์ข้อมูลการตรวจจับโดรน

3.2 User Requirement

3.2.1 Input

- สามารถตรวจจับสัญญาณไวไฟของโดรน ในรัศมี 1-15 เมตรได้

3.2.2 Process

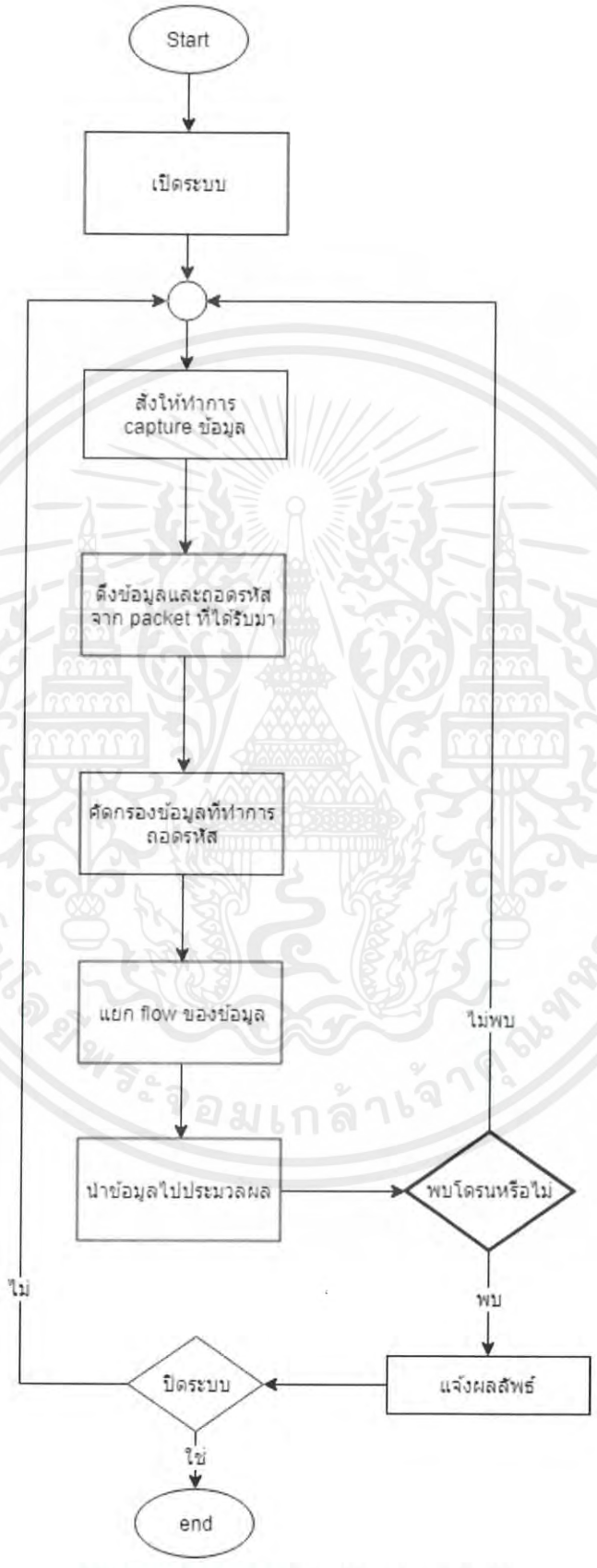
- สามารถวิเคราะห์ข้อมูลจากแพ็คเก็ตที่ตรวจจับมาจากสัญญาณไวไฟได้

3.2.3 Output

- สามารถแจ้งเตือนผู้ใช้เมื่อตรวจพบเจอโดรนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

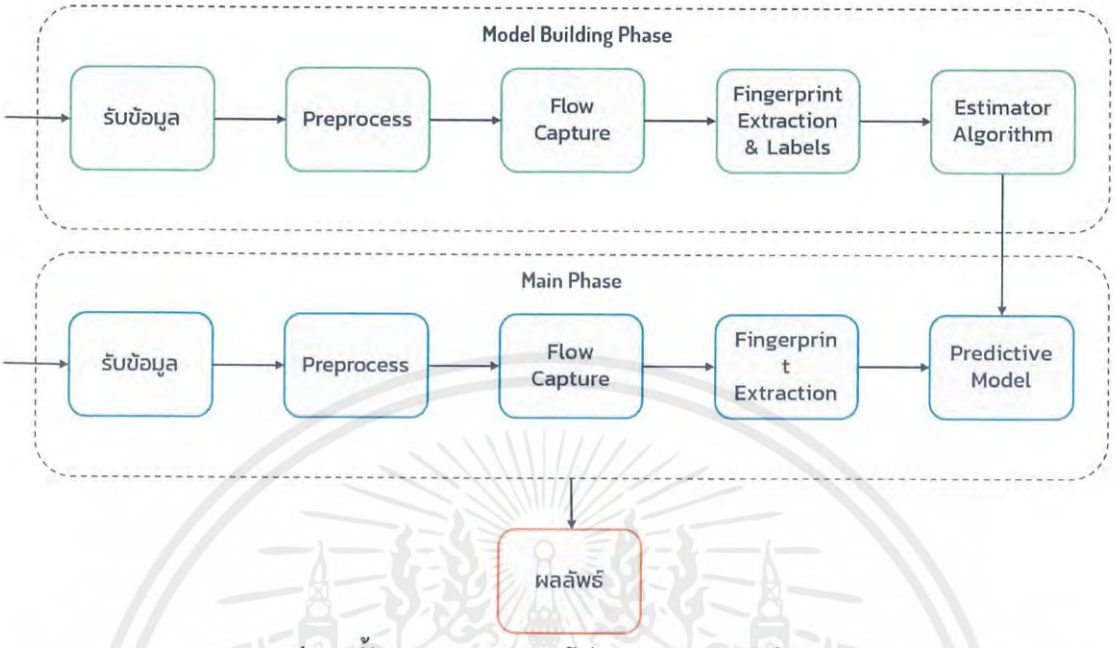
3.3 Flowchart ภาพรวมการทำงานของระบบ



รูป 3.2 Flow Chart การทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

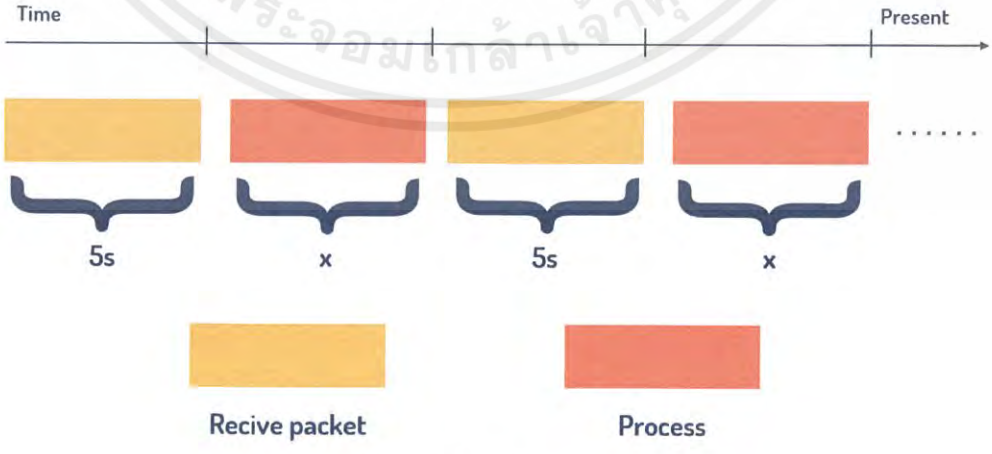
3.4 การพัฒนาโปรแกรมตรวจจับโดรน



รูป 3.3 ขั้นตอนการพัฒนาโปรแกรมตรวจจับโดรน

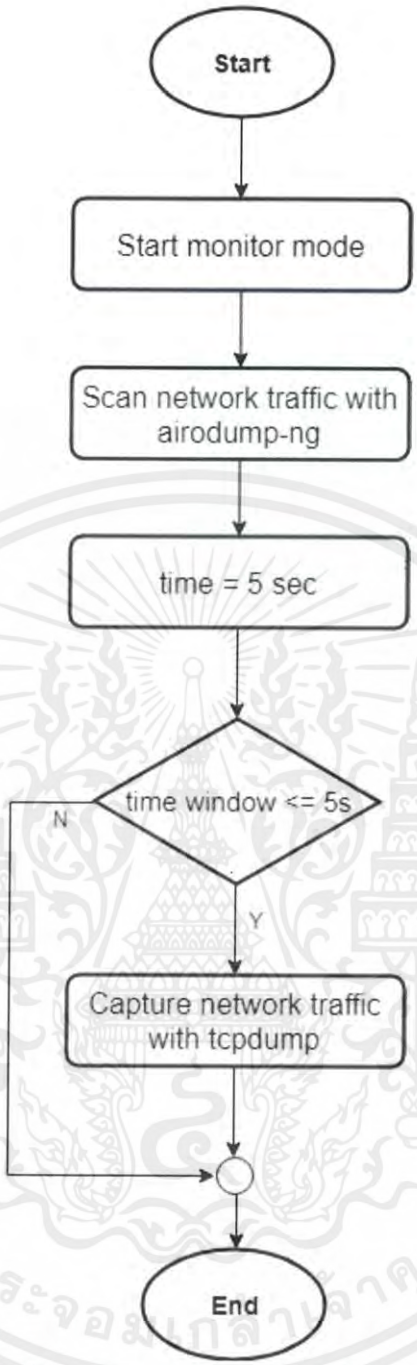
3.4.1 การรับข้อมูล

การตรวจจับการมีอยู่ของโดรน โครงงานนี้จะใช้วิธีการตรวจจับแบบ Passive listening สำหรับการตรวจจับด้วยวิธี Passive listening นี้สามารถดักจับข้อมูลการสื่อสารระหว่าง AP กับ STA ได้โดยไม่จำเป็นต้องเชื่อมต่อกับอุปกรณ์นั้น ๆ การดำเนินงานในส่วนนี้จะทำการกำหนดช่วงเวลา เพื่อแบ่งปริมาณข้อมูลที่ดักจับออกมาเป็นส่วนเล็ก ๆ ส่วนละ 5 วินาที จากนั้นก็นำข้อมูลแต่ละส่วนไปประมวลผลเพื่อคำนวณหาเอกลักษณ์ของข้อมูล (Fingerprint)



รูป 3.4 ขั้นตอนการดักจับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

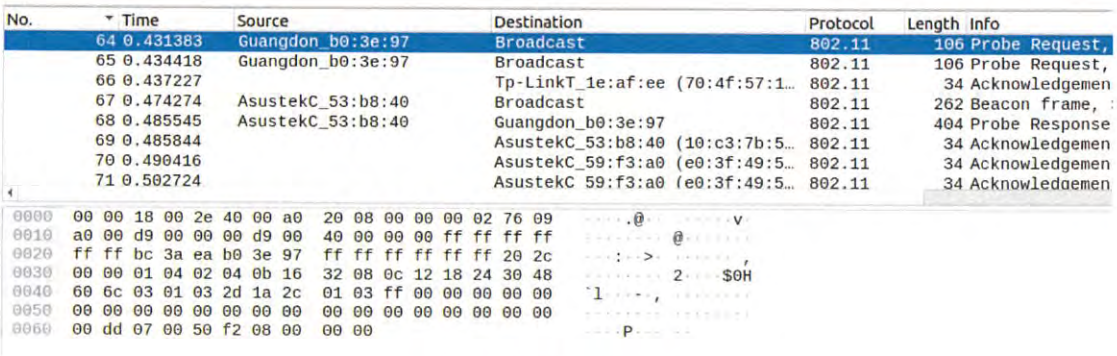


รูป 3.5 Flowchart การดักจับข้อมูล

3.4.2 การเตรียมข้อมูล

ก่อนที่จะนำข้อมูลที่ได้ไปดำเนินการต่อจำเป็นจะต้องทำการคัดกรองข้อมูลที่ไม่เกี่ยวข้องหรือข้อมูลเสียออกจากแหล่งข้อมูลดิบเพื่อให้ได้ข้อมูลที่สัมพันธ์กันก่อน สำหรับข้อมูลที่เราต้องการคือ Timestamp, Source Address, Destination Address, Transmitter Address, Receiver Address, DS flag, Packet Length และข้อมูลที่จำเป็นจะต้องคัดกรองออกคือ ข้อมูลที่ Source หรือ Destination เป็น None และข้อมูลที่ Destination ส่งค่า Broadcast

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



No.	Time	Source	Destination	Protocol	Length	Info
64	0.431383	Guangdon_b0:3e:97	Broadcast	802.11	106	Probe Request,
65	0.434418	Guangdon_b0:3e:97	Broadcast	802.11	106	Probe Request,
66	0.437227		Tp-LinkT_1e:af:ee (70:4f:57:1…	802.11	34	Acknowledgemen
67	0.474274	AsustekC_53:b8:40	Broadcast	802.11	262	Beacon frame,
68	0.485545	AsustekC_53:b8:40	Guangdon_b0:3e:97	802.11	404	Probe Response
69	0.485844		AsustekC_53:b8:40 (10:c3:7b:5…	802.11	34	Acknowledgemen
70	0.490416		AsustekC_59:f3:a0 (e0:3f:49:5…	802.11	34	Acknowledgemen
71	0.502724		AsustekC_59:f3:a0 (e0:3f:49:5…	802.11	34	Acknowledgemen

```
0000  00 00 18 00 2e 40 00 a0  20 08 00 00 00 02 76 09   .....@.. ......v.
0010  a0 00 d9 00 00 00 d9 00  40 00 00 00 ff ff ff ff   ........ @.......
0020  ff ff bc 3a ea b0 3e 97  ff ff ff ff ff ff 20 2c   ...:..>. ...... ,
0030  00 00 01 04 02 04 0b 16  32 08 0c 12 18 24 30 48   ........ 2....$0H
0040  60 6c 03 01 03 2d 1a 2c  01 03 ff 00 00 00 00 00   `l...-., ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 dd 07 00 50 f2 08 00  00 00                     ....P... ..
```

รูป 3.6 ตัวอย่างข้อมูลดิบ

	Timestamp	fc_dsflag	Source	Destination	Transmitter	Receiver	Lenth
3	2019-04-12 19:17:18.732507	1	bc:3a:ea:b0:3e:97	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	24:72:60:26:31:9e	104
13	2019-04-12 19:17:18.757215	2	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	1543
15	2019-04-12 19:17:18.757703	2	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	1543
17	2019-04-12 19:17:18.758139	2	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	1543
19	2019-04-12 19:17:18.760441	2	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	1543
21	2019-04-12 19:17:18.760835	2	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	1543
24	2019-04-12 19:17:18.763989	2	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	1543
26	2019-04-12 19:17:18.764456	2	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	1543
28	2019-04-12 19:17:18.764889	2	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	24:72:60:26:31:9e	bc:3a:ea:b0:3e:97	1543

รูป 3.7 ตัวอย่างข้อมูลที่ทำ Preprocess

3.4.3 การรวบรวม Flow

หลังจากเตรียมข้อมูลเสร็จแล้ว ในขั้นตอนนี้จะทำการรวบรวม flow ของข้อมูล โดยกำหนดให้ flow เป็นกลุ่มข้อมูลของแพ็กเก็ตใด ๆ ที่ถูกจัดกลุ่มโดยใช้ค่า MAC ต้นทาง (Source), MAC ปลายทาง (Destination) และ ถ้าแพ็กเก็ตใดมี MAC ต้นทาง (Source), MAC ปลายทาง (Destination) เหมือนกันหรือสลับกัน ถือว่าแพ็กเก็ตนั้นอยู่ในกลุ่ม flow เดียวกัน

```
((Source, Destination) or (Destination, Source)) {
    Packet i   {Timestamp, Source, Destination, Transmitter, Receiver, DS flag, Packet Length}
    Packet i+1 {Timestamp, Source, Destination, Transmitter, Receiver, DS flag, Packet Length}
    Packet N   {Timestamp, Source, Destination, Transmitter, Receiver, DS flag, Packet Length}
}
```

รูป 3.8 โครงสร้างข้อมูลการสื่อสารระหว่าง AP กับ STA

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 Fingerprint Extraction

ส่วนนี้เราจะอธิบายขั้นตอนการวิเคราะห์เชิงปริมาณที่เกี่ยวข้องกับ flow ของข้อมูล เพื่อนำไปหาค่าของ fingerprint โดยจะนำข้อมูลจากหัวข้อที่แล้วมาใช้ในการคำนวณ ได้ดังนี้

1) ผลรวมจำนวนแพ็กเก็ต (total_pack) คำนวณจากผลรวมของแพ็กเก็ตทั้งหมดใน flow นั้น
2) ระยะเวลารวมของแพ็กเก็ต (duration) คำนวณจาก time delta ระหว่างแพ็กเก็ตที่ N กับแพ็กเก็ตที่ 1
3) ความยาวเฉลี่ยแพ็กเก็ต (mean_len) คำนวณจากผลรวมขนาดของแต่ละแพ็กเก็ตหารด้วยจำนวนแพ็กเก็ตทั้งหมดใน flow นั้น
4) ค่า RMS ความยาวแพ็กเก็ตโดยเฉลี่ย (len_rms) คำนวณจากการหาค่าเฉลี่ยกำลังสองของขนาดแพ็กเก็ตใน flow นั้น
5) ระยะเวลาในการรอคอยของแพ็กเก็ตโดยเฉลี่ย (mean_delta) คำนวณจากผลรวมของ time delta ของแพ็กเก็ตที่ i + 1 กับแพ็กเก็ตที่ i หารด้วยจำนวนแพ็กเก็ต - 1 ใน flow นั้น
6) ค่า RMS ระยะเวลาในการรอคอยของแพ็กเก็ตโดยเฉลี่ย (delta_rms) คำนวณจากการหาค่าเฉลี่ยกำลังสองของ time delta ระหว่างแพ็กเก็ตที่ i + 1 กับแพ็กเก็ตที่ i หารด้วย จำนวนแพ็กเก็ต - 1 ใน flow นั้น
7) จำนวน Embedded Packet (embedded_pack) คำนวณโดยนำค่า MAC ของเครื่องส่ง (Trasmitter) กับ MAC ของเครื่องรับ (Receiver) มาเปรียบเทียบกับเครื่องต้นทาง (Source) และเครื่องปลายทาง (Destination) ตามลำดับถ้าข้อมูลของ MAC ตรงกันให้ถือว่าเป็น Embedded Packet
8) จำนวนของแพ็กเก็ตที่มี DS flags เฉพาะ (mode1, mode2, mode3, mode4) คำนวณโดยคิดจาก DS flags จะมีค่าได้ตั้งแต่ 1 – 4 ซึ่งแต่ละค่าจะทำการเพิ่มค่าให้แต่ละ mode ตามลำดับ เช่นถ้า DS flags มีค่าเป็น 0 ก็จะเพิ่มค่าให้ mode1 ไปหนึ่งค่า เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม 3.1 - Function การคำนวณหา Fingerprint

```
def extraction(flow_dict):
    feature = []
    if flow_dict is None:
        return

    for key, flow in flow_dict.items():
        if len(flow) > 1:
            total_len = 0
            total_time_delta = 0
            total_len_p = 0
            total_time_delta_p = 0
            embedded_pack = 0
            mode1 = 0
            mode2 = 0
            mode3 = 0
            mode4 = 0

            # Total number of packets 1
            total_pack = len(flow)
            # Total duration of the flow 2
            duration = (flow[len(flow) - 1][0] - flow[0][0]).total_seconds()

            for idx, packet in enumerate(flow):
                if idx < len(flow)-1:
                    total_time_delta = total_time_delta + \
                        (flow[idx + 1][0] - packet[0]).total_seconds()
                    total_time_delta_p = total_time_delta_p + \
                        ((flow[idx + 1][0] - packet[0]).total_seconds() ** 2)

                total_len = total_len + packet[1][5]
                total_len_p = total_len_p + (packet[1][5] ** 2)

                # Number of embedded packets 7
                if (packet[1][0] == packet[1][2]) and (packet[1][1] == packet[1][3]):
                    embedded_pack += 1

                # Number of packets with specific DS status flags 8
                if packet[1][4] == 0:
                    mode1 += 1
                elif packet[1][4] == 1:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            mode2 += 1
        elif packet[1][4] == 2:
            mode3 += 1

    # Average packet length 3
    mean_len = total_len / len(flow)
    # Root mean square (RMS) of packet length 4
    rms = total_len_p / len(flow)
    len_rms = math.sqrt(rms)

    # Average packets inter-arrival time 5
    mean_delta = total_time_delta / (len(flow) - 1)
    # RMS of packets inter-arrival time 6
    rms = total_time_delta_p / (len(flow) - 1)
    delta_rms = math.sqrt(rms)

    feature.append((total_pack, duration, mean_len, len_rms,
                    mean_delta, delta_rms, embedded_pack, mode1, mode2, mode3, mode4))

return feature
```

ท้ายที่สุดแล้วจะได้ค่า Fingerprint ที่อธิบาย flow ข้อมูลของแพ็กเก็ต ซึ่งถูกกำหนดให้เป็น feature vector ดังรูป 3.9

	total_pack	duration	mean_len	len_rms	mean_delta	delta_rms	embedded_pack	mode1	mode2	mode3	mode4
0	88	3.921367	1392.386364	1451.276664	0.045073	0.390426	88	0	6	82	0
1	17	0.717716	361.000000	670.920921	0.044857	0.130535	17	0	12	5	0
2	2	0.036881	162.000000	162.197411	0.036881	0.036881	0	0	0	2	0
3	2	0.027364	217.000000	217.147415	0.027364	0.027364	0	0	0	2	0
4	174	4.770308	1387.586207	1453.316398	0.027574	0.282072	174	2	14	158	0

รูป 3.9 ตัวอย่างข้อมูล Feature Vector

3.4.5 การรวบรวม dataset

การเก็บรวมรวม dataset ที่จะนำมา train model ในการตรวจจับโดรนนั้น ทางผู้จัดทำได้ทำการเก็บรวมรวมข้อมูลโดยเริ่มทำตั้งแต่ขั้นตอนที่ 3.4.1 – 3.4.4 ซึ่งจะทำการเก็บรวบรวม flow การทำงานระหว่างโดรนกับเครื่องควบคุม หรือ flow การทำงานของอุปกรณ์อื่น ๆ บนเครือข่ายเน็ตเวิร์ค เราจะทำการเก็บข้อมูลการสื่อสารของโดรนในระยะตั้งแต่ 1 เมตร – 15 เมตร ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Total_pack	Duration	Mean_len	Len_rms	Mean_delta	Deata_rms	Embeeded_p	Mode1	Mode2	Mode3
86	0.28	1431.07	1477.44	0	0.01	86	0	6	80

รูป 3.10 ตัวอย่าง fingerprint ของโดรนในระยะ 1 เมตร

Total_pack	Duration	Mean_len	Len_rms	Mean_delta	Deata_rms	Embeeded_p	Mode1	Mode2	Mode3
70	0.22	1245.89	1366.42	0	0.01	70	4	8	58

รูป 3.11 ตัวอย่าง fingerprint ของโดรนในระยะ 4 เมตร

Total_pack	Duration	Mean_len	Len_rms	Mean_delta	Deata_rms	Embeeded_p	Mode1	Mode2	Mode3
66	0.25	996.76	1218.9	0	0.01	66	16	6	44

รูป 3.12 ตัวอย่าง fingerprint ของโดรนในระยะ 6 เมตร

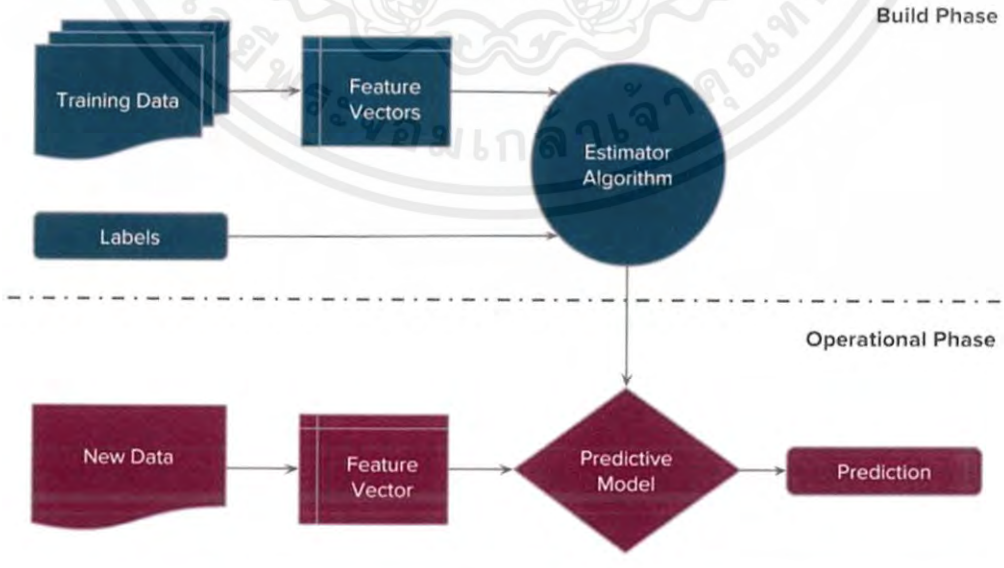
Total_pack	Duration	Mean_len	Len_rms	Mean_delta	Deata_rms	Embeeded_p	Mode1	Mode2	Mode3
43	0.24	1155.05	1313	0.01	0.01	43	0	7	36

รูป 3.13 ตัวอย่าง fingerprint ของโดรนในระยะ 10 เมตร

Total_pack	Duration	Mean_len	Len_rms	Mean_delta	Deata_rms	Embeeded_p	Mode1	Mode2	Mode3
25	4.16	698.44	982.47	0.17	0.64	25	2	11	12

รูป 3.14 ตัวอย่าง fingerprint ของโดรนในระยะ 14 เมตร

3.4.6 การสร้างและทดสอบโมเดลในการตรวจจับโดรน



รูป 3.15 ขั้นตอนการสร้างโมเดลในการตรวจจับโดรน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นตอนนี้เราจะนำข้อมูลที่ได้เก็บรวบรวมมาแยกคอลัมน์ที่เป็น feature กับคอลัมน์ที่เป็น target ออกจากกันก่อน และทำการแยกข้อมูลออกเป็น train กับ test โดยจะแบ่งข้อมูลส่วน train เป็น 80% และข้อมูลส่วน test เป็น 20% หลังจากที่เราแบ่งข้อมูลเสร็จแล้วก็จะทำการ scaling ข้อมูลเพื่อให้ข้อมูลมีค่าที่ใกล้เคียงกัน และลดปริมาณข้อมูลที่มีขนาดใหญ่เกินไป เมื่อเตรียมการทุกอย่างเสร็จแล้วก็จะนำข้อมูลที่ได้แบ่งไว้ไปเทรนโดยใช้ Algorithm Random forest

การ train model จะนำข้อมูลของ feature และข้อมูลของ target ที่ได้เตรียมไว้ในส่วนของการเทรน นำมาเทรนกับอัลกอริทึม Random Forest โดยจะทำการสุ่มปรับค่า parameter ได้แก่ n_estimators (จำนวนต้นไม้ใน Random Forest), max_features (วิธีการโหวต), max_depth (ความลึกของต้นไม้), criterion (วิธีการตัดสินใจของต้นไม้) เพื่อให้ได้ค่า parameter ที่ทำให้โมเดลของเราทำนายผลได้ดีที่สุด โดยในการปรับค่าจะใช้ gridsearchCV เป็นตัวช่วยในการสุ่มค่า เมื่อได้ค่าที่ดีที่สุดแล้วก็ทำการเทรนโมเดลจากข้อมูลที่ได้เตรียมเอาไว้

หลังจากที่ทำการเทรนโมเดลเสร็จแล้ว เราจะนำข้อมูลของ feature และข้อมูลของ target ที่ได้เตรียมไว้ในส่วนของการเทสมาทดสอบประสิทธิภาพของโมเดลว่าดีไหม เมื่อเราพอใจกับผลที่ได้แล้ว ก็สามารถบันทึกโมเดลเพื่อนำไปใช้กับระบบได้

3.4.7 ส่วนแสดงผล

ในส่วนนี้จะแสดงผลด้วย Web Application ที่พัฒนาโดยการใช้ Flask และBootstrap เป็น Web Application ที่เรียกใช้งานแบบ localhost โดยการใช้งานหลัก ๆ มีดังนี้

3.4.7.1 หน้าเริ่มต้นการทำงาน

เป็นหน้าเริ่มต้นเมื่อทำการเปิด Web Application ก่อนที่จะเริ่มดักจับข้อมูลและนำมาวิเคราะห์

รูป 3.16 หน้าต่างเริ่มต้นของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

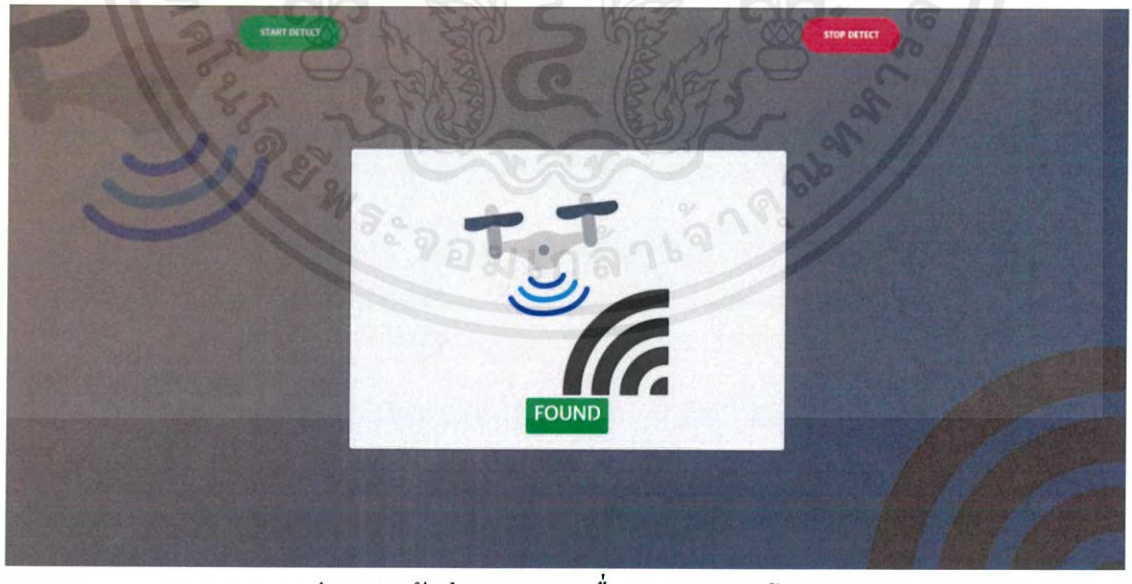
3.4.7.2 หน้าต่างเมื่อเริ่มทำการดักจับข้อมูลและวิเคราะห์

โดยหน้านี้จะทำงานก็ต่อเมื่อมีการคลิกที่ปุ่ม Start Detect มุมบนซ้ายมือ เมื่อคลิกจะทำการเรียก API เพื่อทำการดักจับและวิเคราะห์ข้อมูล

รูป 3.17 หน้าต่างเมื่อคลิกปุ่ม Start Detect

3.4.7.3 หน้าต่างแสดงผลเมื่อมีการตรวจพบ

หน้านี้จะหน้าที่แสดงผลลัพธ์เมื่อตรวจพบเจอโดรนที่บินเข้ามาในพื้นที่



รูป 3.18 หน้าต่างแสดงผลเมื่อตรวจพบเจอโดรน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.7.4 หน้าต่างเมื่อหยุดดักจับข้อมูล

เป็นหน้าที่แสดงผลเมื่อทำการคลิกปุ่ม Stop Detect มุมบนขวามือ เมื่อคลิกจะทำการเรียก API เพื่อหยุดการดักจับและวิเคราะห์ข้อมูล

รูป 3.19 หน้าต่างเมื่อคลิกปุ่ม Stop Detect

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการดำเนินงาน

4.1 การทดลอง

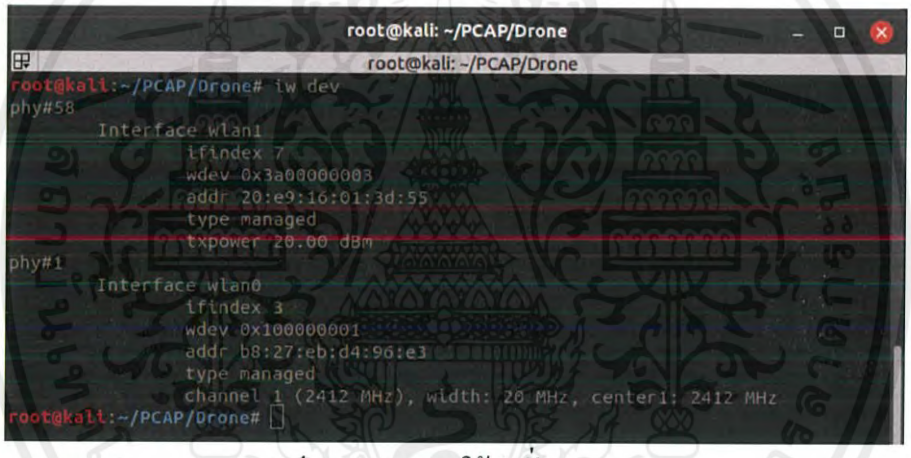
4.1.1 ทดลองการทำงานของ Monitor Mode บน Raspberry Pi 3

4.1.1.1 วัตถุประสงค์

ทดสอบการทำงานของ Monitor Mode บนอินเตอร์เฟสไวไฟที่อยู่ใน Raspberry Pi 3

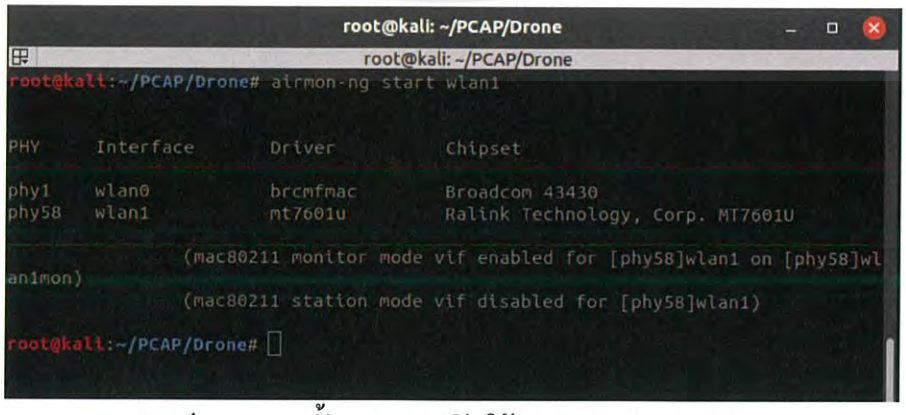
4.1.1.2 วิธีการทดลอง

1) ป้อนคำสั่ง “iw dev” เพื่อเช็คดูอุปกรณ์ Wireless ในเครื่อง



รูป 4.1 แสดงการใช้คำสั่ง iw dev

2) ทำการตั้งค่าอุปกรณ์ Wireless ของ Raspberry Pi 3 ให้เปลี่ยนจาก Managed Mode ให้เป็น Monitor Mode โดยการใช้คำสั่ง airmon-ng start “interface”



รูป 4.2 แสดงขั้นตอนการเปิดใช้งาน Monitor Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.3 ผลการทดลอง

Raspberry Pi 3 สามารถเปิดใช้งาน Monitor Mode ได้

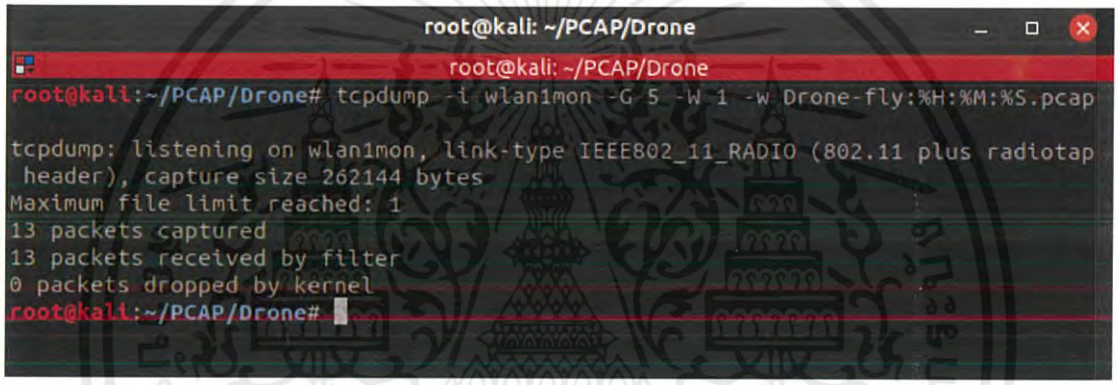
4.1.2 ทดลองดักจับสัญญาณของโดรนบนเครือข่ายไร้สาย

4.1.2.1 วัตถุประสงค์

เพื่อทดลองดักจับสัญญาณที่มีการสื่อสารกับตัวควบคุมและดูการรับส่งข้อมูลของโดรนได้หรือไม่

4.1.2.2 วิธีการทดลอง

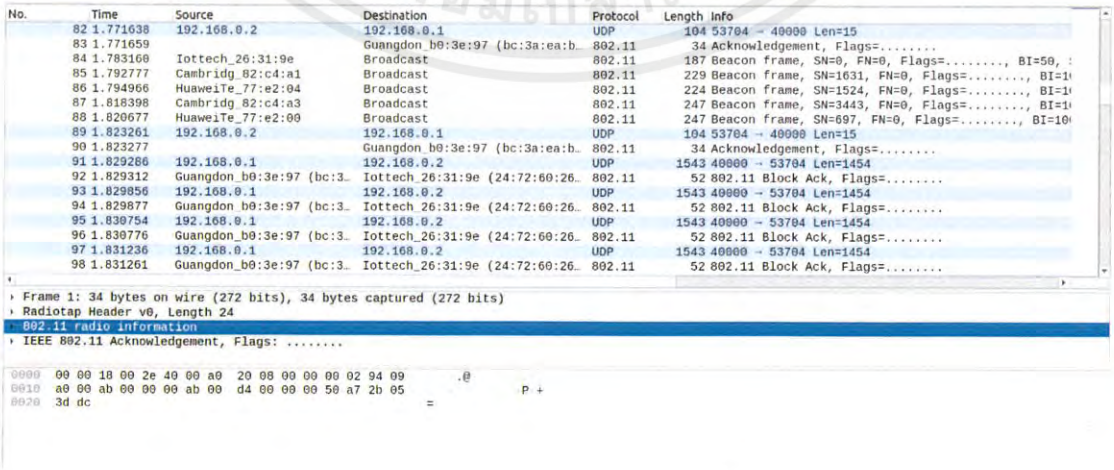
ทำการใช้คำสั่ง tcpdump ซึ่งเป็นคำสั่งในการดักจับสัญญาณผ่านหน้า command line โดยมีการกำหนดช่วงเวลาในการดักจับประมาณ 5 วินาที หลังจากดักจับเสร็จ ก็จะทำการบันทึกไฟล์ที่ดักจับมาได้



รูป 4.3 การใช้คำสั่ง tcpdump

4.1.2.3 ผลการทดลอง

Raspberry Pi 3 สามารถดักจับแพ็คเก็ตที่มีข้อมูลของการรับส่งข้อมูลระหว่างโดรนกับอุปกรณ์ควบคุมได้



รูป 4.4 แสดงข้อมูลแพ็คเก็ตของโดรนที่จับมาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 ทดสอบประสิทธิภาพของโมเดลในการตรวจจับโดรน

4.1.3.1 วัตถุประสงค์

เพื่อวัดประสิทธิภาพความแม่นยำของอัลกอริทึมในการตรวจจับโดรน

4.1.3.2 วิธีการทดลอง

1) ทำการเก็บรวบรวม dataset การบินของโดรน โดยทำการแบ่งเป็นสามระยะคือ ระยะใกล้ (0 - 10 เมตร) ระยะกลาง (20 - 30 เมตร) และระยะไกล (40 - 50 เมตร)
2) นำ dataset ที่ได้มาทำการเทรนโดยใช้โมเดลตรวจจับโดรนที่เราได้บันทึกไว้
3) วิเคราะห์และบันทึกผล

4.1.3.3 ผลการทดลอง

ประสิทธิภาพของโมเดลในการตรวจจับโดรนมีความแม่นยำอยู่ที่ประมาณ 97% - 99% ตามตาราง 4.4 โดยที่ประสิทธิภาพความแม่นยำจะลดลงตามระยะห่างระหว่างอุปกรณ์ที่ใช้ตรวจจับกับโดรน

ตาราง 4.1 ค่า confusion matrix การตรวจจับโดรนในระยะใกล้ (0 - 10 เมตร)

Actual\Predicted	Background	Drone
Background	293	3
Drone	0	20

ตาราง 4.2 ค่า confusion matrix การตรวจจับโดรนในระยะกลาง (20 - 30 เมตร)

Actual\Predicted	Background	Drone
Background	276	0
Drone	4	16

ตาราง 4.3 ค่า confusion matrix การตรวจจับโดรนในระยะไกล (40 - 50 เมตร)

Actual\Predicted	Background	Drone
Background	268	0
Drone	7	10

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 4.4 การเปรียบเทียบโมเดลตรวจจับโดรนในระยะทางต่าง ๆ

Scenario	Accuracy	Class Drone F1-Score
ระยะใกล้ (0 - 10 เมตร)	99.05%	93.05%
ระยะกลาง (20 - 30 เมตร)	98.65%	88.89%
ระยะไกล (40 - 50 เมตร)	97.54%	74.21%

4.1.4 ทดลองตรวจจับการมีอยู่ของโดรนในระยะทางต่าง ๆ

4.1.4.1 วัตถุประสงค์

เพื่อทดสอบอัลกอริทึมว่าสามารถตรวจจับโดรนในระยะทางต่าง ๆ ได้หรือไม่

4.1.4.2 วิธีการทดลอง

1) ตั้งค่าให้อินเตอร์เฟสของ Raspberry Pi 3 ให้ทำงานเป็น Monitor Mode ดังรูป 4.2
2) ทำการรันโปรแกรมเพื่อใช้งานโมเดลตรวจจับและลองเคลื่อนย้ายไปตามระยะทางที่กำหนดไว้ โดยจะทำการดักจับ 10 ครั้งต่อระยะทางที่กำหนด
3) จดบันทึกผล

4.1.4.3 ผลการทดลอง

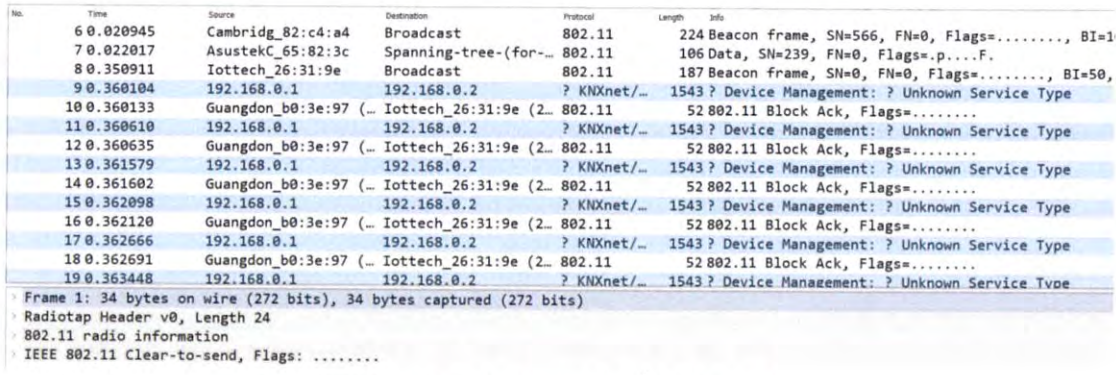
ระบบสามารถตรวจจับโดรนจากโมเดลที่สร้างขึ้นมาได้ โดยมีการเปรียบเทียบกับข้อมูลจริงแล้วว่ามีการรับส่งกันระหว่างโดรนและอุปกรณ์ควบคุม

ตาราง 4.5 แสดงผลลัพธ์ที่ได้จากการตรวจจับ

ระยะทาง/ครั้งที่	1	2	3	4	5	6	7	8	9	10
1 เมตร	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2 เมตร	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5 เมตร	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
15 เมตร	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

ตรวจจับได้ (✓) ตรวจจับไม่ได้ (✕)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



No.	Time	Source	Destination	Protocol	Length	Info
6	0.020945	Cambridg_82:c4:a4	Broadcast	802.11	224	Beacon frame, SN=566, FN=0, Flags=........, BI=1
7	0.022017	AsustekC_65:82:3c	Spanning-tree-(for-...	802.11	106	Data, SN=239, FN=0, Flags=.p....F.
8	0.350911	Iottech_26:31:9e	Broadcast	802.11	187	Beacon frame, SN=0, FN=0, Flags=........, BI=50,
9	0.360104	192.168.0.1	192.168.0.2	? KNXnet/...	1543	? Device Management: ? Unknown Service Type
10	0.360133	Guangdon_b0:3e:97 (...	Iottech_26:31:9e (2...	802.11	52	802.11 Block Ack, Flags=........
11	0.360610	192.168.0.1	192.168.0.2	? KNXnet/...	1543	? Device Management: ? Unknown Service Type
12	0.360635	Guangdon_b0:3e:97 (...	Iottech_26:31:9e (2...	802.11	52	802.11 Block Ack, Flags=........
13	0.361579	192.168.0.1	192.168.0.2	? KNXnet/...	1543	? Device Management: ? Unknown Service Type
14	0.361602	Guangdon_b0:3e:97 (...	Iottech_26:31:9e (2...	802.11	52	802.11 Block Ack, Flags=........
15	0.362098	192.168.0.1	192.168.0.2	? KNXnet/...	1543	? Device Management: ? Unknown Service Type
16	0.362120	Guangdon_b0:3e:97 (...	Iottech_26:31:9e (2...	802.11	52	802.11 Block Ack, Flags=........
17	0.362666	192.168.0.1	192.168.0.2	? KNXnet/...	1543	? Device Management: ? Unknown Service Type
18	0.362691	Guangdon_b0:3e:97 (...	Iottech_26:31:9e (2...	802.11	52	802.11 Block Ack, Flags=........
19	0.363448	192.168.0.1	192.168.0.2	? KNXnet/...	1543	? Device Management: ? Unknown Service Type

รูป 4.5 ตัวอย่างข้อมูลจริงที่ดักจับมาได้

4.1.5 ทดลองระบบแจ้งเตือนผู้ใช้

4.1.5.1 วัตถุประสงค์

เพื่อทดสอบระบบแจ้งเตือนสำหรับผู้ใช้งานเมื่อมีโดรนบินเข้าในพื้นที่ผ่าน Web Application ดังรูป 3.16

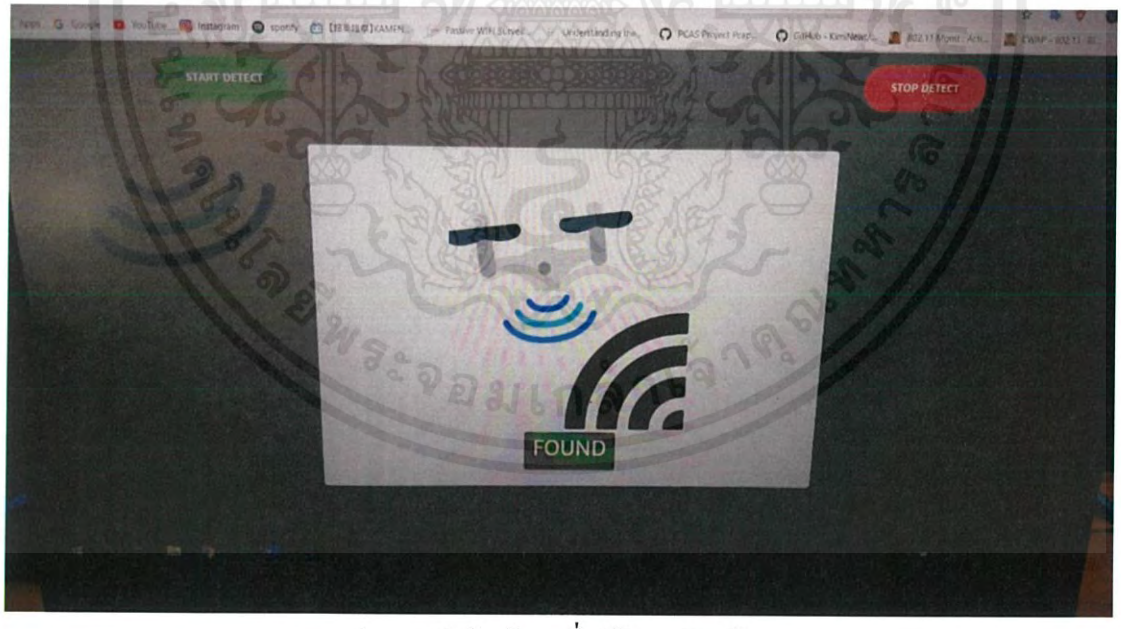
4.1.5.2 วิธีการทดลอง

1) ทำการนำคอมพิวเตอร์เชื่อมต่อกับ Raspberry Pi 3 ในวงเครือข่ายเดียวกัน
2) เข้า Web Application ผ่าน localhost:8000
3) ทดลองปุ่ม Start Detect เพื่อเริ่มระบบ
4) นำโดรนเข้ามาในบริเวณใกล้กับ Raspberry Pi 3 เพื่อทดสอบผลลัพธ์ที่แสดงที่หน้าจอ

รูป 4.6 คอมพิวเตอร์เชื่อมต่อกับ Raspberry Pi 3

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 4.7 โดรนที่ใช้ในการทดลอง Skyhunter X8



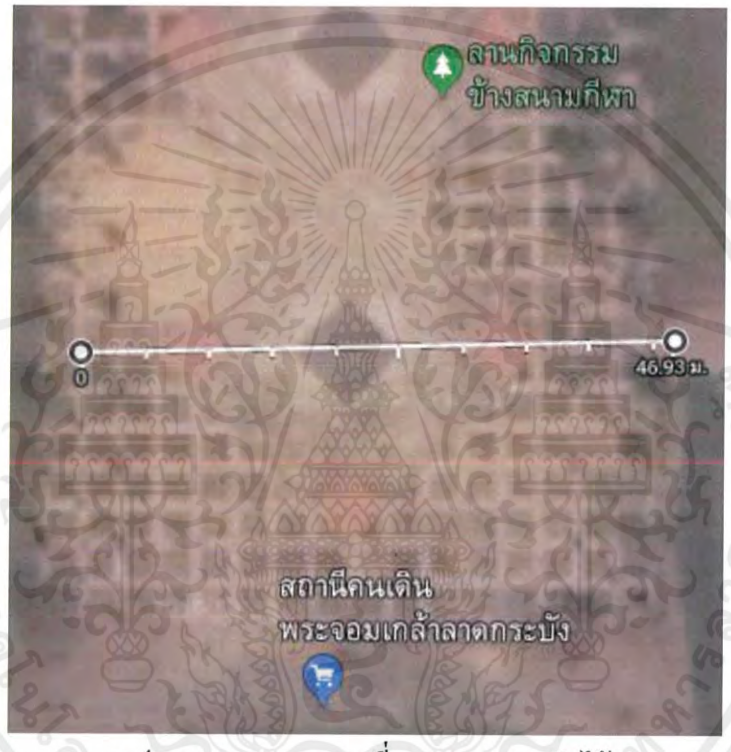
รูป 4.8 รูปแจ้งเตือนเมื่อมีโดรนบินเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 สรุปผลการทดลอง

จากผลการทดลองที่ผ่านมาทำให้สามารถสรุปได้ดังนี้

1) ระบบสามารถตรวจจับการมีอยู่ของโดรนในรัศมี 1-15 เมตรได้ และมีรัศมีการตรวจจับโดรนได้สูงสุดประมาณ 50 เมตร ดังรูป 4.9
2) ระบบสามารถแยกแยะข้อมูลการสื่อสารของโดรนกับอุปกรณ์อื่น ๆ บนเครือข่ายไร้สายได้
3) สามาถแจ้งเตือนผลลัพธ์ผ่าน Web Application เมื่อมีโดรนเข้ามาในพื้นที่ได้ โดยที่ผลการแจ้งเตือนจะมีความหน่วงประมาณ 7-10 วินาที



รูป 4.9 แสดงระยะทางที่สามารถตรวจจับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

5.1.1 ส่วนประมวลผล Raspberry Pi 3

ระบบสามารถตรวจจับโดรนได้ด้วยการวิเคราะห์หาฟีงเกอร์พริ้นท์ แล้วนำฟีงเกอร์พริ้นท์ที่ได้ไปประมวลผลบน Raspberry Pi 3 โดยใช้โมเดลที่สร้างขึ้นมาทำนายผล และสามารถส่งข้อมูลไปให้ Web Application ได้

5.1.2 ส่วนติดต่อผู้ใช้งาน

โครงงานนี้มีส่วนแสดงผลลัพธ์สำหรับการแจ้งเตือนเมื่อมีโดรนเข้ามาในพื้นที่ พัฒนาด้วย Flask และ Bootstrap โดยเมื่อได้รับผลลัพธ์จากโมเดลแล้ว สามารถรับและนำผลลัพธ์นั้นมาแสดงผลได้

5.2 ปัญหาและอุปสรรค

การศึกษาขั้นตอนหรือตอนอัลกอริทึมในการตรวจจับโดรนด้วยวิธีนี้เป็นไปอย่างล่าช้า เนื่องจากมีโดรนที่ใช้พัฒนาเพียงหนึ่งตัวจึงไม่สามารถทดสอบความแม่นยำของโมเดลกับโดรนตัวอื่นได้ การประมวลผลบน Raspberry Pi 3 มีความล่าช้าส่งผลให้ตรวจจับแพ็กเก็ตได้น้อยหรืออาจตรวจจับไม่ได้เลย จำเป็นต้องกำหนดเวลาในการดักจับแพ็กเก็ตอย่างน้อย 3 – 5 วินาที เพื่อให้ Raspberry Pi 3 ประมวลผลแพ็กเก็ตได้ทัน

5.3 แนวทางการพัฒนาต่อ

5.3.1 ส่วนประมวลผล Raspberry Pi 3

เพิ่มประสิทธิภาพและปรับปรุงโมเดลที่ใช้ในการตรวจจับโดรนให้มีการประมวลผลให้เร็วขึ้นเพื่อให้มีผลเป็น Real Time มากขึ้นและสามารถตรวจจับโดรนรุ่นอื่นได้

5.3.2 ส่วนแสดงผล

ปรับปรุงส่วนหน้าแสดงผลให้มีการใช้งานที่หลากหลายมากขึ้นเช่น การแสดงผลข้อมูลที่รับมา และสร้างความปลอดภัยให้กับระบบมากกว่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Phuc Nguyen, Mahesh Ravindranathan, Anh Nguyen, Richard Han and Tam Vu. 2016. "Investigating Cost-effective RF-basedDetection of Drones." Ph.D.Thesis OF University of Colorado, Denver and University of Colorado, Boulder

Waylon Dustin Scheller, 2017. "Detecting drones using machine learning." Ph.D.Thesis and Dissertation, Iowa State University Capstones

I. Bisio, C. Garibotto, F. Lavagetto, A. Sciarrone, and S. Zappatore. "Unauthorized amateur UAV detection based on WiFi statistical fingerprint analysis", **IEEE Communications Magazine**, vol. 56, no. 4, p. 106–111, APRIL 2018.

Raspberry Pi Foundation. 2014. **WHAT IS A RASPBERRY PI?.** [Online]. Available: https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/

Raspberry Pi Foundation. 2015. **ABOUT US.** [Online]. Available: https://www.raspberrypi.org/about/

Raspberry Pi Foundation. 2016. **RASPBERRY PI 3 MODEL B.** [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

Cypress. 2017. **BCM43438 Datasheet.** [Online]. Available: https://www.alldatasheet.com/datasheet-pdf/pdf/1018493/CYPRESS/BCM43438.html

ThaiEasyElec. 2014. **บทความการพัฒนาโปรแกรมบน Raspberry Pi ด้วย Qt.** [Online]. Available: https://thaieasyelec.com/article-wiki/embedded-electronics-application/บทความการพัฒนาโปรแกรมบน-raspberry-pi-ด้วย-qt.html

Microsoft. 2009. **How 802.11 Wireless Works.** [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc757419(v=ws.10)

Fraida Fund. 2017. **Understanding the 802.11 Wireless LAN MAC frame format.** [Online]. Available: https://witestlab.poly.edu/blog/802-11-wireless-lan-2/

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นาวาโทหญิง อาภรณ์ พลเสน. **บทความพิเศษ: รู้จักกับอากาศยานไร้คนขับหรือยูเอวี (Unmanned Aerial Vehicle: UAV).** [Online]. Available: https://www.dti.or.th/page_bx.php?cid=24&cno=4308

Primer. **Wi-Fi: Overview of the 802.11 Physical Layer and Transmitter Measurements.** [Online]. Available: https://www.cnrood.com/en/media/solutions/Wi-Fi_Overview_of_the_802.11_Physical_Layer.pdf

Mike Richardson. 2015. **Drones: Detect, identify, intercept and hijack.** [Online]. Available: https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2015/december/drones-detect-identify-intercept-and-hijack/

Mark LaFay. **UNDERSTANDING HOW YOUR DRONE IS CONTROLLED.** [Online]. Available: https://www.dummies.com/consumer-electronics/drones/understanding-how-your-drone-is-controlled/

Witchapong Daroontham. 2018. **เจาะลึก Random Forest !!!— Part 2 of "รู้จัก Decision Tree, Random Forest, และ XGBoost!!!".** [online]. Available: https://medium.com/@witchapongdaroontham/เจาะลึก-random-forest-part-2-of-รู้จัก-decision-tree-random-forest-และ-xgboost-79b9f41a1c1c

Fraida Fund. 2017. **Understanding the 802.11 Wireless LAN MAC frame format.** [online]. Available: https://witestlab.poly.edu/blog/802-11-wireless-lan-2/

Mike Kershaw. 2016. **Kismet Wireless.** [Online]. Available: https://www.kismetwireless.net/documentation.shtml

Dale Rapp. 2014. **THE TO DS AND FROM DS FIELDS.** [online]. Available: https://dalewifisec.wordpress.com/2014/05/17/the-to-ds-and-from-ds-fields/

Armin Ronacher. **Welcome to Flask.** [online]. Available: http://flask.pocoo.org/docs/1.0/

Wannaphong. 2016. **เริ่มต้นการทำ Machine Learning ด้วย Scikit-learn.** [online]. Available: https://python3.wannaphong.com/2016/01/machine-learning-scikit-learn.html

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Saixiii. 2017. **Flask คืออะไร Python Flask คือ framwork สำหรับทำ web application.** [online]. Available: https://saixiii.com/python-flask-web-application/

Suranart Niamcome. 2013. **Bootstrap คืออะไร + สอนวิธีใช้แบบเข้าใจง่าย**. [online]. Available: http://www.siamhtml.com/bootstrap-คืออะไร-สอนวิธีใช้/

Matana Wiboonyasake. **Machine Learning คืออะไร?.** [online]. Available: https://www.aware.co.th/machine-learning-คืออะไร/

Nuttavut Thongjor. 2016. **[Machine Learning#1] Machine Learning คืออะไร? รู้จักประเภทต่างๆของ Machine Learning**. [online]. Available: https://www.babelcoder.com/blog/posts/machine-learning-introduction

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้