

อภินันทนาการ

สัญญาเลขที่ R2562B02



สำนักหอสมุด

รายงานวิจัยฉบับสมบูรณ์

การวิเคราะห์รูปแบบผนบบพื้นฐานข้อมูลภูมิสารสนเทศเดาร์ตราจอากาศ
ภาคพื้นดินรายละเอียดสูงในลุ่มน้ำยมและน่าน

คณะผู้วิจัย

- | | |
|-----------------------------|-------------------|
| 1. ผศ.ดร.นัฐพล มหาวิค | มหาวิทยาลัยนเรศวร |
| 2. รศ.ดร.ครินทิพย์ แทนราษฎร | มหาวิทยาลัยนเรศวร |
| 3. นายประพุต ยอดไพบูลย์ | กรมอุตุนิยมวิทยา |

| สำนักหอสมุด มหาวิทยาลัยนเรศวร | |
|-------------------------------|-------------|
| วันลงทะเบียน..... | 22 มีค 2565 |
| เลขทะเบียน..... | 1049843 |
| เลขเรียกหนังสือ..... | G 70 |

• 215
• 75
163921
2562

สนับสนุนโดย

งบประมาณแผ่นดินมหาวิทยาลัยนเรศวร
ประจำปี พ.ศ. 2562

กิตติกรรมประกาศ

(Acknowledgement)

โครงการการวิเคราะห์รูปแบบพื้นฐานข้อมูลภูมิสารสนเทศเดкарตตรวจสอบภาคพื้นดินรายละเอียดสูงในส่วนน้ำยมและน่าน ได้รับทุนสนับสนุนจากสำนักงานการวิจัยแห่งชาติ (วช.) ผ่านการจัดสรรงบประมาณรายจ่ายนักวิจัย รหัสโครงการ R2561B031 โดยสามารถพิมพ์ผลงานวิจัยในวารสารที่อยู่ในฐานข้อมูลระดับนานาชาติด้วย 1 ฉบับ ผู้วิจัยขอบขอคุณสำนักงานการวิจัยแห่งชาติและมหาวิทยาลัยเนเรศวรที่พิจารณาจัดสรรทุนในการศึกษาครั้งนี้ กรมอุตุนิยมวิทยาที่อนุเคราะห์ข้อมูล เรเดาร์ สาขาวิชาภูมิศาสตร์ และภาควิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม คณะเกษตรศาสตร์ ธรรมชาติและสิ่งแวดล้อม ที่ให้การสนับสนุนทั้งเรื่องเวลา สถานที่และเครื่องมือในการทำวิจัยเป็นอย่างดี

นัฐพล มหาวิค

คำนำ

โครงการการวิเคราะห์รูปแบบพื้นที่ฐานข้อมูลภูมิสารสนเทศเดкарตตรวจอากาศภาคพื้นดินรายละเอียดสูงในคุณน้ำยมและนาน เป็นโครงการต้นแบบในการพัฒนาระบวนการโมเดลข้อมูล เดкарตด้วยซอฟต์แวร์รหัสเปิด PyQGIS ที่ไม่มีต้นทุนในเรื่องงบประมาณในการจัดซื้อซอฟต์แวร์ทางด้านภูมิสารสนเทศ GIS เนื่องจากมีราคาสูงหลายแสนบาทซึ่งอาจไม่เหมาะสมกับองค์กรของรัฐที่มีงบประมาณจำกัดในหลายหน่วยงาน ในงานวิจัยครั้งนี้ทุกรอบวนการได้พัฒนาจากภาษาไพธอนที่พัฒนาร่วมกับไลบรารีของ QGIS ซึ่งนอกจากจะได้องค์ความรู้ของกระบวนการที่พัฒนาขึ้นมาแล้ว ยังสามารถนำโค้ดที่ได้พัฒนาไปใช้ในการเรียนการสอนในระดับบัณฑิตศึกษารวมทั้งสามารถนำไปสร้างเป็นแบบเรียนให้กับนิสิตที่สนใจในรายวิชาภูมิอากาศวิทยาและบรรยายภาษาอังกฤษได้อีกด้วย อันจะเป็นการสร้างบุคลากรของประเทศไทยให้มีองค์ความรู้ที่พร้อมนำไปปรับใช้กับหน่วยงานจริงของประเทศไทย หลังจากได้จากการศึกษาแล้ว นอกจากนั้นงานวิจัยครั้งนี้ยังได้มีการส่งผ่านโค้ดที่ได้พัฒนาจริงให้กับหน่วยงานกรมอุตุนิยมวิทยาร่วมกับให้คำปรึกษาในเรื่องการใช้งาน งานวิจัยครั้งนี้ถือเป็นครั้งแรกของประเทศไทยเท่าที่ทราบว่ามีการนำข้อมูลเดкарตมาโมเดลกันเป็นผลิตภัณฑ์ฝนคุณน้ำซึ่งสามารถนำไปปรับใช้กับการติดตามสถานการณ์จริงในคุณน้ำอื่นๆ ของประเทศไทยแบบระบบการติดตามใกล้เวลาจริง ซึ่งข้อมูลที่ผลิตได้จากโค้ดต้นแบบนี้จะช่วยในกระบวนการเตือนภัยที่มาพร้อมกันกับช่วงมรสุมของประเทศไทย เพื่อลดภัยพิบัติที่ส่งผลกระทบต่อประชาชนในพื้นที่ประจำบ้าน

นรรพล มหาวิค

บทคัดย่อ

ข้อมูลฝนประมาณค่าจากเรดาร์ตรวจอากาศภาคพื้นดินในระดับลุ่มน้ำของประเทศไทยนั้นยังไม่มีอย่างเป็นทางการ รวมถึงโค้ดต้นแบบที่ใช้ในการประมวลผลข้อมูลเรดาร์ที่เผยแพร่ในเชิงสาธารณะนั้นยังไม่มีในประเทศไทย การทำให้ทั้งข้อมูลฝนประมาณค่าและโค้ดต้นแบบเกิดขึ้นในประเทศไทยจึงเป็นงานที่ท้าทายในครั้งแรกนี้ เพื่อให้เกิดการระบบเตือนภัยแบบใกล้เวลาจริงในพื้นที่ระดับลุ่มน้ำอันจะนำมาสู่การลดผลกระทบต่อประชาชนในพื้นที่ประจำของประเทศไทย ผู้วิจัยจึงมีความเห็นว่าควรจะต้องพัฒนาโค้ดต้นแบบที่ประมวลผลข้อมูลเรดาร์ด้วยการเขียนต่อข้อมูลหลายสถานีที่ครอบคลุมพื้นที่ลุ่มน้ำได้ฯ ของประเทศไทย ให้สามารถประมวลผลข้อมูลแบบหลายห้วงเวลาและสามารถสร้างผลิตภัณฑ์เชิงพื้นที่ในรูปของแผนที่ที่มีฟอร์แมทที่สามารถส่งต่อให้กับวิจัย นักวิชาการหรือผู้ที่สนใจนำไปวิเคราะห์ หรือสร้างแบบจำลองเรื่องที่เกี่ยวข้องในช่วงระหว่างพายุรุ่มของประเทศไทย ดังนั้นงานวิจัยครั้งนี้จึงได้มีการพัฒนาระบวนการโมเสคเรดาร์ในพื้นที่ลุ่มน้ำโดยใช้ระบบภูมิสารสนเทศ GIS ในระบบซอฟท์แวร์ที่สเปิดที่ชื่อว่า PyQGIS ซึ่งเป็นไลบรารีของโปรแกรม GIMP ที่ชื่อ QGIS ในพื้นที่ลุ่มน้ำยมและน่าน ในช่วงพายุเซญติญาปี พ.ศ. 2561 โดยทำการพัฒนาระบวนการตั้งแต่การนำเข้าข้อมูล การตรวจหาค่า radar flare การวิเคราะห์การบดบังของลำบีมด้วยภูมิประเทศ การโมเสคเรดาร์ของกรมอุตุนิยมวิทยา การแปลงค่าฝนจากค่าการสะท้อนไปสู่ค่าฝนประมาณค่าด้วยความสัมพันธ์ Z-R สามแบบ รวมถึงการตรวจสอบความถูกต้องของการประมาณค่าด้วยข้อมูลฝนภาคพื้นดินของกรมอุตุนิยมวิทยาที่พบว่าการใช้ความสัมพันธ์ของ Rosenfeld Tropical Z-R ให้การประมาณค่าฝนที่ดีในช่วงพายุเซญติญา จนถึงการพัฒนาการตัดขอบเขตแผนที่ผลิตภัณฑ์ฝนประมาณค่าให้พอดีกับพื้นที่ลุ่มน้ำโดยทำการส่งออกเป็นฟอร์แมท Geotiff ที่สามารถนำไปใช้ต่อในระบบ GIS หรือ แบบจำลองอุทกศาสตร์ได้ โค้ดต้นแบบที่ได้นี้ดูถูกส่งต่อให้กับหน่วยงานกรมอุตุนิยมวิทยาเพื่อใช้เป็นส่วนหนึ่งในงานภาคปฏิบัติและทำการปรับปรุงในเวอร์ชันต่อไปของการพัฒนาโค้ดบนฐานความรู้ภูมิสารสนเทศเดкарตตรวจอากาศ ผู้สนใจสามารถนำไปใช้ต่อที่คณะผู้วิจัยได้พัฒนาที่มีในส่วนภาคผนวกของเล่มรายงานนี้ไปศึกษาและพัฒนาต่อยอดได้อย่างเสรี

นัฐพล มหาวิค

สารบัญ

| | หน้า |
|--|------|
| กิตติกรรมประกาศ | ก |
| คำนำ | ข |
| บทคัดย่อ | ค |
| สารบัญ | ง |
| สารบัญรูป | จ |
| บทที่ 1 บทนำ | 1-1 |
| 1.1 ที่มาและความสำคัญของปัญหางานวิจัย | 1-1 |
| 1.2 วัตถุประสงค์ของโครงการวิจัย | 1-2 |
| 1.3 ขอบเขตของโครงการวิจัย | 1-2 |
| 1.4 ระเบียบวิธีวิจัย | 1-3 |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับ | 1-4 |
| 1.6 ระยะเวลาดำเนินการวิจัย | 1-5 |
| บทที่ 2 วรรณกรรม/สารสนเทศ (Information) ที่เกี่ยวข้อง | |
| 2.1 เคราร์ตตรวจสอบภาคพื้นดิน | 2-1 |
| 2.2 งานวิจัยเด darüber ที่เกี่ยวข้อง | 2-4 |
| 2.3 แผนที่ไม่เสียข้อมูลเรดาร์ตรวจสอบภาค | 2-7 |
| บทที่ 3 วิธีดำเนินการวิจัย | |
| 3.1 พื้นที่ศึกษา | 3-1 |
| 3.2 เทศบาลพาย | 3-3 |
| 3.3 ข้อมูลลุมน้ำ | 3-4 |
| 3.4 ข้อมูล | 3-5 |
| 3.5 เครื่องมือ ซอฟท์แวร์ | 3-15 |
| 3.6 วิธีการ | 3-22 |
| บทที่ 4 ผลการศึกษา | |
| 4.1 การพัฒนาโปรแกรมเพื่อเตรียมข้อมูลเรดาร์ก่อนการประมาณผล | 4-1 |
| 4.2 การพัฒนากระบวนการตรวจสอบจาก radar flare | 4-25 |
| 4.3 การวิเคราะห์การบดบังข้อมูลจากลักษณะภูมิประเทศ | 4-6 |
| 4.4 การพัฒนาโค้ด PyQGIS เพื่อการสร้างแผนที่ไม่เสียเรดาร์ในอุ่มน้ำ | 4-10 |
| 4.5 การตรวจสอบผลการประมาณค่าฝนเรดาร์ด้วยผนที่วัดได้จากสถานีวัดฝนภาคพื้นดิน | 4-20 |
| บทที่ 5 สรุปผลการศึกษาและข้อเสนอแนะ | |
| 5.1 สรุปผลการศึกษา | 5-1 |
| 5.2 ข้อเสนอแนะ | 5-2 |
| เอกสารอ้างอิง | 6-1 |
| ภาคผนวก ก, ข, ค, ง, จ, ฉ, ช | |

สารบัญภาพ

| | หน้า |
|---|------|
| ภาพที่ 1.1 ภาพที่ 1.1 สถานีตรวจสอบภาคพื้นดินสถานีเรดาร์พิษณุโลก | 1-3 |
| ภาพที่ 2.1 การเก็บข้อมูลของเรดาร์ตรวจสอบภาคแบบหลายมุมยก | 2-1 |
| ภาพที่ 2.2 ฝนตกเข้าที่เกิดจากการเมืองออกเข้าทางทิศทางการเคลื่อนที่ของกระแสลม | 2-4 |
| ภาพที่ 2.3 สถานีเรดาร์ที่เก็บข้อมูลทั่วภูมิภาคอินโดจีน | 2-7 |
| ภาพที่ 3.1 พื้นที่ศึกษาลุ่มน้ำแม่น้ำยม | 3-1 |
| ภาพที่ 3.2 พื้นที่ศึกษาลุ่มน้ำแม่น้ำน่าน | 3-2 |
| ภาพที่ 3.3 การเคลื่อนที่ของพายุเช็นติบูมทิศทางเข้าทางทิศเหนือของประเทศไทย | 3-3 |
| ภาพที่ 3.4 สภาพลำน้ำยมที่แห้งเหตุผลในช่วงฤดูฝน | 3-4 |
| ภาพที่ 3.5 สถานีเรดาร์ตรวจสอบภาคของกรมอุตุนิยมวิทยาจังหวัดเชียงราย | 3-6 |
| ภาพที่ 3.6 สถานีเรดาร์ตรวจสอบภาคจังหวัดพิษณุโลกและอุปราช | 3-6 |
| ภาพที่ 3.7 การแลกเปลี่ยนข้อมูลและการเยี่ยมชมอุปกรณ์สถานีเรดาร์ตรวจสอบภาค | 3-6 |
| ภาพที่ 3.8 สถานีเรดาร์ตรวจสอบภาค่น่าน อำเภอท่าวังผา | 3-7 |
| ภาพที่ 3.9 ระบบ雷达ตรวจสอบภาคสถานีท่าวังผา จังหวัดน่าน | 3-7 |
| ภาพที่ 3.10 สถานีเรดาร์ตรวจสอบภาคฝนหลวง อำเภอร้องกวาง จังหวัดแพร่ | 3-8 |
| ภาพที่ 3.11 แผนที่และภาระเบรเดาร์ฝนหลวง อำเภอร้องกวาง จังหวัดแพร่ | 3-8 |
| ภาพที่ 3.12 ตารางสถานีโทรมาตรทั้ง 43 สถานี | 3-9 |
| ภาพที่ 3.13 สถานีวัดผันโนทรมารตรของกรมอุตุนิยมวิทยา | 3-10 |
| ภาพที่ 3.14 สถานีวัดผันโนทรมารตรของกรมอุตุนิยมวิทยา ติดตั้งที่เทศบาลตำบลไผ่ร่อง | 3-11 |
| ภาพที่ 3.15 สถานีวัดผันโนทรมารตรของกรมชลอุตุนิยมวิทยา ติดตั้งที่อบต.กำแพงดิน | 3-12 |
| ภาพที่ 3.16 สถานีวัดผันโนทรมารตรของกรมอุตุนิยมวิทยา ติดตั้งที่เชิงสะพานพระร่วง | 3-13 |
| ภาพที่ 3.17 เวปด้านโพลเด็กข้อมูล SRTM DEM 3Arc | 3-14 |
| ภาพที่ 3.18 โปรแกรม QGIS โดยทำการรันโปรแกรมที่พัฒนาด้วย PyQGIS ใน python console | 3-15 |
| ภาพที่ 3.19 เวปไซต์สำหรับดาวน์โหลดโปรแกรม WCT | 3-16 |
| ภาพที่ 3.20 การเปิดข้อมูลเรดาร์ตรวจสอบภาคด้วย WCT จากข้อมูล UF | 3-16 |
| ภาพที่ 3.21 โปรแกรม WCT เวอร์ชัน 4.3.1 | 3-17 |
| ภาพที่ 3.22 กระบวนการสำหรับเลือกค่าพารามิเตอร์ใน WCT | 3-18 |
| ภาพที่ 3.23 ตัวแปรที่ต้องการส่งออกจากโปรแกรม WCT | 3-18 |
| ภาพ 3.24 ค่าพิกัดที่ต้องการส่งผลลัพธ์ออกในรูปแบบ shapefile ในโปรแกรม WCT | 3-19 |
| ภาพที่ 3.25 โปรแกรม IDV ในส่วนของ Dashboard | 3-20 |
| ภาพที่ 3.26 โปรแกรม IDV ในส่วน MapView ที่แสดงค่าการสะท้อนของเรดาร์ของสถานีพิษณุโลก | 3-21 |
| ภาพที่ 3.27 ขั้นตอนการดำเนินการวิจัยเพื่อการพัฒนาวิธีการการประมวลผลข้อมูล | 3-22 |
| ภาพที่ 3.28 ค่าการสะท้อนของสถานีเรดาร์พิษณุโลกวันที่ 17 เดือนกรกฎาคม | 3-23 |
| ภาพที่ 3.29 ลักษณะข้อมูลเรดาร์แบบ polar coordinate | 3-23 |
| ภาพที่ 3.30 ค่าแอทริบิวต์ของข้อมูลเรดาร์ | 3-24 |

| | |
|--|------|
| ภาพที่ 3.31 ตัวอย่าง radar flare ของสถานีพิษณุโลก | 3-25 |
| ภาพที่ 3.32 ลักษณะการตรวจตามมุมยก (Elevation) เป็นระนาบทาทางเดียว (Ray) จากแนวรัศมี | 3-26 |
| ภาพที่ 4.1 โค้ดแสดงการหาไฟล์และระบบเส้นโครงแผนที่ให้เป็น UTM โซน 47N | 4-1 |
| ภาพที่ 4.2 ฟังก์ชัน reprojectRad | 4-2 |
| ภาพที่ 4.3 โค้ดการหา flare รอบที่ 1 | 4-3 |
| ภาพที่ 4.4 โค้ดการหา flare รอบที่ 2 | 4-4 |
| ภาพที่ 4.5 ผลการตรวจพบ radar flare ของสถานีพิษณุโลกวันที่ 15 กรกฎาคม พ.ศ.2561 | 4-5 |
| ภาพที่ 4.6 ผลการวิเคราะห์การบดบังของลำบีมด้วยลักษณะภูมิประเทคโนโลยีของสถานีพิษณุโลก | 4-6 |
| ภาพที่ 4.7 การบดบังของลำบีมด้วยภูมิประเทคโนโลยีเรดาร์พิษณุโลกแสดงใน QGIS | 4-7 |
| ภาพที่ 4.8 ผลการวิเคราะห์การบดบังของลำบีมด้วยภูมิประเทคโนโลยีของสถานีเชียงรายและพิษณุโลก | 4-8 |
| ภาพที่ 4.9 โค้ดการหาบริเวณที่ลำบีมของเรดาร์ถูกบดบังด้วยภูมิประเทคโนโลยี | 4-9 |
| ภาพที่ 4.10 ฟังก์ชันหลักที่ใช้ในการโมเสค | 4-10 |
| ภาพที่ 4.11 ฟังก์ชัน clipRadar | 4-11 |
| ภาพที่ 4.12 ฟังก์ชัน calParaGrid | 4-11 |
| ภาพที่ 4.13 ฟังก์ชัน createGrid | 4-12 |
| ภาพที่ 4.14 โค้ดโมเสคส่วนที่ 1 | 4-13 |
| ภาพที่ 4.15 โค้ดโมเสคส่วนที่ 2 | 4-13 |
| ภาพที่ 4.16 แปลงข้อมูลผลลัพธ์การประมาณค่าการสะท้อนในหน่วย dBZ ไปเป็นฝนประมาณค่า | 4-14 |
| ภาพที่ 4.17 ฟังก์ชันในการแปลงเวลาของเรดาร์จาก UTC เป็น localtime | 4-15 |
| ภาพที่ 4.18 ฟังก์ชันในการส่งออกข้อมูลฝนประมาณค่าจากข้อมูลการสะท้อนของเรดาร์ | 4-16 |
| ภาพที่ 4.19 ฝนประมาณค่าจากการโมเสคเรดาร์สถานีเชียงรายและพิษณุโลกในพื้นที่ลุ่มน้ำบ้าน | 4-17 |
| ภาพที่ 4.20 แผนที่โมเสคค่าฝนประมาณค่ารายชั่วโมงจากข้อมูลเรดาร์ในพื้นที่ลุ่มน้ำบ้าน | 4-18 |
| ภาพที่ 4.21 แผนที่โมเสคค่าฝนประมาณค่ารายชั่วโมงจากข้อมูลเรดาร์ในพื้นที่ลุ่มน้ำบ้าน | 4-19 |
| ภาพที่ 4.22 แผนที่แสดงการกระจายตัวของสถานีวัดฝนภาคพื้นดินจำนวน 10 สถานี | 4-20 |
| ภาพที่ 4.23 การตรวจสอบความถูกต้องของการประมาณค่าฝนจากค่าสะท้อนเรดาร์ในแต่ละ Z-R | 4-21 |
| ภาพที่ 4.24 เปรียบเทียบค่า Mean Field Bias ของความสัมพันธ์ Z-R ทั้งสามในแต่ละวัน | 4-21 |
| ภาพที่ 4-25 ผลการตรวจสอบความถูกต้องของการประมาณค่าฝนจากการประมาณค่าเรดาร์ | 4-23 |
| ภาพที่ 4-26 การตรวจสอบความถูกต้องของฝนสะสมราย 24 ชั่วโมงโดยใช้ค่า mean field bias | 4-24 |

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของปัญหางานวิจัย

ความเสียหายจากภาระน้ำท่วมส่งผลกระทบต่อสภาพเศรษฐกิจของประเทศไทยโดยรวม โดยจะเห็นว่าในช่วงทศวรรษนี้และที่ผ่านมา มีความรุนแรงของสภาพอากาศประจำวันกับการใช้ทรัพยากรธรรมชาติและป่าไม้อย่างขาดการอนุรักษ์ในแต่ละพื้นที่ทำให้เกิดผลกระทบที่รุนแรงทั้งทางกายภาพ เศรษฐกิจและสังคม ไม่ว่าจะเป็นน้ำท่วมในปี 2554 ที่พบว่ามูลค่าความเสียหายทางเศรษฐกิจสูงถึง 1.44 ล้านล้านบาท และมีประชาชนผู้ได้รับผลกระทบจำนวน 16,224,302 คน (คณะกรรมการป้องกันและบรรเทาสาธารณภัยแห่งชาติ 2558) บาท รวมไปถึงต้นปี 2560 ซึ่งเป็นภาระผ่อนในช่วงฤดูมรสุมของภาคใต้ทำให้ภาคใต้ฝั่งตะวันออกได้รับความเสียหายส่งผลกระทบต่อสภาพเศรษฐกิจของประเทศไทยโดยรวม นอกจากนั้นปริมาณฝนที่มากจากเดี๋ยรสองซึ่งเกิดจากการอ่อนกำลังลงของพายุเขินการทำให้ภาคตะวันออกเฉียงเหนือและภาคเหนือของประเทศไทยเกิดภาระน้ำท่วมในช่วงปลายเดือนกรกฎาคม พ.ศ. 2560

การเปลี่ยนแปลงภูมิอากาศของโลกส่งผลกระทบต่อระบบการหมุนเวียนของบรรยากาศ ไม่เพียงแต่ การเปลี่ยนแปลงในระดับโลกร่วมถึงในระดับท้องถิ่นก็ได้รับผลกระทบด้วยเช่นกัน (IPCC 2014) มีการคาดการณ์ถึงสภาพอากาศที่ทวีความรุนแรงในอนาคตอันใกล้จากแบบจำลองสภาพภูมิอากาศ ซึ่งรวมไปถึงลักษณะทางกายภาพของปริมาณน้ำฝนในแต่ละภูมิภาค โดยพบว่าในประเทศไทยนั้นจะมีการเพิ่มขึ้นและลดลงของปริมาณฝนเปลี่ยนแปลงไปในแต่ละภูมิภาคอย่าง (Endo et al. 2009) ซึ่งก็ยังไม่มีความแน่นอนในการประมาณค่าอยู่เนื่องมาจากข้อมูลที่ใช้ในการศึกษาเป็นเพียงแค่ตัวแปรที่ไม่ต่อเนื่องกันของสถานีวัดปริมาณน้ำฝน ทำให้ไม่สามารถหาการกระจายตัวเชิงพื้นที่ของปริมาณฝนได้ ซึ่งเป็นข้อมูลที่สำคัญที่ต้องเร่งสร้างฐานความรู้นี้องจากจะเป็นฐานข้อมูลในการตรวจสอบความถูกต้องของแบบจำลองภูมิอากาศวิทยาต่อไป ข้อมูลที่สำคัญนั้นก็คือข้อมูลน้ำฝนประมาณค่าจากเรดาร์ตรวจอากาศพื้นดิน เนื่องจากมีความละเอียดเชิงพื้นและความละเอียดเชิงห้วงเวลาที่สูง ซึ่งเมื่อข้อมูลดังกล่าวมาพิจารณาร่วมกับข้อมูลที่ตรวจฝนสถานีอากาศพื้นดินจะทำให้ได้_fnประมาณค่าที่มีความน่าเชื่อถือมากยิ่งขึ้น เนื่องจากงานวิจัยที่เกี่ยวข้องยังไม่มีการศึกษามากนักจึงเป็นประเด็นที่น่าสนใจในการนำข้อมูลที่มีคุณภาพมาทำการวิจัยเพื่อสร้างองค์ความรู้เชิงพื้นที่เพื่อเป็นข้อมูลสนับสนุนการตัดสินใจการจัดการทรัพยากรน้ำได้อย่างมีประสิทธิภาพในรูปแบบฐานข้อมูลเชิงเลขที่ใช้ข้อมูลได้เปรียบของระบบภูมิสารสนเทศในการจัดการข้อมูล การทำความเข้าใจกับลักษณะฝนในประเทศไทยจำเป็นต้องเข้าใจลักษณะฝนที่เกิดในช่วงก่อนฤดูมรสุมและฤดูฝน ซึ่งจะเป็นผลมาจากการอิทธิพลของลมมรสุมตะวันตกเฉียงใต้เป็นหลักในช่วงฤดูฝน

ของประเทศไทย

ประกอบกับการได้รับปริมาณน้ำฝนเพิ่มขึ้นในช่วงหลังของฤดูฝนซึ่งเป็นความแปรปรวนที่มาจากการเปลี่ยนแปลงทางภูมิศาสตร์ทางอากาศที่สำคัญของประเทศไทย ทั้งสองลุมน้ำมีปัญหาน้ำท่วมเป็นประจำทุกปี เนื่องจากเป็นที่คุ่มต่ำอีกด้วยทั้งมีกรรมการใช้พื้นที่อย่างเข้มข้นเนื่องมาจากเพิ่มของประชากรในอัตราที่สูง เป็นผลให้มีการใช้ทรัพยากรป่าไม้ในพื้นที่ร่องรอยของร่องรอยเป็นปัญหาระดับชาติ ทำให้มีปริมาณน้ำฝนที่มากก็จะเกิดปัญหาน้ำท่วมตามมาซึ่งอาจจะได้รับผลกระทบจากความแปรปรวนของสภาพอากาศเนื่องมาจากการเปลี่ยนแปลงของสภาพภูมิอากาศจะที่ส่งผลกระทบต่อปริมาณน้ำและความรุนแรงของถักน้ำฝนที่เพิ่มขึ้นในอนาคตได้ ดังนั้นการจัดระดับความรุนแรงของรูปแบบพื้นที่ฐานข้อมูลที่มีคุณภาพและหลักการวิเคราะห์เชิงระบบจะช่วยให้เกิดองค์ความรู้ของภัยพันธุ์จะสนับสนุนภารกิจของรัฐต่อไปได้มีประสิทธิภาพยิ่งขึ้น เพื่อเป็นฐานความรู้ในการพัฒนารูปแบบข้อมูลภูมิสารสนเทศฝนที่นำความรู้ทางภัยพันธุ์มาช่วยในการบริหารจัดการน้ำต่อไป

จากเหตุผลที่กล่าวมาข้างต้นจึงมีความจำเป็นอย่างยิ่งในการพัฒนาวิธีการทางภูมิสารสนเทศเพื่อนำข้อมูลเดкарตตรวจอากาศภาคพื้นดินมาวิเคราะห์ฝนเชิงพื้นที่ที่จะทำให้ทราบถึงกลไกและรูปแบบทางภัยพันธุ์ของฝนเชิงพื้นที่และเวลา โดยมีความจำเป็นเร่งด่วนที่จะต้องศึกษาฝนในพื้นที่คุ่มน้ำเพื่อเป็นต้นแบบและทางเลือกให้กับหน่วยงานของรัฐในการนำองค์ความรู้ที่ได้จากการวิจัยครั้งนี้ไปประยุกต์ใช้เพื่อสนับสนุนการตัดสินใจในการบริหารจัดการน้ำต่อไป

1.2 วัตถุประสงค์ของโครงการวิจัย

1. เพื่อพัฒนาวิธีการทางภูมิสารสนเทศเพื่อประเมินค่าฝนเชิงพื้นที่และเวลาจากค่าการสะท้อนเรเดาร์ตรวจอากาศในช่วงฤดูมรสุม
2. เพื่อศึกษาและวิเคราะห์ถึงรูปแบบเชิงพื้นที่ค่าเบี่ยงเบนของฝนประมาณค่าจากค่าการสะท้อนของเรเดาร์ตรวจอากาศที่สัมพันธ์กับค่าฝนที่วัดได้ภาคพื้นที่

1.3 ขอบเขตของโครงการวิจัย

การศึกษารังน้ำมุ่งพัฒนาวิธีการทางภูมิสารสนเทศเพื่อประเมินค่าฝนเชิงพื้นที่และเวลาจากค่าการสะท้อนเรเดาร์ตรวจอากาศในช่วงฤดูมรสุม โดยเรดาล์ตรวจอากาศภาคพื้นดินสถานีอุตุนิยมวิทยาจังหวัดพิษณุโลก เชียงรายและน่าน ทำการตรวจข้อมูลจำนวน 4 ครั้งต่อชั่วโมงโดยกรมอุตุนิยมวิทยาแห่งประเทศไทย (Thai Meteorological Department : TMD) ซึ่งจะพัฒนาแผนที่ฐานข้อมูลค่าการสะท้อนในระดับความสูงในช่วงปี 2561 เพื่อให้ได้วิธีการในโปรแกรมรหัสเปิด โดยใช้ภาษาไพธอน PyQGIS โดยเลือกช่วงเวลาของการเกิดพายุเขิญติณในปี 2561 ในพื้นที่ลุ่มน้ำยมและน่าน



ภาพที่ 1.1 สถานีตรวจอากาศภาคพื้นดินสถานีเรดาร์พิษณุโลก (a) หอดอยเรดาร์ตรวจอากาศ สังกัดกรมอุตุนิยมวิทยา ตั้งอยู่อำเภอเมือง จังหวัดพิษณุโลก เก็บข้อมูลในคลื่นความถี่ C band ทำการวัดข้อมูลทุกๆ 15 นาที (b) ข้อมูลค่าการสะท้อนของเรดาร์ตรวจอากาศ พิษณุโลก

1.4 ระเบียบวิธีวิจัย

1. ทบทวนวรรณกรรมที่เกี่ยวข้อง
2. รวบรวมข้อมูลทางด้านอุตุนิยมวิทยาได้แก่ เรดาร์ตรวจอากาศภาคพื้นดิน ฝนที่วัดได้จากสถานีวัดฝนภาคพื้นดิน ข้อมูลตรวจอากาศ จากสถานีอุตุนิยมวิทยาพิษณุโลก เชียงรายและน่าน
3. รวบรวมข้อมูลฝนที่วัดได้จากสถานีไทรมาตราของกรมอุตุนิยมวิทยา
4. ตรวจสอบความถูกต้องของข้อมูลฝนที่วัดได้จากสถานีภาคพื้นดิน
5. ตรวจสอบความสมบูรณ์ของข้อมูลเรดาร์ตรวจอากาศ
6. ออกแบบวิธีการและกระบวนการเพื่อประมวลผลข้อมูลเรดาร์ตรวจอากาศในระบบภูมิสารสนเทศรหัสเปิด
7. ประมวลผลของข้อมูลค่าการสะท้อนของเรดาร์
8. ทำการพัฒนาระบวนการตรวจสอบค่าฝนที่แปลงค่าการสะท้อนของเรดาร์ตรวจอากาศ
9. ทำการออกแบบที่ภาคสนามเพื่อสำรวจดูพื้นที่ที่ได้รับผลกระทบ
10. อกปรายผลการวิจัยร่วมกับเจ้าหน้าที่กรมอุตุนิยมวิทยา

11. เปิดอบรมเผยแพร่องค์ความรู้และการแลกเปลี่ยนองค์ความรู้ร่วมกันในเรื่องการเฝ้าระวังและเตือนภัยฝนที่รุนแรง ณ มหาวิทยาลัยศรีปทุม
12. เจียนรายงานสรุป
13. เผยแพร่ผลงานวิจัยในรูปแบบการตีพิมพ์วารสารนานาชาติ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

โครงการวิจัยนี้มีเป้าประสงค์เพื่อเป็นต้นแบบในการจัดทำแผนที่เดินทางโดยไม่ต้องออกจากบ้าน ตรวจสอบจากเดินทางตรวจสอบจากภาคพื้นดินในพื้นที่ระดับอุ่มน้ำยมและป่า ซึ่งเป็นพื้นที่สำคัญที่เป็นน้ำดันทุนให้กับแม่น้ำเจ้าพระยาของประเทศไทย การใช้ออฟฟิเวอร์หัสเปิดในการประเมินผล ข้อมูลเดินทางที่มีความละเอียดเชิงพื้นที่และเวลาที่สูงนั้น จะทำให้เป็นต้นแบบในการเผยแพร่และการเรียนรู้ของหน่วยงานรัฐ มหาวิทยาลัย นักวิจัยและประชาชนทั่วไปในการเข้าถึงกระบวนการประเมินผลจากโคลด์ตั้งต้น การใช้ออฟฟิเวอร์หัสเปิดจะทำให้ประหยัดต้นทุนของชาติรวมถึงเปิดโอกาสในการเข้าถึงและเรียนรู้เพื่อพัฒนาเป็นมาตรฐานข้อมูลของผู้ประกอบการที่ในระดับอุ่มน้ำของชาติ

- 1 การผลิตบทความวิจัยจำนวน 1 บทความเพื่อตีพิมพ์ในวารสารวิชาการนานาชาติ
- 2 การพัฒนานักวิจัยรุ่นใหม่ โดยโครงการนี้มีผู้เชี่ยวชาญร่องอุทก-ภูมิภาควิทยาเป็นนักวิจัยพี่เลี้ยงซึ่งมีประสบการณ์ในการทำงานด้านการวิเคราะห์ข้อมูลน้ำฝนเชิงพื้นที่มาอย่างเชี่ยวชาญ และยังมีการประสานงานร่วมวิจัยกับนักวิจัยในกรมอุตุนิยมวิทยาที่จะคอยให้คำแนะนำปรึกษาได้ ตรงตามความต้องการของหน่วยงาน
- 3 การพัฒนานิสิตทั้งระดับปริญญาตรีและโทในฐานะผู้ช่วยนักวิจัย เพื่อเรียนรู้กระบวนการวิจัยด้านการประเมินผลข้อมูลเดินทางตรวจสอบจากภาคพื้นดินเพื่อจัดทำเป็นฐานข้อมูลเชิงพื้นที่อย่างเป็นระบบ
- 4 การส่งเคราะห์งานวิจัยให้เป็นกรณีศึกษา เพื่อนำไปเป็นหนึ่งในบทเรียนของตำราในรายวิชา บรรยายภาควิทยาเบื้องต้น วิชาภูมิศาสตร์ภาควิทยาและการรับรู้ระยะไกลในระดับปริญญาตรี/โท

1.6 ระยะเวลาดำเนินการวิจัย

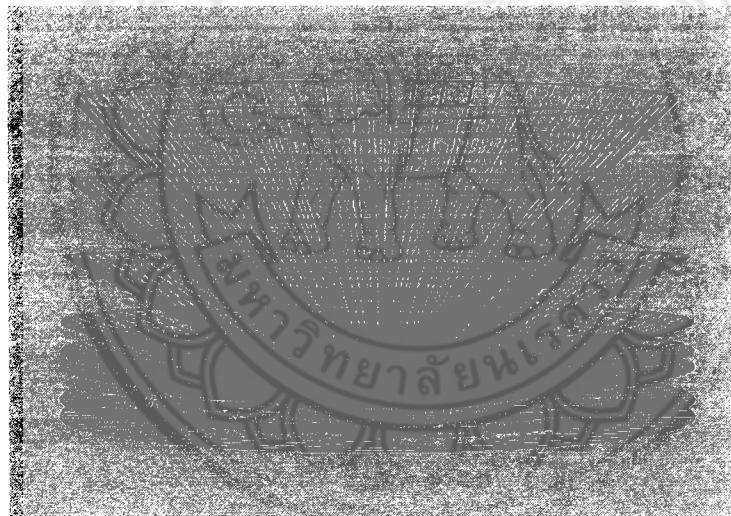
| ปี | กิจกรรม | พ.ค. | พ.ย. | ธ.ค. | ม.ค. | ก.พ. | มี.ค. | เม.ย. | พ.ค. | มิ.ย. | ก.ค. | ส.ค. | ก.ย. |
|------|--|------|------|------|------|------|-------|-------|------|-------|------|------|------|
| 2561 | ทบทวนวรรณกรรมที่เกี่ยวข้อง | X | X | X | | | | | | | | | |
| 2561 | รวบรวมข้อมูลน้ำฝนสถานีและข้อมูลเรดาร์ตรวจอากาศ | X | X | X | X | X | | | | | | | |
| 2562 | ตรวจสอบความถูกต้องข้อมูล | | | | X | X | X | | | | | | |
| 2562 | ประมาณผลข้อมูลเรดาร์เบื้องต้น | | | X | X | X | X | X | X | | | | |
| 2562 | ออกแบบสถานที่ | | | | | | X | X | X | | | | |
| 2562 | ออกแบบและพัฒนาการไม้สักแม่น้ำ | | | | | | X | X | X | X | | | |
| 2562 | จัดอบรมเชิงมานาคอมพิวเตอร์องค์ความรู้ | | | | | | | | | | | X | |
| 2562 | จัดทำรายงานฉบับสมบูรณ์ | | | | | | | | | | | | X |



บทที่ 2 วรรณกรรม/สารสนเทศ (Information) ที่เกี่ยวข้อง

2.1 เรดาร์ตรวจอากาศภาคพื้นดิน

เรดาร์ตรวจอากาศภาคพื้นดิน ซึ่งมีหลักการเก็บข้อมูลกลุ่มฝนในรูปแบบของการสัมผัสระยะไกล (Remote sensing) ซึ่งโดยปกติจะทำการวัดสัญญาณในมุมยกใดๆ ให้ครบ 360° จากนั้นจะทำการเปลี่ยนมุมยกเพื่อทำการวัดให้ครบจำนวนมุมยกที่ได้ถูกออกแบบเอาไว้เรียกวิธีการเก็บข้อมูลชนิดนี้ว่าการวัดแบบปริมาตรหรือ Volume scanning ดังแสดงในภาพที่ 2 ซึ่งจะสามารถนำมาวิเคราะห์ข้อมูลของโครงสร้างภายนอกกลุ่มฝนในรูปแบบปริมาตรสามมิติ โดยกรมอุตุนิยมวิทยาแห่งประเทศไทยได้ออกแบบการเก็บข้อมูลเป็นจำนวน 4 ครั้งต่อหนึ่งชั่วโมงซึ่งรวมถึงสถานีเรดาร์พิชณุโลก โดยมีรัศมี 240 กิโลเมตรจากตัวสถานี นอกจากนั้นยังมีมุมยกเป็นจำนวน 4 มุม ที่สามารถเก็บข้อมูลอย่างต่อเนื่องเพื่อการเฝ้าระวังและเตือนภัยสภาพอากาศที่จะเกิดขึ้นกับประชาชน



ภาพที่ 2.1 การเก็บข้อมูลของเรดาร์ตรวจอากาศแบบหลายมุมยกซึ่งประกอบกันเป็นปริมาตรหรือ Volume scan (Djurcilov and Pang, 2000)

หลักการเรดาร์ตรวจอากาศภาคพื้นดินสามารถศึกษาได้จากทำรายละเอียด (e.g., Battan 1973; Sauvageot 1992; Doviak and Zrinc 1993; Rinehart 1997) ในที่นี้จะกล่าวถึงอย่างย่อเรดาร์ตรวจอากาศภาคพื้นดินนั้นจะปล่อยพลังงานแม่เหล็กไฟฟ้าออกจากจานเรดาร์ในช่วงคลื่นของไมโครเวฟโดยมีความยาวช่วงคลื่นอยู่ระหว่าง 1-10 ซม. ที่ความถี่จาก 3 ถึง 30 GHz และจะรับค่า

การสะท้อนของสัญญาณเมื่อเจอกลุ่มฝนเรียกว่า Radar echo เข้าสู่ตัวรับสัญญาณในรูปแบบความเข้มของพลังงาน โดยที่ระดับความเข้มของพลังงานที่สะท้อนกลับจะขึ้นอยู่กับ ปริมาณและความแรงของกลุ่มฝน ขนาดของเม็ดฝนที่พลังงานแม่เหล็กไฟฟ้าไปกระทบซึ่งจะถูกวัดในรูปแบบของปริมาตร โดยค่าการสะท้อนกลับของพลังงานนั้นสามารถคำนวณได้จาก

$$Z = \frac{P_B \cdot r^2}{C \cdot k^2}$$

โดยที่

Z [mm⁶/m³] ค่าการสะท้อนของวัตถุเป้าหมายในรูปแบบปริมาตร

P_B [W] พลังงานในการสะท้อนกลับของคลื่นแม่เหล็กไฟฟ้า

r [m] ระยะทางของวัตถุเป้าหมาย

C [Wm⁵/mm⁶] ค่าสัมประสิทธิ์เรดาร์

k [-] ค่าคงที่ของการสะท้อนกลับของเม็ดฝน

โดยค่า k เป็นค่าเชิงประจักษ์ที่อยู่ในช่วง k=0.964 สำหรับเม็ดฝน ไปจนถึง k=0.456 สำหรับหิมะ ค่า Z โดยปกติแล้วจะใช้หน่วย dBZ แทน [mm⁶/m³]

$$dBZ = 10 \log_{10}(Z)$$

ข้อมูลที่ได้รับจากค่าการสะท้อนสามารถนำมาสร้างเป็นแผนที่ค่าการสะท้อนที่สามารถนำไปประมวลผลและวิเคราะห์ในระบบซอฟต์แวร์ที่มีความสามารถจัดการข้อมูลเชิงพื้นที่ได้ซึ่งที่ใช้กันอย่างแพร่หลายก็คือระบบ Geographic Information Systems หรือ GIS อย่างไรก็ตามในประเทศไทยเนื่องจากไม่มีการบริการข้อมูลเรดาร์ตรวจอากาศจากหน่วยงานของรัฐบาลในรูปแบบแผนที่เชิงเลขที่สามารถใช้ในการทำวิจัยต่อยอดได้ ดังนั้นในการวิจัยครั้งนี้จึงต้องทำการสร้างระบบฐานข้อมูลภูมิสารสนเทศฝนประมาณค่าจากเรดาร์ขึ้นมาใช้ในการวิเคราะห์ระบบ GIS สามารถนำเข้า จัดการ วิเคราะห์และแสดงผลข้อมูลเชิงตำแหน่งได้อย่างมีประสิทธิภาพ เนื่องจากสามารถเชื่อมโยงข้อมูลแผนที่ซึ่งประกอบด้วยข้อมูลปริภูมิทรายชนิดเข้ากับข้อมูลเชิงอรรถเอียง ทำให้การวิเคราะห์ทั้งเชิงห่วงเวลาและพื้นที่เป็นไปได้ง่าย อีกทั้งความสามารถในการแสดงผลเชิงตำแหน่งทำให้สามารถแสดงผลการวิเคราะห์ได้อย่างชัดเจน (Matyas 2014; Xie et al. 2005) ซึ่งความสามารถเหล่านี้ได้ถูกนำไปใช้ในหลายหลายในงานด้านอุทกวิทยา อุตุอุทกวิทยา อุตุนิยมวิทยาร่วมกับงานทางด้านภูมิศาสตร์วิทยา (Ogden 2001; Matyas 2014) อย่างไรก็ตามยังไม่มีงานวิจัยที่แพร่หลายในประเทศไทยเนื่องจากข้อจำกัดทางด้านเทคโนโลยีความรู้รวมถึงข้อจำกัดด้านข้อมูลเรดาร์ที่จะให้แก่นักวิจัยในการหาองค์ความรู้

การเปลี่ยนแปลงภูมิอากาศของโลก ส่งผลกระทบต่อความรุนแรงและความถี่ของฝน ซึ่ง Iwashima & Yamamoto (1993) ได้ทำการศึกษาโดยให้ค่าคงทนสถานีวัดปริมาณน้ำฝนของญี่ปุ่นเปรียบเทียบ กับสหราชอาณาจักรได้พบการเปลี่ยนแปลงนี้ โดยแนวโน้มของฝนมีการเปลี่ยนแปลงพบว่าฝนที่ตกหนักมี แนวโน้มเพิ่มขึ้นอย่างมีนัยสำคัญ อย่างไรพบว่าค่าเฉลี่ยฝนจะไม่ได้มีแนวโน้มที่เพิ่มขึ้น ซึ่งสอดคล้องกับ การวิเคราะห์ฝนที่รุนแรงของ Endo et al. (2009) ที่ศึกษาในภูมิภาคเอเชียตะวันออกเฉียงใต้รวมทั้ง ประเทศไทยโดยใช้ข้อมูลฝนสถานีที่มีรายละเอียดของตำแหน่งข้อมูลที่หนาแน่น ฝนในช่วงฤดูร้อน ของประเทศไทยนั้นได้รับอิทธิพลของลมมรสุมตะวันตกเฉียงใต้ ร่วมกับแนวประเทศไทยตัวเขต ร้อน แนวร่องมรสุม ความกดอากาศต่ำ รวมถึงพายุโซนร้อนที่เคลื่อนที่มาจากการหมุนเวียนแอตแลนติกตะวัน ตกก่อนจะอ่อนกำลังลงเป็นดีเปรสชัน ซึ่งจากการศึกษาของ Takahashi and Yasunari (2008) พบว่าฝนในช่วงครึ่งแรกของฤดูฝนจะอยู่ระหว่างเดือนพฤษภาคมถึงมิถุนายน ซึ่งได้รับอิทธิพลของลม มรสุมตะวันตกเฉียงใต้ ในขณะที่ฝนครึ่งหลังของฤดูมรสุมจะอยู่ระหว่างเดือนกรกฎาคมถึงกันยายน โดย ได้รับอิทธิพลของพายุโซนร้อนในช่วงเดือนกันยายน ซึ่งการเคลื่อนที่ของพายุโซนร้อนจะเคลื่อนที่เข้าหา ชายฝั่งตะวันตกของแปซิฟิก พบว่าความประปรวนของฝนในแต่ละปีจะขึ้นกับจำนวนพายุโซนร้อนนี้ที่ เคลื่อนที่เข้ามาในเขตอินโดจีน (e.g. Yokoi and Satomura 2008, Fukutomi and Yasunari 1999) เป็นที่น่าสนใจว่ารูปแบบน้ำฝนในระดับลุ่มน้ำจะมีการกระจายตัวอย่างไรหากนำข้อมูลที่มี รายละเอียดสูงอย่างเดาร์ตราชากาศภาคพื้นดินมาศึกษา โดย Okumura et al. (2003) ได้ใช้ เดาร์ตราชากาศสถานีอีกห้าแห่งเพิ่มเติมที่ตั้งอยู่ในภาคเหนือ ภาคกลาง และภาคใต้ เพื่อการเกณฑ์ทำการตรวจสอบรูปแบบ ฝนในบริเวณภาคเหนือเทียบกับผู้ที่ตั้งตัวอยู่ในภาคกลางและภาคใต้ ซึ่งครอบคลุมประเทศไทยเมียนมาร์ พบร่องน้ำฝนในภาคเหนือมีการเคลื่อนที่จากตะวันตกไปตะวันออกในลักษณะ squall line ซึ่งสอดคล้อง กับแบบจำลองอากาศระดับภูมิภาค Satomura (2000) อย่างไรก็ตามการศึกษาถึงรูปแบบเชิงพื้นที่ ยังไม่ได้ถูกศึกษาอย่างละเอียดโดยเฉพาะในภาคเหนือตอนบน แม้ว่าในงานของ Klongvessa (2017) จะกล่าวว่าฝนในพื้นที่ภูเขาของภาคเหนือของประเทศไทยมีแนวโน้มความรุนแรงที่สูงขึ้น แต่ก็เป็น เพียงการอนุมานจากข้อมูลฝนสถานีที่ไม่ได้มีความละเอียดเชิงพื้นที่มากนัก จึงไม่อาจจะสรุปรูปแบบ ของฝนในเชิงพื้นที่ได้อย่างสมบูรณ์ การเกิดฝนมักจะมีความสัมพันธ์กับภูมิประเทศซึ่งได้ถูกกล่าวไว้ โดย Houze (2012) ที่อธิบายถึงอิทธิพลของภูเขาที่ขวางกั้นทิศทางลมที่หอบความชื้นมาทำให้เกิดฝน ที่สัมพันธ์กับลักษณะภูเขาและทิศทางการวางในหลายรูปแบบดังตัวอย่างในภาพที่ 3 กระแสนนี้จะ เกิดการยกตัวของมวลอากาศขึ้นตามลักษณะภูมิประเทศ องค์ประกอบของการเคลื่อนที่ของมวล

อากาศนี้จะสนับสนุนให้เกิดเมฆที่ก่อให้เกิดฝนตกบริเวณด้านรับลม ส่วนแนวด้านหลังเข้าจะเกิดการระเหยของฝนไป



ภาพที่ 2.2 ฝนภูเขาที่เกิดจากการมีเทือกเขาทางตัวขวาทิศทางการเคลื่อนที่ของกระแสลม (a) ภูเขามีขนาดใหญ่ (b) ภูเขามีขนาดเล็ก (Houze 2012)

การวิเคราะห์ฟันที่รุนแรงเชิงพื้นที่จำเป็นที่จะต้องมีฐานข้อมูลที่มีความถูกต้องสูงซึ่งจะต้องมีการปรับแก้ร่วมกับผู้สำรวจที่ผ่านการตรวจสอบคุณภาพมาจากการแล้วก่อนจึงจะนำมาวิเคราะห์รูปแบบของฝนได้โดยใช้ความไม่แน่นอนของปัจจัยที่ควบคุมได้ในตอนต้น

2.2 งานวิจัยเดาร์ตราชากาศที่เกี่ยวข้อง

การทำความเข้าใจกับการกระจายตัวและความรุนแรงของสภาพฟันทางกายภาพนั้นจำเป็นที่จะต้องนำข้อมูลที่มีความละเอียดสูงเชิงห้วงเวลาและพื้นที่มาทำการศึกษา ซึ่งในประเทศไทยที่พัฒนาแล้วยกตัวอย่างเช่นประเทศไทยหรือเมริกาได้มีการสร้างข้อมูลเชิงพื้นที่จากเดาร์ตราชากาศภาคพื้นดินเพื่อใช้ในการตรวจและเฝ้าระวังภัยในลักษณะใกล้เวลาจริง (Near real time) ซึ่งข้อมูลจะถูกนำมาสร้างเป็นแผนที่ระบบกริดที่พร้อมจะนำเข้ามาวิเคราะห์ในระบบภูมิสารสนเทศให้กับนักอุตุนิยมวิทยาใช้ในการวิเคราะห์สภาพอากาศ นอกจากนั้นข้อมูลดังกล่าวยังถูกต้องเก็บไว้อย่างเป็นระบบเพื่อที่จะเตรียมไว้ให้นักวิจัยทางด้านภูมิอากาศศาสตร์ นักอุตุกวิทยา นักภูมิศาสตร์ หรือนักวิจัยที่เกี่ยวข้องกับศาสตร์ด้านการจัดการพื้นที่ไว้ใช้ในงานด้านวิจัยอันจะสร้างองค์ความรู้ที่นับมูลค่าไม่ได้ให้กับประเทศไทย

ในประเทศไทยที่กำลังพัฒนาดั้งเด่นประเทศไทยแม้ว่าจะมีการซื้อข้อมูลเดาร์ตราชากาศเข้ามาใช้ในงานทางด้านอุตุนิยมวิทยาเชิงปฏิบัติการ แต่ในเบื้องต้นการวิจัยยังมีให้เห็นน้อยโดยเฉพาะงานวิจัยที่เกี่ยวข้องกับอุตุนิยมวิทยาหรือสภาพฟันที่รุนแรง ซึ่งจะพบในการประยุกต์ใช้ข้อมูลเดาร์ตราชากาศของกรมฝนหลวงและการบินการเกษตรที่ได้นำมาใช้ในการหาความสัมพันธ์ระหว่างฝนภาคพื้นดินกับค่าการสะท้อนของเดาร์ตราชากาศในภาคตะวันออกของประเทศไทย (ภักดี จันทร์เกษ 2557)



22 มี.ค. 2565

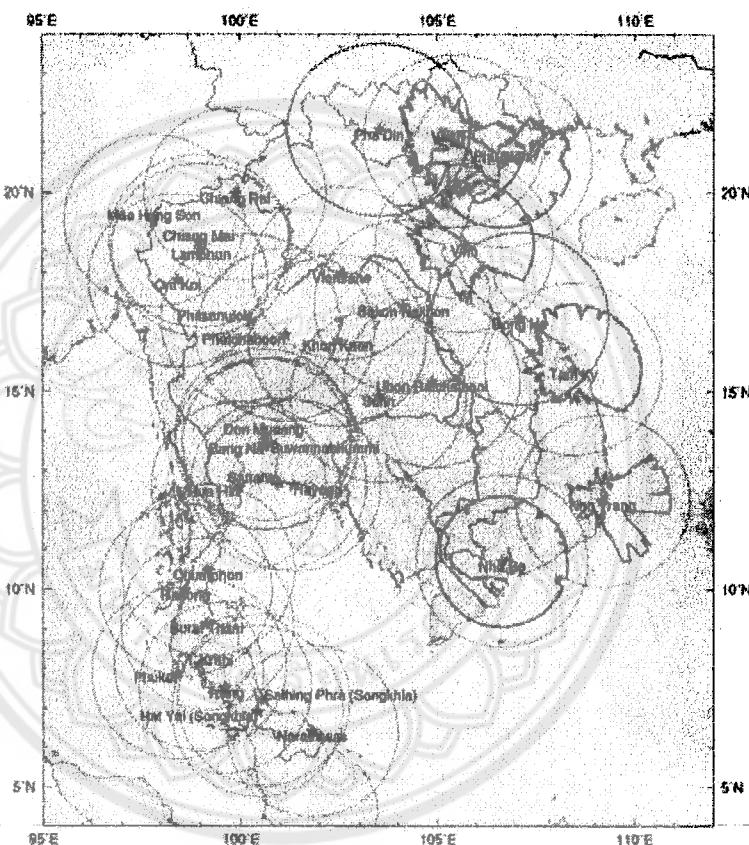
นอกจากนั้นกลุ่มวิจัยกลุ่มเดียวกันยังได้นำเรเดาร์ตรวจจากมาศึกษาลักษณะของการเกิดปรากฏการณ์ลูกเห็บในภาคเหนือของประเทศไทยซึ่งทำให้ทราบถึงลักษณะการเกิด ความรุนแรง และภัยภาพที่เกี่ยวข้องได้ซึ่งถือว่าเป็นงานขั้นแรกที่ทำโดยกลุ่มของนักวิจัยไทย (Chantraket 2015) ส่วนงานทางด้านการประมาณค่าฝนเพื่อหาความสัมพันธ์ของการสะท้อนเรเดาร์กับฟันภาคพื้นดินนั้นได้ถูกนำมาเป็นหัวข้อวิจัยค่อนข้างมากซึ่งส่วนใหญ่จะถูกตีพิมพ์ลงงานจากการใช้ข้อมูลเรเดาร์ของกรมฝนหลวงและการบินการเกษตรรวมถึงเรเดาร์ของกรุงเทพมหานครโดย (Chumchean 2010) อาย่างไรก็ตามในการนำเรเดาร์ของกรมอุตุนิยมวิทยามาใช้ในการวิจัยนั้นก็มีนักวิจัยหลายกลุ่มที่พยายามจะนำมาหาความสัมพันธ์ร่วมกับสถานีวัดฝนภาคพื้นดิน โดย Tantanee and Prakarnrat (2008) ได้หาความสัมพันธ์ของเรเดาร์พิชณ์โลก กับ ฟันภาคพื้นดินนี้ด้วยการใช้เทคนิคแนวกันชนในระบบภูมิสารสนเทศ Mahavik (2013) ได้ทำการหาความสัมพันธ์เพื่อหาค่าสัมประสิทธิ์ปรับแก้ค่าฝนประมาณค่าของสถานีเรเดาร์สกัดครึ่งให้ได้ผลการปรับแก้ที่ค่อนข้างดีด้วยวิธีการที่ง่ายที่สามารถนำไปปรับใช้ในงานภาคปฏิบัติการได้ นอกจากนั้น Mahavik (2017a) ได้ใช้เรเดาร์ของกรมอุตุนิยมวิทยาสถานีเพชรบูรณ์ ในรูปแบบของปริมาตรร่วมกับฟันภาคพื้นดินเพื่อหาความสัมพันธ์ระหว่างฝนประมาณค่าในระดับความสูง 3 กิโลเมตรกับฟันภาคพื้นดินด้วยการเปรียบเทียบ 5 วิธีการเพื่อทำการปรับแก้ค่าความเบี่ยงเบน โดยงานวิจัยนี้ได้ใช้ไลบรารีเรเดาร์รหัสเปิด Wradlib (Heimannmn, 2013) ในการจัดการข้อมูลเรเดาร์ด้วยภาษาไพธอน ซึ่งผลลัพธ์สามารถนำมายังวิเคราะห์ในระบบภูมิสารสนเทศได้

แม้ว่างานวิจัยที่กล่าวมาข้างต้นจะใช้ข้อมูลฝนประมาณค่าในการวิจัย อย่างไรก็ตามยังมีการวิจัยที่จำกัดในเรื่องการหารูปแบบฟันด้วยการใช้ข้อมูลเรเดาร์ตรวจจากมาศึกษาที่สัมพันธ์กับลักษณะอุตุนิยมวิทยา และภูมิศาสตร์ที่เป็นความรู้พื้นฐานในการใช้อธิบายเหตุการณ์ทางธรรมชาติ การศึกษาในอดีต ด้วยข้อมูลเรเดาร์พบว่ามีกลุ่มของ Prof. Satomura (อมกอย) ที่ใช้ข้อมูลเรเดาร์ของกรมฝนหลวงและการบินการเกษตรในการวิเคราะห์ลักษณะการเคลื่อนที่ของกลุ่มฝนในรอบวัน (Diurnal cycle) โดยใช้เรเดาร์ของสถานีเมืองก่ออยในการเปรียบเทียบลักษณะฝนในรอบวันระหว่างช่วงก่อนฤดูฝนและหลังฤดูฝน นอกจากนั้นกลุ่มวิจัยเดียวกันยังได้นำค่าการสะท้อนของเรเดาร์ตรวจจากมาศึกษาในภาคอินโดจีนตอนกลางพบว่ามีความเร็วของการเคลื่อนที่ของฝนแตกต่างกับความเร็วที่วัดได้ของสถานีเมืองก่ออย จากสองงานที่กล่าวมานี้ไม่ได้นำค่าฝนสถานีมาทำการวัดสอบ ซึ่งต่อมาก็มี Prof. Mahavik et al. (2014) ได้นำข้อมูลฝน ณ สถานีภาคพื้นดินจากกรมอุตุนิยมวิทยามาทำการเทียบสอบเพื่อแปลงค่าการสะท้อนจากเรเดาร์ตรวจจากมาศึกษาเป็นฝนประมาณค่าของสถานีเวียงจันทร์ เพื่อทำการหารูปแบบฟันเชิงพื้นที่ของฝนด้วยการใช้สถิติในการสกัดหารูปแบบที่เป็นเอกลักษณ์ของฟันจากวิธีการ Empirical Orthogonal

Function ทำให้พบว่าฝนบริเวณที่ราบสูงโครงสร้างและแอ่งสกلن cynn มีความสัมพันธ์กับลักษณะภูมิประเทศจะมีปริมาณฝนมากับบริเวณหน้าเข้าตามรูปแบบที่ได้จากการสกัดค่าจากเดкарตตรวจอากาศ นอกจากนั้นงานวิจัยในระดับปริญญาของ Mahavik et al. (2015) และ Mahavik et al. (2017c submitted) ยังได้มีการนำเดкарตตรวจอากาศทั้ง 3 สถานีในภูมิภาคนี้มาทำการต่อ กันให้เป็นแผนที่ฝนประมาณค่าที่ถูกปรับแก้ด้วยฝนสถานีในฤดูร้อนของปี 2553 เพื่อทำการหาลักษณะฝนเชิงพื้นที่ในรูปแบบวัฏจักรฝนรายวันทำให้พบการเคลื่อนที่ของฝนในบริเวณเดкарต เวียงจันทน์นั้นเคลื่อนที่จากทิศตะวันตกเฉียงเหนือไปทิศตะวันออกเฉียงใต้ในช่วงเข้ามีดซึ่งจะเป็นประโยชน์ต่อการตรวจสอบแบบจำลองอุตุนิยมวิทยาต่อไป อย่างไรก็ตามงานวิจัยในภูมิภาคนี้ยังขาดการเขื่อมโยงในการนำข้อมูลเดкарตตรวจอากาศมาวิเคราะห์เชื่อมโยงกับระบบภูมิสารสนเทศอย่างเป็นระบบ ซึ่งจำเป็นจะต้องใช้ความรู้ทางภูมิศาสตร์ อุตุนิยมวิทยาและภูมิอากาศวิทยาในทุกกระบวนการแม้ว่าในงานของ Mahavik (2017a) จะสร้างข้อมูลที่สามารถนำเข้าในระบบภูมิสารสนเทศได้แล้วก็ตามแต่ก็ยังไม่ได้นำมาวิเคราะห์ฝนเชิงพื้นที่และเชิงเวลาเพื่อเข้าใจในเชิงกายภาพในระดับลุ่มน้ำยมซึ่งเป็นที่คาดว่าจะพบองค์ความรู้ของฝนภายในพื้นที่ขนาดเล็ก (local climate) อันจะเป็นประโยชน์ในการนำองค์ความรู้ไปประยุกต์ใช้เพื่อพัฒนาประเทศไทยต่อไป

2.3 แผนที่โมเดลข้อมูลเรดาร์ตรวจอากาศ

ข้อมูลเรดาร์ตรวจอากาศถูกใช้ในประเทศไทยอย่างแพร่หลายทั่วประเทศที่พัฒนาแล้ว Satomura et al. (2013) ได้สร้างข้อมูลเรดาร์โมเดลที่ถูกนำมาใช้ในงานวิจัยด้านอุตุนิยมวิทยาในภูมิภาคอินโดจีนเป็นกลุ่มแรกๆ โดยทำการทำโมเดลเรดาร์ในภูมิภาคอินโดจีนในช่วงที่พายุ Lekima สายตัวเพื่อทำความเข้าใจเกี่ยวกับโครงสร้างของการสลายตัวของพายุ ดังภาพที่ 2.3

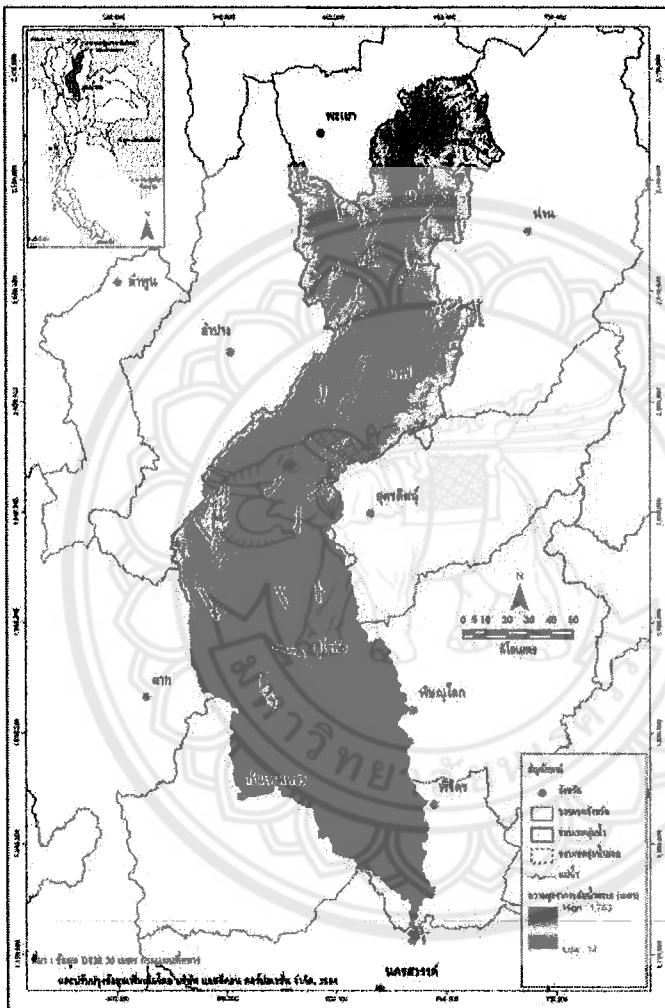


ภาพที่ 2.3 สถานีเรดาร์ที่เก็บข้อมูลทั่วภูมิภาคอินโดจีน แสดงเป็นลักษณะวงกลมรัศมีการสั่งเกตการณ์ของเรดาร์ตรวจอากาศ

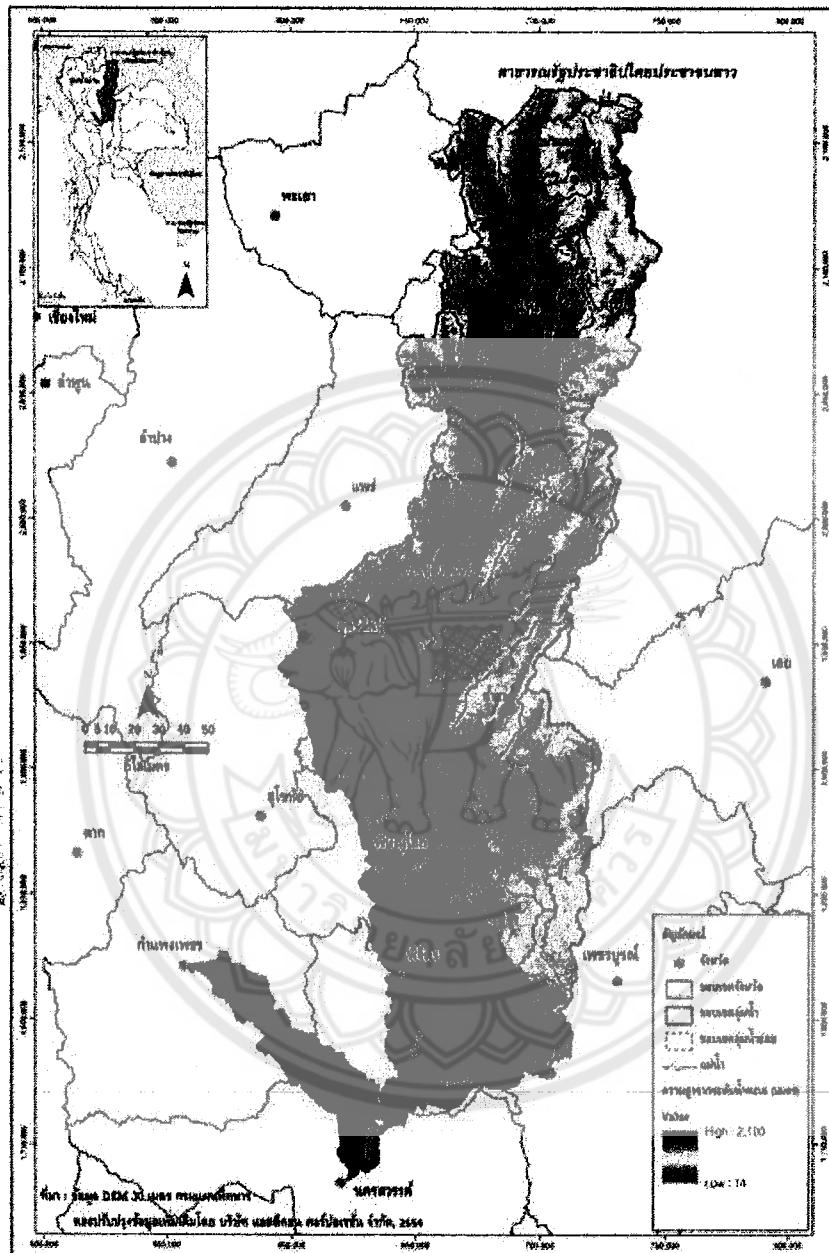
บทที่ 3 วิธีดำเนินการวิจัย

ในบทนี้จะกล่าวถึง พื้นที่ศึกษา ข้อมูลและเครื่องมือวิธีการที่ใช้ในการพัฒนาระบวนการโมเดลเดкарต ในลุ่มน้ำยมและน่านโดยใช้ซอฟต์แวร์ภูมิสารสนเทศที่เปิด QGIS ดังมีรายละเอียดดังที่ว้าข้อต่อไปนี้

3.1 พื้นที่ศึกษา



ภาพที่ 3.1 พื้นที่ศึกษาลุ่มน้ำยม (<http://mekhala.dwr.go.th/knowledge-basin-yom.php>)

ภาพที่ 3.2 พื้นที่ศึกษาคุณแม่น้ำท่วม (<http://mekhala.dwr.go.th/knowledge-basin-nan.php>)

พื้นที่ศึกษาประกอบไปด้วยคุณแม่น้ำท่วมและแม่น้ำน่าน โดยมีเดкарต์ตรวจสอบภาคจำนวน 3 สถานี ได้แก่ สถานีเรดาร์เชียงราย น่านและพิษณุโลกที่ถูกนำมาใช้ในการพัฒนากระบวนการโมเดลด้วยการใช้ไฟอ่อนสคิปท์ที่รันในโปรแกรม QGIS รวมเรียกว่า PyQGIS

3.2 เหตุการณ์พายุ

ช่วงเวลาที่ได้ทำการวิเคราะห์ข้อมูลนั้นครอบคลุมดูผ่านเดือนกรกฎาคม พ.ศ. 2561 ช่วงเวลาที่เสือกคือระหว่างวันที่ 15 กรกฎาคม ถึง 24 กรกฎาคม พ.ศ. 2561 ซึ่งเป็นครอบคลุมตั้งแต่พายุก่อตัวจากหย่อมความกดอากาศต่ำแล้วทวีกำลังเป็นพายุดีเปรสชันทางทิศตะวันตกเฉียงเหนือของมนิลาฟลิปปินส์ จนนั้นในวันที่ 17 กรกฎาคม พายุนี้ได้ทวีกำลังแรงขึ้นเป็นพายุโซนร้อน โดยในวันที่ 18 กรกฎาคมพายุนี้ได้ขึ้นฝั่งที่ทางทิศเหนือของเวียดนาม และได้อ่อนกำลังเป็นหย่อมความกดอากาศต่ำทางตอนเหนือของอินโดจีน ซึ่งส่งผลกระทบต่อทางเหนือของประเทศไทย โดยลุ่มน้ำยมและ่น่านได้รับอิทธิพลของการถล่มตัวของพายุเชินติญน์ การถล่มตัวของพายุนี้โดยทั่วไปมักไม่มีการศึกษาในรายละเอียดเนื่องจากขาดข้อมูล ทำให้ประเทศไทยมีความรู้เรื่องปริมาณฝนที่เกิดจากการถล่มตัวของพายุอย่างดีพอ ซึ่งเป็นองค์ความรู้ที่สำคัญมากเนื่องจากประเทศไทยไม่ได้รับอิทธิพลจากพายุโซนร้อนโดยตรง หากแต่ได้รับอิทธิพลจากการถล่มตัวของพายุโซนร้อนเป็นเวลาหลายวัน



ภาพที่ 3.3 การเคลื่อนที่ของพายุเชินติญน์ที่สิ่งทางเข้าทางทิศเหนือของประเทศไทย

(<https://www.prachachat.net/general/news-192629>)

3.3 ข้อมูลลุ่มน้ำ

ข้อมูลลุ่มน้ำยม

ลุ่มน้ำยม ตั้งอยู่ทางตอนเหนือของประเทศไทย ลักษณะลุ่มน้ำกว้างตัวตามแนวเหนือ-ใต้ มีพื้นที่ลุ่มน้ำรวมทั้งสิ้น 24,046.89 ตร.กม. ตำแหน่งที่ตั้งของลุ่มน้ำอยู่ระหว่างเส้นรุ้งที่ $14^{\circ}50'$ เหนือ ถึงเส้นรุ้งที่ $18^{\circ}25'$ เหนือ และระหว่างเส้นวางที่ $99^{\circ}16'$ ตะวันออก ถึงเส้นวางที่ $100^{\circ}40'$ ตะวันออก มีอาณาเขตติดต่อกับลุ่มน้ำหลักอื่นๆ 4 ลุ่มน้ำ คือ

- ทิศเหนือติดกับลุ่มน้ำโขง
- ทิศใต้ติดกับลุ่มน้ำปิง
- ทิศตะวันตกติดกับลุ่มน้ำวังและลุ่มน้ำปิง
- ทิศตะวันออกติดกับลุ่มน้ำน่าน



ภาพที่ 3.4 สภาพลำน้ำยมที่แห้งเหือดในช่วงฤดูฝน ถ่ายเมื่อ 19 กรกฎาคม พ.ศ. 2563 จากบน
สะพานลำน้ำ อำเภอโพธิ์ประทับช้าง จังหวัดพิจิตร

ภาพด้านบนแสดงให้เห็นลักษณะของลุ่มน้ำยมที่แห้งเหลือด แม้ว่าจะอยู่ในช่วงฤดูฝนของปี 2563 ลุ่มน้ำยมที่ตั้งเรื่องในช่วงจังหวัดพิจิตรนี้สามารถพัฒนาไป ซึ่งนอกจากลำน้ำยมจะมีกิจกรรมของน้ำท่วมแล้ว กัยแล้งยังเป็นวิกฤตที่พบเห็นบ่อยครั้ง ทำให้ประชาชนในพื้นที่ได้รับความเดือดร้อน ข้อมูลลุ่มน้ำน่าน

ลุ่มน้ำน่าน ตั้งอยู่ทางภาคเหนือของประเทศไทย มีพื้นที่ลุ่มน้ำรวมทั้งสิ้น 34,682.04 ตร.กม. พื้นที่ครอบคลุม 11 จังหวัด ได้แก่ จังหวัดกำแพงเพชร พะเยา แพร่ น่าน เลย สุโขทัย อุตรดิตถ์ พิษณุโลก พิจิตร เพชรบูรณ์ และนครสวรรค์ ลักษณะลุ่มน้ำกว้างตัวตามแนวทิศเหนือ-ใต้ ตั้งอยู่ระหว่างเส้นรุ้งที่ $15^{\circ} 42'$ เหนือ ถึงเส้นรุ้งที่ $18^{\circ} 37'$ เหนือ และระหว่างเส้นแบ่งที่ $99^{\circ} 51'$ ตะวันออก ถึงเส้นแบ่งที่ $101^{\circ} 21'$ ตะวันออก มีทิศเหนือของลุ่มน้ำติดกับลุ่มน้ำโขง ทิศใต้ติดกับลุ่มน้ำเจ้าพระยา ทิศตะวันออกติดกับลุ่มน้ำโขงและลุ่มน้ำป่าสัก และทิศตะวันตกติดกับลุ่มน้ำปราบ

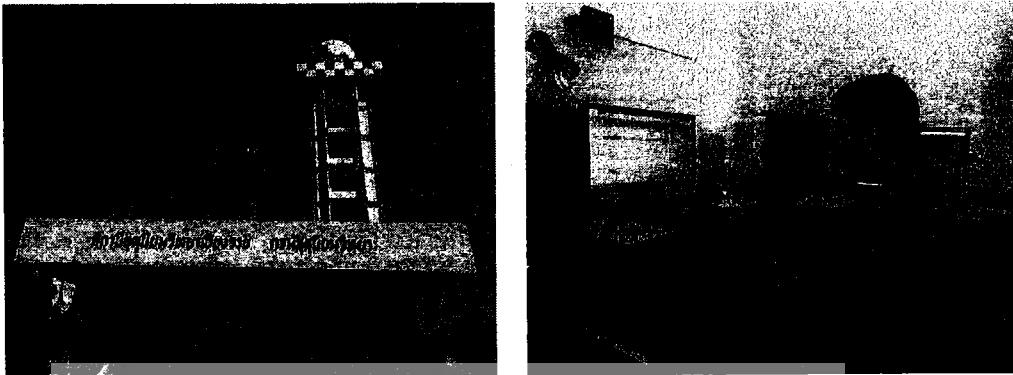
3.4 ข้อมูล

3.4.1 ข้อมูลเรดาร์

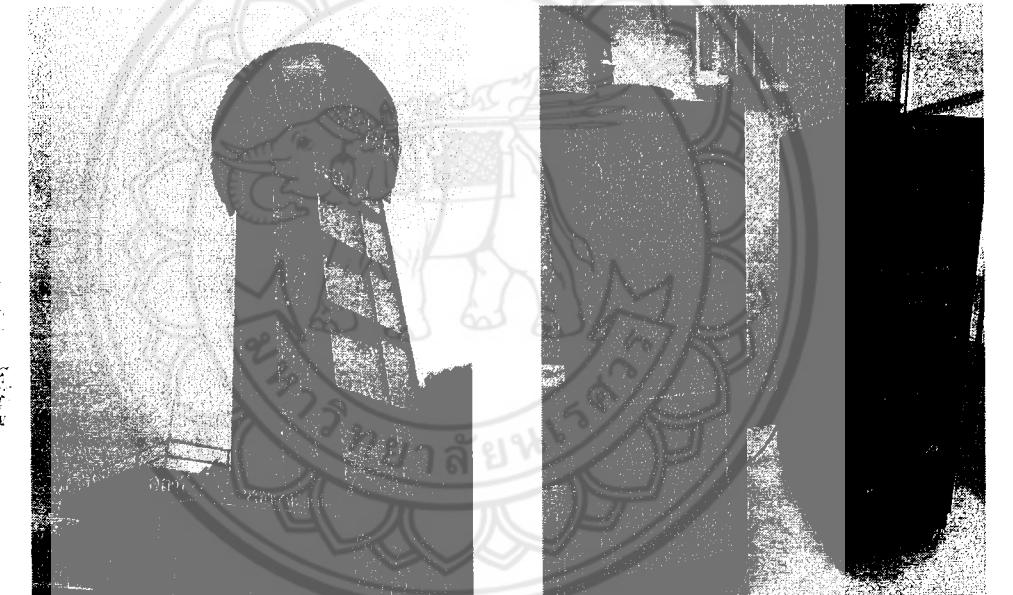
ตาราง 3.1 รายละเอียดเรดาร์ตรวจสอบอากาศ

| ชื่อสถานี | มุมยกที่ 1 | ระบบการตรวจวัด | บริษัทผู้ผลิต | ระบบการตรวจวัด |
|-----------|---------------|----------------|---------------|----------------|
| เชียงราย | 1.0° | C-band | EEC อเมริกา | 2 ระนาบ |
| น่าน | 0.5° | C-band | EEC อเมริกา | 2 ระนาบ |
| พิษณุโลก | 0.5° | C-band | EEC อเมริกา | 2 ระนาบ |

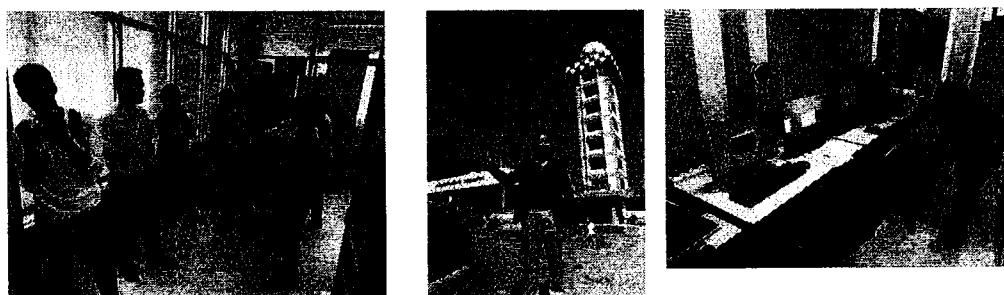
ข้อมูลเรดาร์ที่ใช้เป็นข้อมูลดิบที่ได้จากการอุตุนิยมวิทยาในช่วงพายุเซียบูลูในปี 2561 ซึ่งมีนามสกุล uf โดยจะต้องทำการแปลงด้วยซอฟต์แวร์ WCT เวอร์ชัน 4.3.1 (NOAA, 2020) เพื่อให้เป็นข้อมูลฟอร์แมต shapefile ที่จะถูกนำไปใช้ในซอฟต์แวร์ QGIS



ภาพที่ 3.5 สถานีเรดาร์ตรวจอากาศของกรมอุตุนิยมวิทยาจังหวัดเชียงราย ตั้งอยู่ในพื้นที่สนามบิน เชียงราย



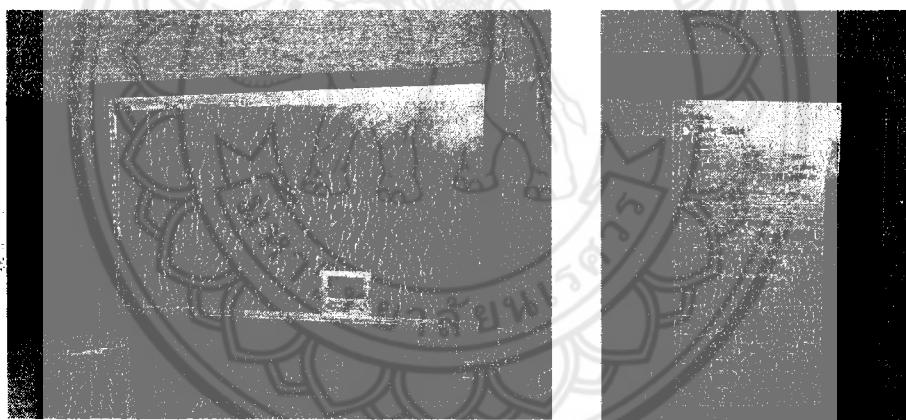
ภาพที่ 3.6 สถานีเรดาร์ตรวจอากาศจังหวัดพิษณุโลกและอุปกรณ์



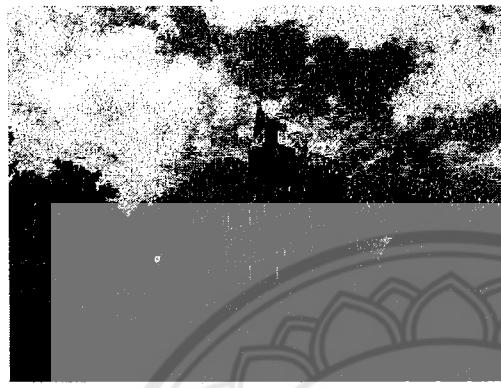
ภาพที่ 3.7 การแลกเปลี่ยนข้อมูลและการเยี่ยมชมอุปกรณ์สถานีเรดาร์ตรวจอากาศพิษณุโลก เชียงราย และสถานีเรดาร์กรมฝนหลวงย่ามแกร้วองกวาง จังหวัดแพร่



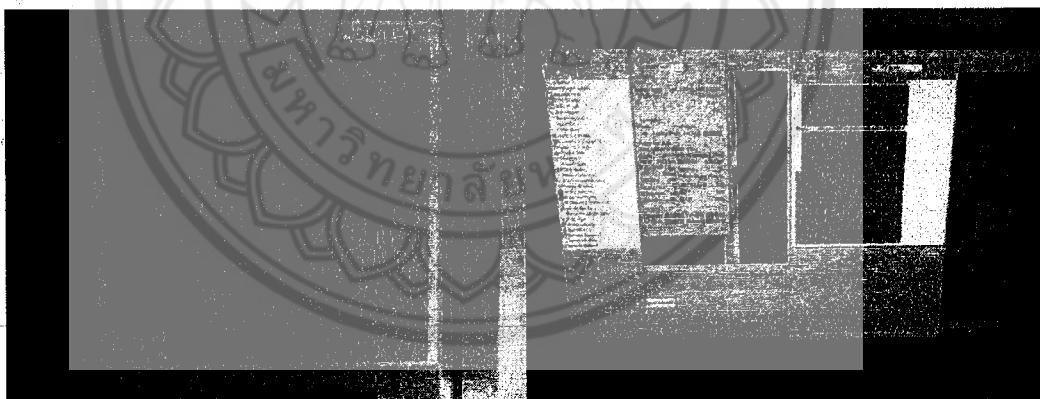
ภาพที่ 3.8 สถานีเรดาร์ตรวจอากาศน่าน อำเภอท่าวังผา



ภาพที่ 3.9 ระบบเรดาร์ตรวจอากาศสถานีท่าวังผา จังหวัดน่าน (ซ้าย) สภาพภูมิประเทศล้อมรอบสถานีท่าวังผา (ขวา) ระบบซอฟต์แวร์การตรวจวัดของเรดาร์ตรวจอากาศ ภาพด้านบนจะพบว่าสถานีท่าวังผา จังหวัดน่าน ตั้งอยู่ในภูมิประเทศที่รายล้อมไปด้วยภูเขา ซึ่งจะมีภูเขาระดับกลางด้านทิศตะวันตกของสถานีเรดาร์ในมุมยกแรก

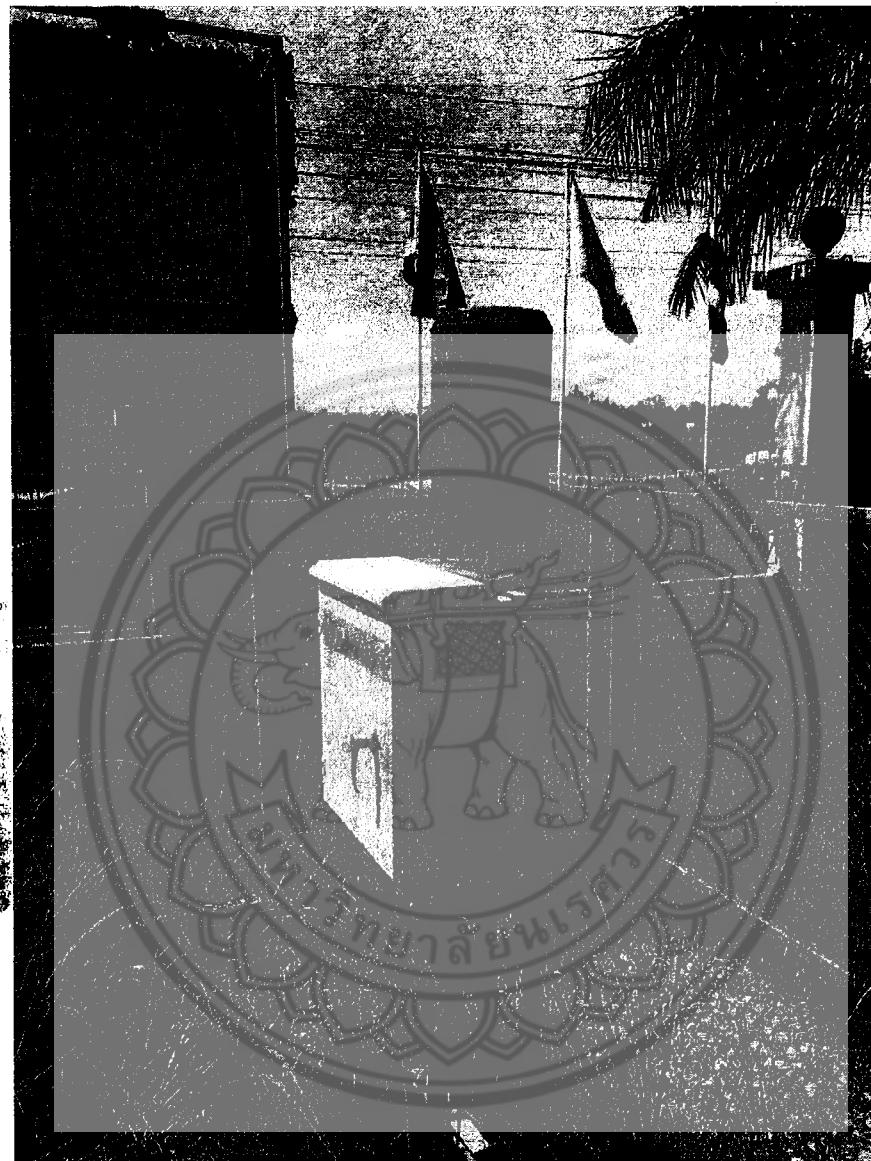


ภาพที่ 3.10 สถานีเรดาร์ตรวจอากาศฝันหลวง อำเภอร้องกวาง จังหวัดแพร่ ตั้งอยู่ในสำนักงานพัฒนาที่ดิน



ภาพที่ 3.11 แผนที่และภาพระบบเรดาร์ฝันหลวง อำเภอร้องกวาง จังหวัดแพร่ (ซ้าย)แผนที่ภูมิประเทศในขอบเขตการสังเกตการณ์ของเรดาร์ฝันหลวง อำเภอร้องกวาง จังหวัดแพร่ (ขวา) ภาพระบบประมวลผลข้อมูลเรดาร์สถานีร้องกวางในระบบ EEC

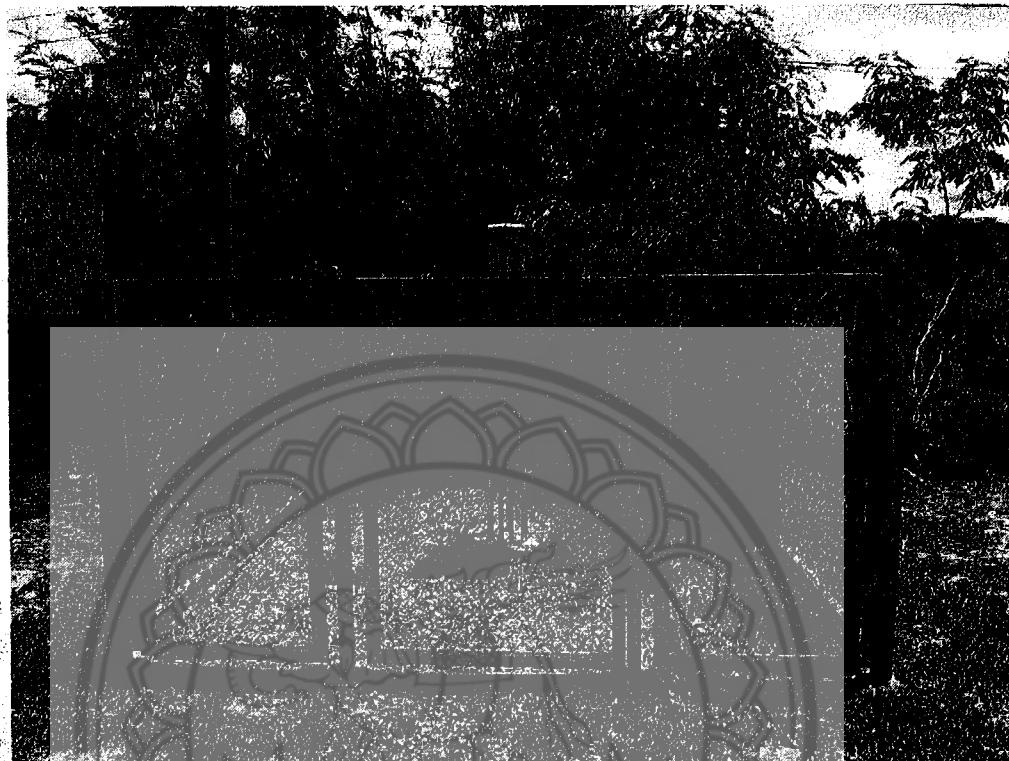
สถานีเรดาร์ของกรมฝันหลวงและการบินเกษตรนั้นมีหลายสถานีในประเทศไทย หนึ่งในนั้นตั้งอยู่ที่ อำเภอร้องกวาง จังหวัดแพร่ โดยเป็นสถานีแบบเคลื่อนที่ได้ทำการตรวจวัดโดยมีรัศมีทำการ 120 กิโลเมตร



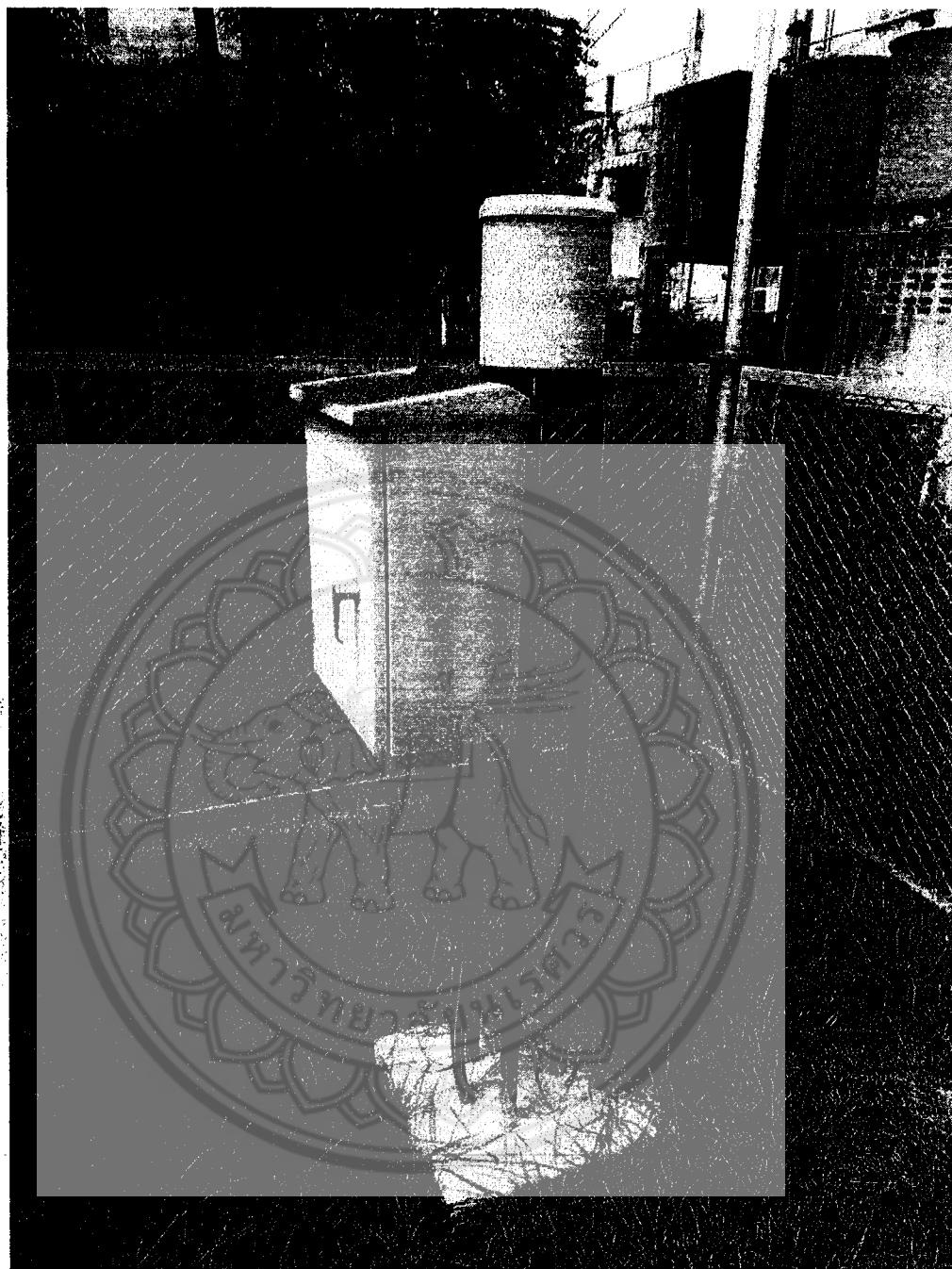
ภาพที่ 3.13 สถานีวัดฝันโกรมาตรของกรมอุตุนิยมวิทยา



ภาพที่ 3.14 สถานีวัดผ่นโลหะมาตรฐานของกรมอุตุนิยมวิทยา ติดตั้งที่เทศบาลตำบลไผ่รอบ อำเภอโพธิ์
ประทับช้าง จังหวัดพิจิตร



ภาพที่ 3.15 สถานีวัดฝันโถมารตรของกรมชลอุตุนิยมวิทยา ติดตั้งที่อบต.กำแพงดิน อําเภอสามจ่ำม จังหวัดพิจิตร



ภาพที่ 3.16 สถานีวัดผนโนกรรมตราของกรมอุตุนิยมวิทยา ติดตั้งที่เขิงสะพานพระร่วง อำเภอเมือง จังหวัดสุโขทัย

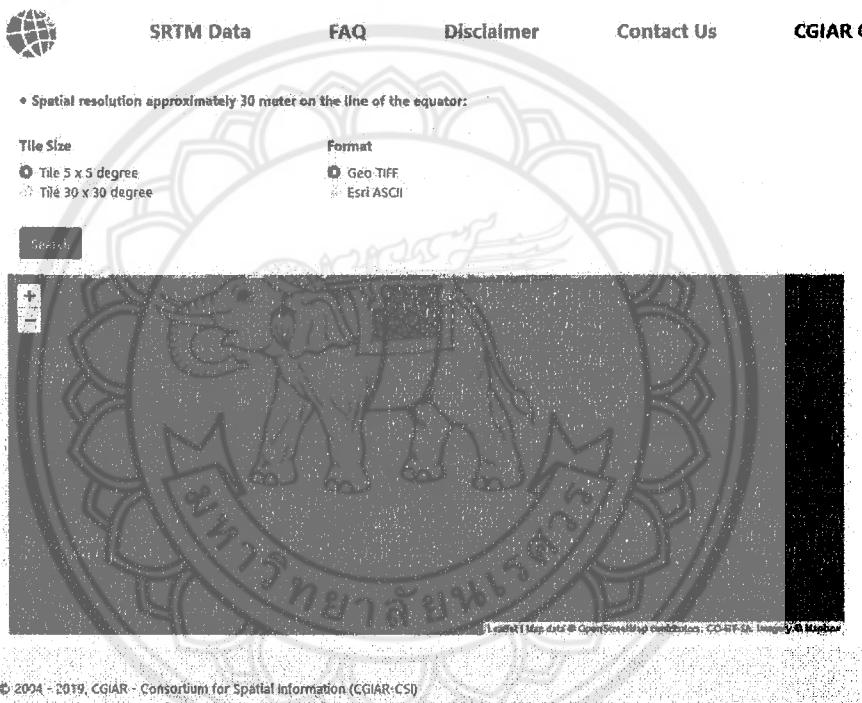
3.4.3 ข้อมูลแบบจำลองภูมิประเทศ

แบบจำลองภูมิประเทศ SRTM DEM เวอร์ชัน 4.1 ที่มีความละเอียดเชิงพื้นที่ระดับ 90 เมตรถูกนำมาใช้ในการวิเคราะห์การบดบังของเรดาวีม

ดาวน์โหลดจากเวปไซต์

<https://cgiarcsi.community/data/srtm-90m-digital-elevation-database-v4-1/> แบบจำลองภูมิประเทศจะทำการต่อและตัดให้ครอบพื้นที่สังเกตการณ์ของเรดาร์แต่ละสถานีเพื่อประยุกต์

ทรัพยากรในการประมวลผล

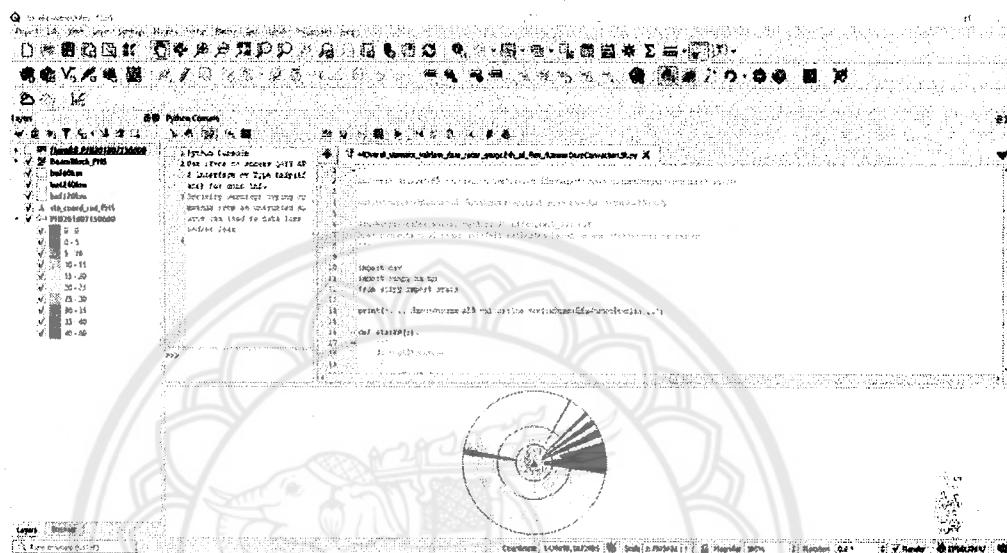


ภาพที่ 3.17 เวปดาวน์โหลดข้อมูล SRTM DEM 3Arc <http://srtm.csi.cgiar.org/srtmdata/>

3.5 เครื่องมือ ซอฟต์แวร์

3.5.1 ภาษาคอมพิวเตอร์เพื่อสอนใน QGIS

ภาษาไพธอน V3 ถูกนำมาใช้ในการพัฒนากระบวนการโมเดล化的ในโปรแกรม QGIS (QGIS, 2020) เวอร์ชัน 3.12 ที่ได้ดาวน์โหลดจากเวปไซต์หลัก โดยสามารถนำมาร่วมกับภาษาไพธอน เรียกว่า PyQGIS (PyQGIS cookbook, 2020) โดยทำพัฒนาโดยใช้ Python console

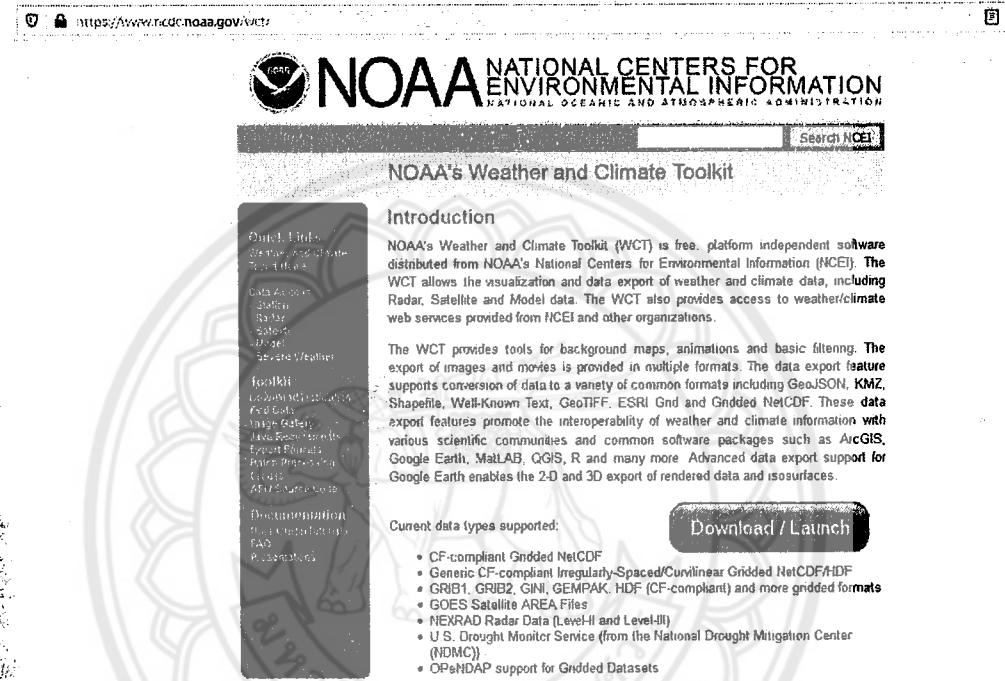


ภาพที่ 3.18 โปรแกรม QGIS โดยทำการรันโปรแกรมที่พัฒนาด้วย PyQGIS ใน python console

PYQGIS นั้นใช้ผู้ใช้เลือกใช้เวอร์ชัน 3 ซึ่งสามารถทำให้การประมวลผลอย่างมีประสิทธิภาพเร็วขึ้น เมื่อเทียบกับเวอร์ชัน 2 แหล่งเรียนรู้ PyQGIS มีหลายแหล่งที่ดีมาก แต่ส่วนใหญ่จะเป็นโค้ดแบบไฟล์ 2 ซึ่งมีความแตกต่างจากเวอร์ชัน 3 มาก อย่างไรก็ตามก็ยังมีแหล่งเรียนรู้ในอินเตอร์เน็ตที่มีผู้เชี่ยวชาญทำการเผยแพร่ในรูปแบบทั้งบล็อกและวิดีโอด้วยตัวเอง เช่น ของ Graser (2020) ที่สอนในรูปแบบของบล็อกตัวเรียนเรียน ส่วนของ OpensourceOptions (2020) จะทำการเผยแพร่สอน PyQGIS ทั้งแบบฟรีและไม่ฟรีในรูปแบบบล็อกที่สามารถศึกษาในวิดีโอที่อัปโหลดเข้าไปในยูทูปได้ นอกจากนั้นผู้ใช้ยังได้ศึกษาเพิ่มพูนองค์ความรู้ด้านการเรียนโปรแกรมด้วย PyQGIS จาก Udemy ที่ทำการสอนโดยอาจารย์ Miller (2020) ที่สอนในการใช้งาน PyQGIS 3.xx เพื่อประมวลผลข้อมูลภูมิสารสนเทศใน QGIS

3.5.2 โปรแกรม WCT

เนื่องจากมีการพัฒนากระบวนการแบบเวกเตอร์เพื่อทำการโมเดล化สภาพอากาศใน QGIS จึงต้องทำการแปลงข้อมูลจากรูปแบบ af มาเป็น shapefile โดยในที่นี้ได้ใช้โปรแกรม WCT เวอร์ชัน 4.3.1 ซึ่งเป็นซอฟต์แวร์ที่ใช้ในการวิเคราะห์ข้อมูลรวมถึงการสร้างออกข้อมูลเป็นไฟล์ที่ต้องการ

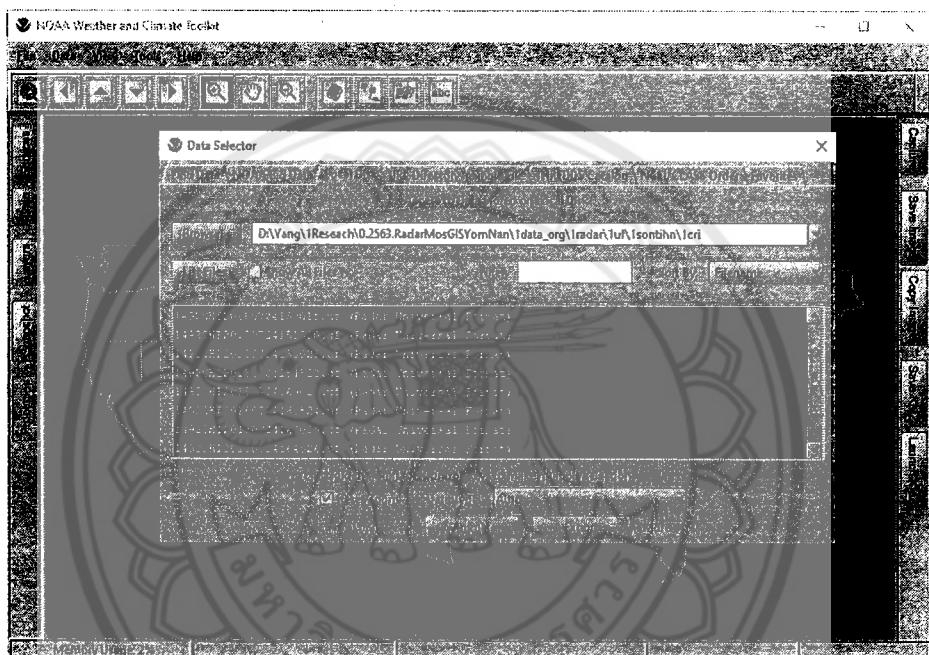


ภาพที่ 3.19 เว็บไซต์สำหรับการโหลดโปรแกรม WCT (<https://www.ncdc.noaa.gov/wct/>)



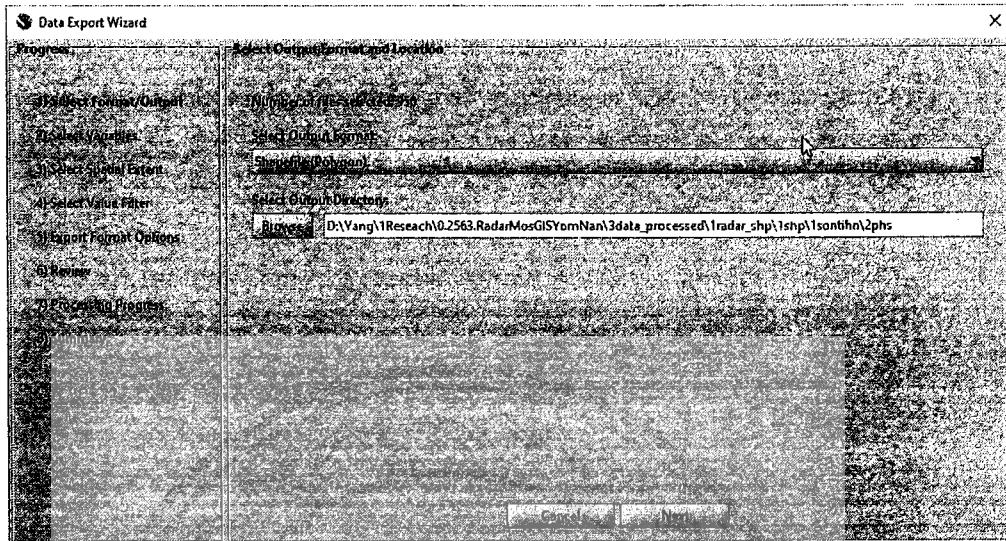
ภาพที่ 3.20 การเปิดข้อมูลเรดาร์ตรวจอากาศด้วย WCT จากข้อมูล UF

ภาพด้านบนจะพบว่าโปรแกรม WCT สามารถที่จะเปิดข้อมูลเรดาร์ตรวจอากาศนามสกุล nf มาวิเคราะห์ได้ สามารถเลือกว่าจะใช้มุมยกเท่าใดมาวิเคราะห์ หรือ เลือก Moment ซึ่งหมายถึงค่าตัวแปรใดๆที่ถูกเก็บไว้ในข้อมูลเรดาร์ได้ ในที่นี้ได้เลือกค่า CorrecteddBZCZ นั้นคือเป็นค่าการสะท้อนที่ถูกปรับแก้มาแล้ว ซึ่งจะพบว่าค่าการสะท้อนของเรดาร์พิเศษโดยในช่วงพายุเขินติ่งวันที่ 17 กรกฎาคม พ.ศ. 2561 นั้นมีค่าการสะท้อนที่เป็นสีน้ำเงินครอบคลุมพื้นที่ตอนเหนือของประเทศไทย ซึ่งคาดว่าจะมีข้อมูลของเรดาร์ที่ได้จากการอุตุนิยมวิทยาจะเป็นข้อมูลที่เก็บในเวลา UTC ซึ่งหากจะปรับให้เป็นเวลาท้องถิ่นต้องนำเวลาไป +7



ภาพ 3.21 โปรแกรม WCT เวอร์ชัน 4.3.1

เมื่อเปิดโปรแกรมมาแล้วจะต้องทำการ เลือกโฟลเดอร์ที่เก็บข้อมูล nf ของไฟล์ที่จะทำการประมวลผล ไว้ เพื่อจะทำการเช็คค่าพารามิเตอร์ที่ต้องการแปลงข้อมูลไปสู่ shapfile โดยทำการเลือกตรง export

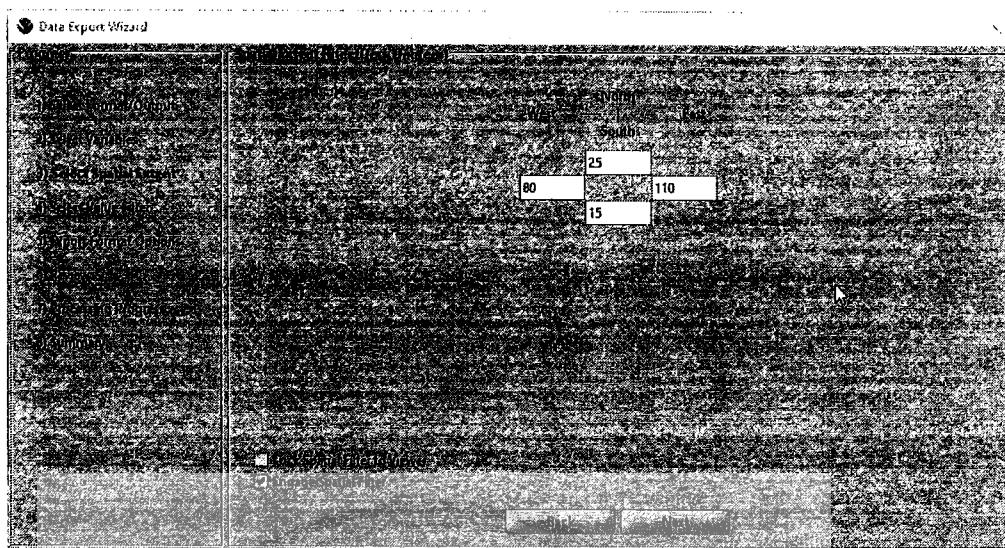


ภาพที่ 3.22 กระบวนการสำหรับเลือกค่าพารามิเตอร์ใน WCT ที่ต้องการในการส่งออกข้อมูลเป็น shapefile



ภาพที่ 3.23 ตัวแปรที่ต้องการส่งออกจากโปรแกรม WCT

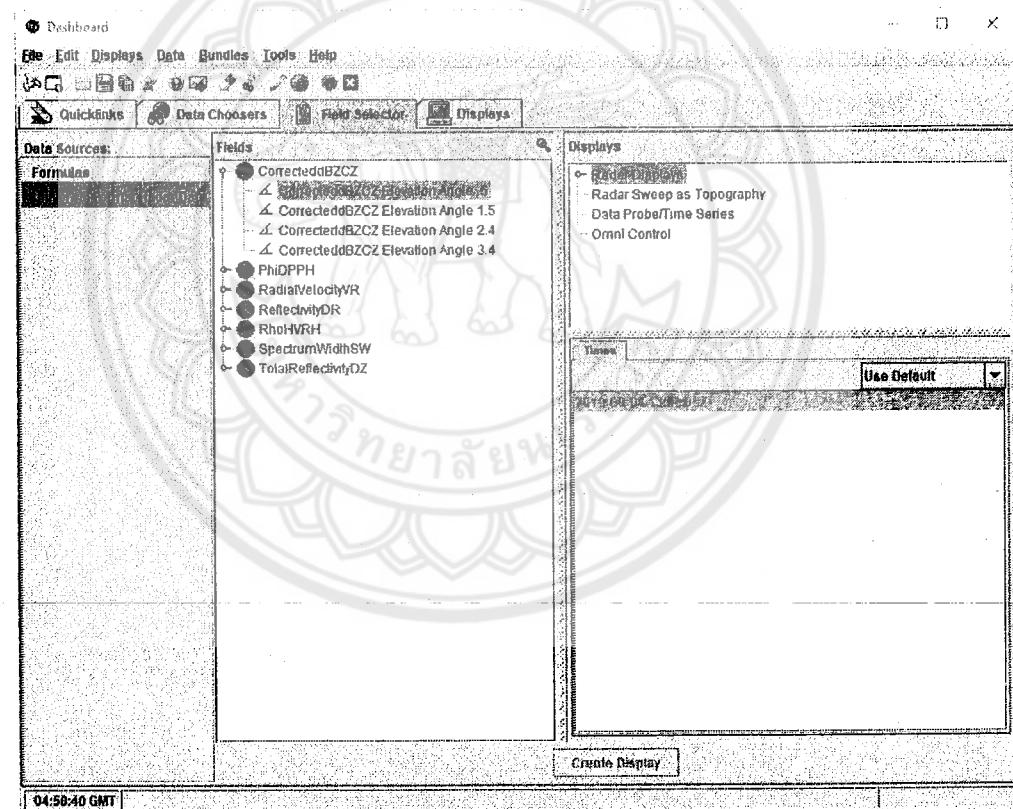
จากภาพด้านบนจะพบว่าในครั้งนี้ต้องการที่จะเลือกค่า Moment ที่เป็นค่า CorrecteddBZCZ นั่นคือค่าการสะท้อนที่ถูกปรับแก้มาแล้ว โดยเลือกมุมยกแรกที่มีค่ามุมยกเท่ากับ 1.0 องศา ซึ่งเป็นค่ามุมยกแรกของเรดาร์เชิงราย ในการศึกษารั้งนี้จะใช้มุมยกแรกเท่านั้น ต่อไปจะตั้งค่าพิกัดที่ต้องการส่งออก



ภาพ 3.24 ค่าพิกัดที่ต้องการส่งผลลัพธ์ออกในรูปแบบ shapefile ในโปรแกรม WCT
จากภาพด้านบน ค่าพิกัดที่ส่งออกจะต้องตั้งค่าให้ตรงกับพิกัดของรัศมีเรเดาร์ที่กำลังจะส่งออก โดยต้อง^{ให้มีขนาดที่ใหญ่กว่าจะป้องกันความผิดพลาดในการส่งออก} ต้องระมัดระวังหากมีการตั้งค่าเล็กกว่า
รัศมีของเรเดาร์ จะทำให้การส่งออกข้อมูลนั้นไม่ครบถ้วน

3.5.3 โปรแกรม IDV

โปรแกรม Intergrated Data Viewer (IDV) ถูกนำมาใช้เพื่อการวิเคราะห์กลุ่มฝนที่เกิดระหว่างพายุเชิงต่อเนื่องเพื่อตรวจสอบข้อมูลค่าการสะท้อนของเรดาร์ IDV เป็นซอฟต์แวร์ที่พัฒนาโดยหน่วยงาน University Corporation for Atmospheric Research (UCAR) ซึ่ง National Science Foundation (NSF) ของสหรัฐอเมริกาเป็นผู้ให้ทุนในการพัฒนาซอฟต์แวร์ ในงานวิจัยครั้งนี้ได้ใช้ IDV เวอร์ชัน 5.7 ในการวิเคราะห์ค่าการสะท้อน 4 มุมยกของกรมอุตุนิยมวิทยา โดย IDV จะแบ่งเป็นสองส่วน คือ Dashboard ดังภาพด้านล่างที่จะใช้ควบคุมในการเปิดปิดขั้นตอน ข้อมูล การเลือกค่าพารามิเตอร์ที่จะมาแสดงในส่วนที่สองคือ Map View



ภาพที่ 3.25 โปรแกรม IDV ในส่วนของ Dashboard

ภาพด้านบนแสดงส่วน Dashboard ที่ควบคุมในส่วนของ Mapview หากต้องการแสดงผลข้อมูลใดๆใน Mapview สามารถเลือกเปิดปิดข้อมูลในส่วนของ Dashboard ในแท็ปของ Data chosers ได้ โดยในการวิจัยครั้งนี้ใช้ข้อมูลเรดาร์ฟอร์แมท UF ที่สามารถเปิดด้วย IDV ได้โดยตรง



ภาพที่ 3.26 โปรแกรม IDV ในส่วนMapView ที่แสดงค่าการสะท้อนของเรดาร์ของสถานีพิษณุโลก
แสดงในมุมยกที่ 1 ในแบบ 2และ3มิติ

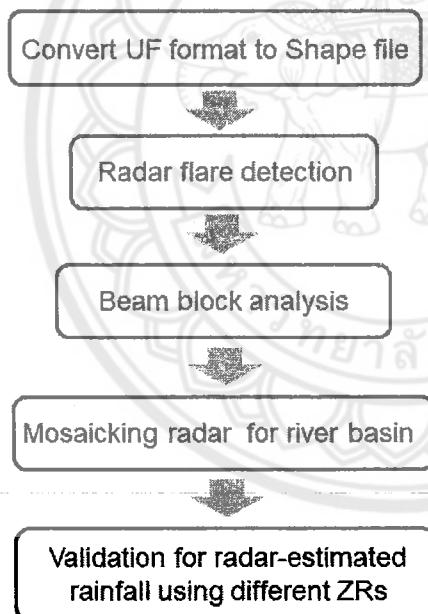
ภาพด้านบนแสดงมุมยกที่ 1 ทั้งในแบบ2และ3มิติในส่วนของ Mapview ซึ่งความสามารถของ IDV ที่สามารถแสดงในระบบ3มิตินี้สามารถช่วยในการวิเคราะห์โครงสร้างของตัวพายุ สามารถทราบได้ว่า พายุนี้ประกอบไปด้วยเมฆ convective หรือ stratiform ที่มีความสูงระดับใด นอกจากนั้น IDV ยังสามารถแสดงข้อมูลเรดาร์ในรูปแบบแอนิเมชันหลอยทั่วเวลา ทำให้สามารถดูพัฒนาการของพายุได้ เพื่อให้เข้าใจถึงกลไกหรือลำดับขั้นของพัฒนาการของตัวพายุในแต่ละช่วงเวลาที่สนใจ หากผู้สนใจในการนำไปโปรแกรม IDV ไปใช้ในการวิเคราะห์งานสามารถเรียนรู้ได้จากเวปไซต์ตามลิงค์ <https://www.unidata.ucar.edu/software/idv/gallery/> ซึ่งความสามารถของ IDV ไม่ได้จำกัดแค่ การแสดงผลของเรดาร์ แต่สามารถแสดงผลของข้อมูลบรรยายกาศและภูมิอากาศที่เป็นลักษณะของ ข้อมูล 4 มิติที่มีมิติเรื่องเวลาเข้ามาเกี่ยวข้อง นอกจานั้นโปรแกรม IDV ยังสามารถดึงข้อมูล อุตุนิยมวิทยาแบบใกล้เวลาจริงที่ให้บริการจากหน่วยงานอุตุนิยมวิทยาในสหรัฐอเมริกา เช่น ข้อมูลสภาพดาวเทียม ข้อมูลเรดาร์ทั้ง level II และ level III เป็นต้น ซึ่งเหมาะสมแก่การนำมาใช้ศึกษา ด้านอุตุนิยมวิทยาหรือใช้ในการเรียนการสอนในระดับอุดมศึกษาเป็นอย่างมาก ซึ่งจะดีกว่านี้หาก

ข้อมูลในประเทศไทยมีการให้บริการประชาชนแบบใกล้เวลาจริงตามฟอร์แมทที่โปรแกรม IDV สามารถอ่านค่า จะทำให้เกิดการเรียนรู้กับเยาวชนเนื่องจากสามารถเข้าถึงข้อมูลเหล่านี้ได้ตลอดเวลา ในท้ายที่สุดจะทำให้ประเทศไทยพัฒนาองค์ความรู้ด้านอุตุนิยมวิทยาขึ้นไปอีกระดับ

3.6 วิธีการ

ทุกรอบวนการในการศึกษาวิจัยครั้งนี้เกี่ยวข้องกับการวิเคราะห์ข้อมูลเชิงเลขทั้งหมด โดยได้มุ่งเน้น การพัฒนากระบวนการในระบบ GIS ในเรื่องการไม่เสียข้อมูลเรดาร์ตรวจอากาศ ซึ่งได้ใช้วิธีการ ประมวลผลแบบแบกเตอร์และรัสเตอร์เข้าช่วยในการประมวลผลข้อมูลที่มีความถี่ในการเก็บข้อมูลที่สูง

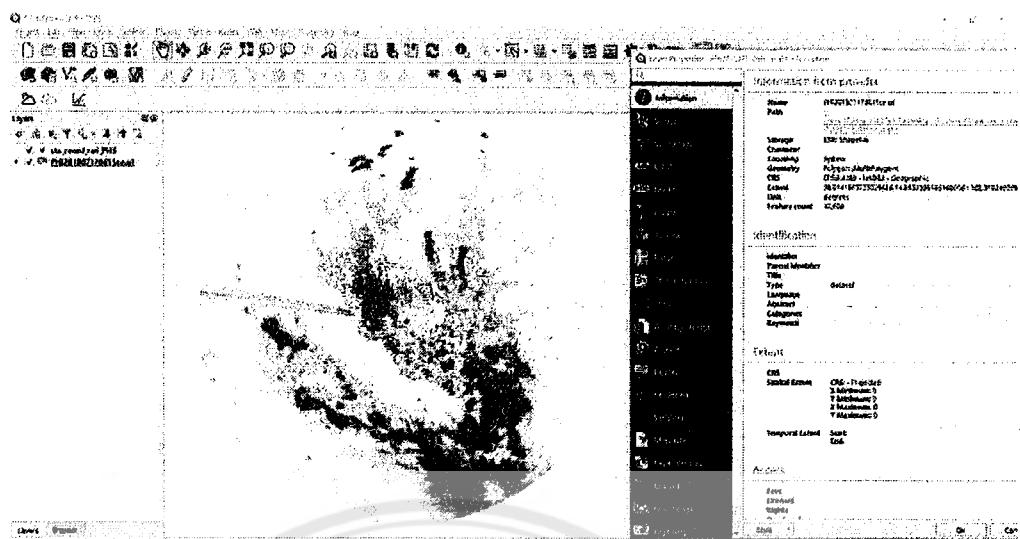
workflow



ภาพที่ 3.27 ขั้นตอนการดำเนินการวิจัยเพื่อการพัฒนาวิธีการการประมวลผลข้อมูล

3.6.1 การแปลงข้อมูล uf ไปเป็น shapefile

กระบวนการนี้จะแปลงข้อมูลดิบที่ได้รับมาจากกรมอุตุนิยมวิทยา เป็นรูปข้อมูลที่สามารถนำไปพัฒนากระบวนการใน GIS ได้ นั่นคือข้อมูลลักษณะแบบ vector ในรูปแบบ polygon ซึ่งข้อมูลที่ได้จากการแปลงด้วยซอฟต์แวร์ WCT นี้จะมีพิกัดเป็นแบบ NAD83-Geographic ซึ่งรูปปิดของข้อมูลวางแผนตัวตามแนวรัศมีของเรดาร์หรือ polar coordinate



ภาพที่ 3.28 ค่าการลงทะเบียนของสถานีเรดาร์พิษณุโลกวันที่ 17 เดือนกรกฎาคม ปี 2561 เวลา 06:15 UTC (+7 เวลาท้องถิ่น)



ภาพที่ 3.29 ลักษณะข้อมูลเรดาร์แบบ polar coordinate ที่มีการวางแผนตามมุมอչิมุทของรัศมีเรดาร์ตรวจสอบอากาศ

ภาพด้านบนจะพบว่าหากทำการขยายบางส่วนของข้อมูล จะพบลักษณะข้อมูลของเรดาร์จะวางตัวตามแนวมุมอչิมุทและระยะทางตามรัศมีของเรดาร์หรือที่เรียกว่า polar coordinate จะประกอบไปด้วยมุมอչิมุท (azimuth angle) และระยะทางตามแนวรัศมี (range bin) ซึ่งจะแตกต่างกันไปตามการตั้งค่าของแต่ละสถานี

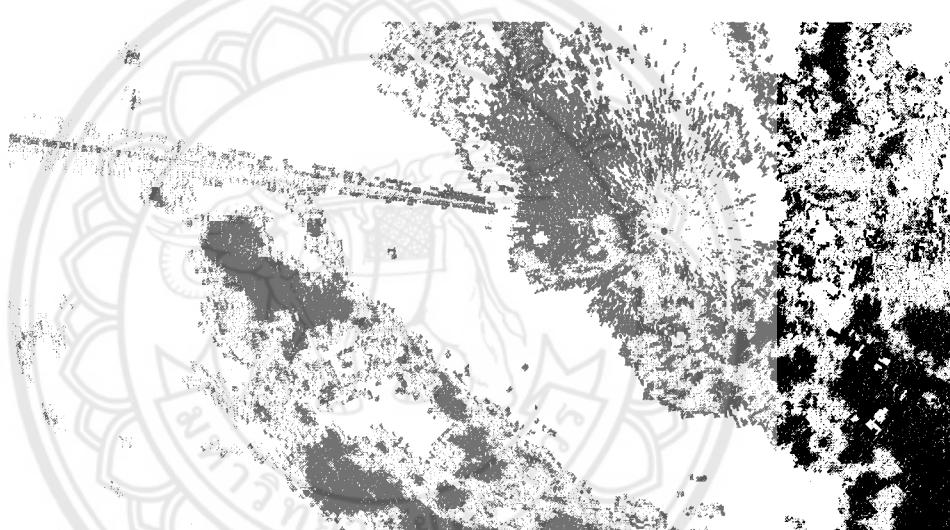
| PHI201807170615ce.uf :: Features Total: 47609, Filtered: 47609, Selected: 0 | | | | | | | | | |
|---|--------------------|-----------|-------|-----------|------------|------------|-----------|-----------|--|
| sweep | sweepTime | elevAngle | value | radialAng | begGateRan | endGateRan | heightRel | heightASL | |
| 1 | 0 2018-07-17T06... | 0.406 | 45.00 | 86.1875 | 1499.9430 | 2499.9048 | 10.7679 | 82.7679 | |
| 2 | 0 2018-07-17T06... | 0.406 | 34.00 | 86.1875 | 2499.9048 | 3499.8669 | 18.0937 | 90.0937 | |
| 3 | 0 2018-07-17T06... | 0.406 | 5.00 | 86.1875 | 7499.7148 | 8499.6768 | 56.4893 | 128.4893 | |
| 4 | 0 2018-07-17T06... | 0.406 | 15.00 | 86.1875 | 8499.6768 | 9499.6387 | 64.5217 | 136.5217 | |
| 5 | 0 2018-07-17T06... | 0.406 | 15.50 | 86.1875 | 9499.6387 | 10499.6000 | 72.6720 | 144.6720 | |
| 6 | 0 2018-07-17T06... | 0.406 | 18.00 | 86.1875 | 17499.5340 | 18499.2960 | 142.1136 | 214.1136 | |
| 7 | 0 2018-07-17T06... | 0.406 | 38.50 | 86.1875 | 18499.2960 | 19499.2570 | 151.3238 | 223.3238 | |
| 8 | 0 2018-07-17T06... | 0.406 | -4.50 | 86.1875 | 29498.8770 | 30498.8390 | 260.4069 | 332.4069 | |
| 9 | 0 2018-07-17T06... | 0.406 | -6.00 | 86.1875 | 30498.8390 | 31498.8000 | 271.0324 | 343.0324 | |
| 10 | 0 2018-07-17T06... | 0.406 | 6.00 | 86.1875 | 41498.4210 | 42498.3820 | 395.6638 | 467.6638 | |
| 11 | 0 2018-07-17T06... | 0.406 | 15.00 | 86.1875 | 42498.3820 | 43498.3430 | 407.7006 | 479.7006 | |
| 12 | 0 2018-07-17T06... | 0.406 | 13.50 | 86.1875 | 43498.3430 | 44498.3080 | 419.8551 | 491.8551 | |
| 13 | 0 2018-07-17T06... | 0.406 | 17.00 | 86.1875 | 44498.3080 | 45498.2690 | 432.1274 | 504.1274 | |
| 14 | 0 2018-07-17T06... | 0.406 | 19.50 | 86.1875 | 45498.2690 | 46498.2300 | 444.5175 | 516.5175 | |
| 15 | 0 2018-07-17T06... | 0.406 | 22.00 | 86.1875 | 46498.2300 | 47498.1910 | 457.0254 | 529.0254 | |
| 16 | 0 2018-07-17T06... | 0.406 | 17.00 | 86.1875 | 47498.1910 | 48498.1560 | 469.6510 | 541.6510 | |
| 17 | 0 2018-07-17T06... | 0.406 | 16.50 | 86.1875 | 48498.1560 | 49498.1170 | 482.3944 | 554.3944 | |
| 18 | 0 2018-07-17T06... | 0.406 | 15.50 | 86.1875 | 49498.1170 | 50498.0780 | 495.2556 | 567.2556 | |
| 19 | 0 2018-07-17T06... | 0.406 | 19.00 | 86.1875 | 50498.0780 | 51498.0390 | 508.2346 | 580.2346 | |
| 20 | 0 2018-07-17T06... | 0.406 | 13.00 | 86.1875 | 53497.9560 | 54497.9250 | 547.8781 | 619.8781 | |

ภาพที่ 3.30 ค่าแอทริบิวต์ของข้อมูลเรดาร์

ภาพด้านบนเป็นค่าแอทริบิวต์ของข้อมูลเรดาร์ตรวจอากาศ ซึ่งจะมีคอลัมน์ที่ได้จากการประมวลผลในขั้นตอนการส่งออกข้อมูลเป็น shapefile จากโปรแกรม WCT ซึ่งจะมีข้อมูลของมุมยก (sweep) ในที่นี่ค่าเป็น 0 คือมุมยกแรก เวลาของการตรวจอุณหภูมิ (sweepTime) มุมยก (elevAngle) ค่าการสะท้อนเมื่อนวายเป็น dBZ (value) ค่ามุมอุปจัมภีเมื่อนวายเป็นองศา (radialAng) ค่าระยะทางของแต่ละรูปปิดของเรดาร์หรือที่เรียกว่า bin gate นั่นคือค่าความละเอียดของการตรวจอุปจัมภีหรือ spatial resolution ในที่นี่เรียกว่า begGateRan จะพบว่ามีระยะห่างกัน 1000 เมตร นี่คือระยะเริ่มต้นของแต่ละรูปปิด ส่วนระยะทางสุดท้ายของแต่ละรูปปิดคือ endGateRan ความสูงลำบีมจากระดับน้ำทะเลปานกลางเมื่อนวายเป็นเมตร (heightRel) มีหน่วยเป็นเมตร ความสูงของลำบีมจากระดับน้ำทะเลปานกลางเมื่อนวายเป็นเมตร (heightASL) ซึ่งค่าคอลัมน์ทั้งหมดนี้ สามารถนำมาใช้ในการพัฒนาอัลกอริทึมวิธีการโมเดลเรดาร์ในระบบ GIS ต่อไปได้

3.6.2 การตรวจจับ Radar flare

Flare คือบริเวณที่ไม่มีฝนแต่ทำการสะท้อนกลับพบว่ามี ซึ่งทำให้การวิเคราะห์ปริมาณฝนในพื้นที่ที่สนใจผิดไปจากปริมาณจริง ดังนั้นจึงควรที่จะพัฒนาระบวนการในการตรวจและทำให้เป็นค่า NaN คือไม่มีค่าการสะท้อนก่อนเพื่อตัดข้อมูลนี้ออกไปจากการวิเคราะห์ ค่า flare นี้จะเปลี่ยนไปในแต่ละช่วงเวลาและสถานที่ ซึ่งทางเจ้าหน้าที่กรมอุตุนิยมวิทยากล่าวว่าค่า flare เกิดจากการประปของสัญญาณเดcar ในระบบ C-band ที่กรมอุตุนิยมวิทยาใช้ในการปฏิบัติการเดcar ตรวจอากาศ ที่มักจะประปกับสัญญาณโทรศัพท์ความของเอกชน ทำให้ flare นั้นปรากฏในตำแหน่งและเวลาที่ไม่ซ้ำกัน ดังนั้นจึงเป็นการยากที่จะจัดบริเวณ flare นี้ออกไป ดังนั้นงานวิจัยขึ้นนี้ส่วนหนึ่งจึงมุ่งขัด radar flare จากข้อมูลเวลาเตอร์ของ GIS ก่อน ซึ่งจะต้องหาความสัมพันธ์เพื่อสร้างสมการในการจัด radar flare

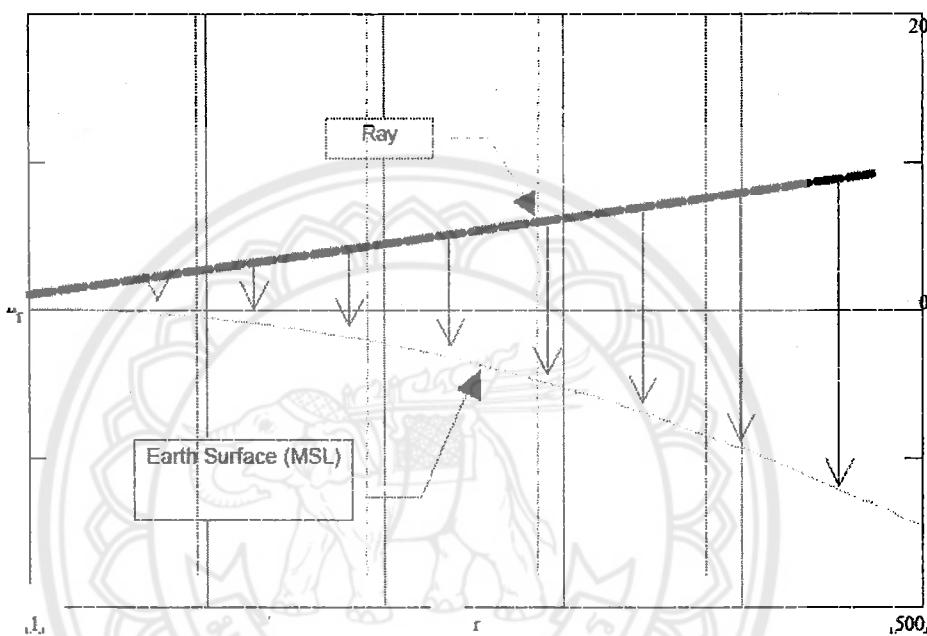


ภาพที่ 3.31 ตัวอย่าง radar flare ของสถานีพิษณุโลก วันที่ 17 เดือนกรกฎาคม ปี 2561 เวลา 06:15 UTC (+7 เวลาท้องถิ่น)

ภาพด้านบนจะพบว่า radar flare มีลักษณะเป็นเส้นตรงที่พุ่งออกจากตัวสถานีเรเดcar โดยมีค่าการสะท้อนที่ค่อนข้างเพิ่มความแรงขึ้นตามระยะทางที่ห่างออกไป ดังนั้นผู้วิจัยจึงใช้การสังเกตการนี้สร้างความสัมพันธ์ด้วยการใช้ regression analysis ในการตรวจจับ radar flare ในแต่ละมุมที่มีความสัมพันธ์ในเรื่องของการเพิ่มความแรงของค่าการสะท้อนตามระยะทางที่ใกล้ออกไปจากสถานีเรเดcar

3.6.3 การเคลื่อนที่ของเรดาร์บีมในชั้นบรรยากาศ

การจำลองการเคลื่อนที่ของเรดาร์บีมในชั้นบรรยากาศนั้นจะสามารถคำนวณถึงระยะทางและความสูงของเรดาร์บีมในแต่ละระยะการวัดตามมุมยกและมุมอչมัทที่กำหนดเข้าไปในโปรแกรมที่พัฒนาขึ้นมาได้



ภาพที่ 3.32 ลักษณะการวัดตามมุมยก (Elevation) เป็นระยะทางใดๆ (Ray) จากแนวรัศมี โดยยิ่งระยะทางที่ใกลอก็ไปจากตัวสถานีเรดาร์จะทำให้ระดับความสูงจากพื้นโลกห่างไปเนื่องจากความโค้งของโลก (Edge software)

$$H = (r^2 + R'^2 + 2 r R' \sin(\phi))^{1/2} + R' + H_0$$

โดยที่ H คือ ความสูงของของ gate, r คือระยะทางของ gate, ϕ คือมุมยก, R' คือ $4/3$ ของรัศมีโลก, H_0 คือความสูงของสถานีเรดาร์

เมื่อนำระยะและความสูงใดๆที่คำนวณได้มาทำการเบรี่ยนเที่ยบกับค่าความสูงจากแบบจำลองภูมิประเทศจะทำให้สามารถกำหนดค่าการบดบังอุกมาได้ ซึ่งค่าที่ได้นี้จะเป็นบวกถึงคุณภาพของการตรวจวัดเรดาร์ในบริเวณพื้นที่นั้นๆได้เป็นอย่างดีว่าควรจะมีการปรับปรุงหรือไม่

3.5.4 การโมเสคเดкарต์

การโมเสคเดкарต์เป็นสิ่งจำเป็นเมื่อมุ่งวิเคราะห์พื้นที่ลุ่มน้ำที่มีขนาดใหญ่ โดยที่การสังเกตการณ์ด้วยเดкарต์จากสถานีเดียวันนี้ไม่สามารถที่จะครอบคลุมพื้นที่ศึกษาได้ ซึ่งการโมเสคหรือนำเดкарต์หลายสถานีมาต่อ กันนั้นจะทำให้ได้แผนที่ฝนประมาณค่าจากเดкарต์ตรวจสอบ อย่างไรก็ตามยังไม่มีโปรแกรมสำเร็จรูปที่จะใช้ในการต่อข้อมูลเดкарต์เข้าเป็นแผนที่ด้วยกันที่สามารถจัดการข้อมูลหลายช่วงเวลาได้ ดังนั้นจำเป็นที่จะต้องมีการพัฒนาระบวนการในการต่อโมเสคข้อมูลเพื่อให้ง่ายต่อการนำไปใช้ในระบบ GIS ผู้วิจัยจึงพัฒนากระบวนการโมเสคข้อมูลเดкарต์ด้วยภาษาไพธอนในระบบ GIS รหัสเปิดด้วยการใช้ PyQGIS ใน QGIS ซึ่งเวอร์ชันของไพธอนที่ใช้นี้เป็นเวอร์ชัน 3 ซึ่งถือว่า มีหลักไวยากรณ์ในการเขียนที่แตกต่างจากเวอร์ชัน 2 อย่างมาก มีการปรับปรุงให้มีความเร็วของอัลกอริทึ่มในการเรียกใช้แต่ละโมดูลของ PyQGIS ที่แตกต่างจากเวอร์ชันก่อน ซึ่งในบทตอนนี้จะกล่าวถึงผลการพัฒนาวิธีการโมเสคเดкарต์หรืออัลกอริทึ่มใน PyQGIS ต่อไป การโมเสคนี้ได้ใช้กระบวนการแบบแก้เตอร์ในการต่อข้อมูลที่คาดในแต่ละช่วงเวลาราย 15 นาที ส่วนฝนสะสมนั้นใช้การแปลงข้อมูลจากฝนราย 15 นาทีเป็นฝนสะสมตามเวลาที่ต้องการ เพื่อให้มีการคำนวณที่มีประสิทธิภาพจึงใช้การคำนวณแบบราสเตอร์ในการสร้างฝนสะสม

ในการทดสอบค่าความสัมพันธ์ระหว่างค่าการสะท้อนเรเดาร์กับฝนภาคพื้นดินหรือ Z-R relationship นั้นในการวิจัยครั้งนี้เลือกค่า Z-R มา 3 ค่าที่เป็นที่นิยมใช้กัน เพื่อหาความสัมพันธ์ค่าฝนที่ประมาณค่าจากค่าสมการ Z-R โดยที่ให้ฝนได้ใกล้เคียงกับฝนที่วัดจากสถานีตรวจอัตราฝนภาคพื้นดิน ดังมีรายละเอียดสมการ Z-R ดังต่อไปนี้

| Z-R relationship | Equation | Description |
|----------------------------------|----------------|--|
| 1.Marshall/Palmer Stratiform Z/R | $Z=200R^{1.6}$ | ความสัมพันธ์ของค่าการสะท้อนเรเดาร์ของฝนแบบ stratiform |
| 2.Rosenfeld Tropical Z/R | $Z=250R^{1.2}$ | ความสัมพันธ์ของค่าการสะท้อนเรเดาร์ของฝนแบบในเขตตropical |
| 3.Summer Deep Convection Z/R | $Z=300R^{1.4}$ | ความสัมพันธ์ของค่าการสะท้อนเรเดาร์ของฝนแบบในช่วงพาดปกติที่ก่อตัวในแนวตั้งในช่วงฤดูร้อน |

3.5.5 การตรวจสอบผลการโมเดลด้วยข้อมูลฝนสถานี (validation)

ฝนสถานีถูกนำมาใช้ในการตรวจสอบค่าความเบี่ยงเบน (bias) ของผลลัพธ์การโมเดลในทุกเวลา โดยทำการสักด้าฝนเรดาร์ที่มีพิกัดตรงกันกับพิกัดของฝนสถานี แล้วนำมาหาค่าตามสมการต่อไปนี้

1. ค่าสัมพันธ์

$$R = \frac{\sum_{i=1}^n (R_i - \bar{R}_i)(G_i - \bar{G}_i)}{\sqrt{\sum_{i=1}^n (R_i - \bar{R}_i)^2 * (G_i - \bar{G}_i)^2}}$$

2. ค่า Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (R_i - G_i)^2}$$

3. ค่า Mean Field Bias

$$(G/R)_{MFB} = \frac{\sum_{i=1}^n G_i}{\sum_{i=1}^n R_i}$$

4. ค่า Relative error

$$RE = \frac{\sum_{i=1}^n |R_i - G_i|}{\sum_{i=1}^n G_i}$$

โดยที่ค่า R คือฝนที่ได้จากการประมาณค่าจากค่าการสะท้อนของเรดาร์ด้วยความสัมพันธ์ของ Z-R ;

G คือ ปริมาณฝนที่วัดได้จริงจากถังวัดฝนอัตโนมัติ

นอกจากนี้ยังได้นำค่าสถิติใน Mean Field Bias มาวิเคราะห์ค่าความเบี่ยงเบนของฝนเรดาร์ในเชิงพื้นที่อีกด้วย

บทที่ 4 ผลการศึกษา

ในบทนี้จะกล่าวถึงผลการศึกษาที่ได้ทำการพัฒนาระบวนการเพื่อพัฒนาวิธีการโมเสคข้อมูลเรดาร์จากหน่วยงานกรมอุตุนิยมวิทยาในระบบ GIS เพื่อมุ่งสร้างข้อมูล汾ประมาณค่าในพื้นที่ระดับลุ่มน้ำในโปรแกรม PyQGIS ด้วยภาษาไพธอนในพื้นลุ่มน้ำแม่น้ำและน่าน อัลกอริทึมที่ได้ถูกพัฒนาในการโมเสคข้อมูลค่าการสะท้อนของเรดาร์เพื่อแปลงข้อมูลให้เป็นกริด โดยได้ดัดแปลงความสามารถนี้ไปปรับใช้ในพื้นที่ลุ่มน้ำอื่นๆได้ ในบทนี้จะประกอบไปด้วย การพัฒนาโปรแกรมเพื่อเตรียมข้อมูลเรดาร์ก่อนการประมวลผล ผลงานของการพัฒนาระบวนการตรวจหา radar flare ผลการพัฒนาการวิเคราะห์การบดบังสำนึมด้วยลักษณะภูมิประเทศ ผลการพัฒนาอัลกอริทึมโมเสค และผลการตรวจสอบความเบี่ยงเบนของข้อมูล汾ประมาณค่าจากเรดาร์ตรวจอากาศ ดังมีรายละเอียดดังหัวข้อต่อไปนี้

4.1 การพัฒนาโปรแกรมเพื่อเตรียมข้อมูลเรดาร์ก่อนการประมวลผล

ขั้นแรกของการพัฒนาอัลกอริทึมนั้น จำต้องมีพัฒนาระบวนการเพื่อเตรียมความพร้อมของข้อมูล ก่อนที่จะเริ่มกระบวนการประมวลผลข้อมูลเรดาร์ ก่อนอื่นต้องทำการตรวจสอบข้อมูลของสถานีที่จะทำการโมเสคนั้นมีครบหรือไม่ในเวลาเดียวกัน และทำการสร้างฟังค์ชันที่จะคัดลอกshapefileต้นฉบับ ให้เป็นไฟล์ใหม่ในmemoryเพื่อจะทำการประมวลผลจากไฟล์ที่ถูกคัดลอกนี้ จากนั้นจะต้องแปลง ข้อมูลมีลักษณะเป็นเวกเตอร์รูปปั๊ดที่ยังมีระบบพิกัดเป็นแบบ NAD83-Geographic ระบบนี้เป็นระบบ พิกัดภูมิศาสตร์ซึ่งสร้างความยุ่งยากในการประมวลผลก่อระบบพิกัดที่เป็นแบบเมตริกหรือ UTM ดังนั้นการแปลงค่าระบบพิกัดให้อยู่ในรูป UTM ตามโซนของลุ่มน้ำที่กำลังศึกษาจะทำให้การ ประมวลผลในขั้นตอนต่อไปทำได้ง่ายขึ้นดังโค้ดภาพด้านล่าง

```
path = "D:/Python37/research/0.2/GISproject/PyQGIS/3dmap_project/1kadar.php?1&shp:1&ontime/0jhs/*file{0}*.shp"
pat = "/*file{1}.shp"
file = fmatch.filter(fns2, pat)

path_rad = "D:/Python37/research/0.2/GIS/GISproject/PyQGIS/3dmap_project/1kadar.tif"
patch_beamblock = "D:/Python37/research/0.2/GIS/GISproject/PyQGIS/3dmap_project/1kadar.tif"
rad_name='1RC'
epsg = 32647 #กำหนดรหัสพิกัดที่ต้องการ
binSize=1000.0
sta_coor=(106.1179637,16.774999) #กำหนดจุดที่ต้องการshapefile
shapefile_w=EPSG:4326 ->WGS84_epsg:32647_UTM47N
path_reprojectRad(path_rad,rad_name)
```

ภาพที่ 4.1 โค้ดแสดงการหาไฟล์และระบบเส้นโครงแผนที่ให้เป็น UTM โซน 47N โดยเรียกใช้ฟังค์ชัน ที่ได้พัฒนาขึ้นมา

ภาพด้านบนจะเห็นว่ามีการเรียกใช้ฟังค์ชัน reprojectRad ที่ได้พัฒนาขึ้นมาเพื่อแปลงระบบเส้นโครง แผนที่จาก NAD83 ไปเป็น UTM WGS84 โซน 47N

```

def reprojectRad(path_rad,rad_name):
    #แปลงพิกัดจาก NAD83 เป็น WGS84 อีก UTM47N
    print('reprojection....'+rad_name)
    parameter = {'INPUT': path_rad, 'TARGET_CRS': 'EPSG:32647',
    'OUTPUT': 'memory:Rad_Reprojected'}
    result = processing.run('native:reprojectlayer', parameter)
    path_rad=result['OUTPUT']

    return path_rad

```

ภาพที่ 4.2 ฟังก์ชัน reprojectRad

ภาพด้านบนแสดงฟังก์ชัน reprojectRad ที่เรียกใช้เครื่องมือ reproject ของ QGIS ใน processing toolbox เพื่อแปลง shapefile เردาร์ไปสู่เส้นโครงแผนที่ใหม่ ต่อไปอธิบายผลของการพัฒนาส่วนของการตรวจหา radarflare

4.2 การพัฒนากระบวนการตรวจหา radar flare

การตรวจหา radar flare ถูกออกแบบเป็นสองส่วน โดยหลักการในการตรวจหานั้นได้มาจากการสังเกตพบว่าค่าการสะท้อนของส่วนที่เป็น flare เมื่อมองด้วยตาเปล่าจะพบว่ามีความสัมพันธ์กับระยะทางนั้นคือ เมื่อระยะทางจากแนวรัศมีเพิ่มขึ้นค่าการสะท้อนที่ตรวจวัดได้จะเพิ่มขึ้นด้วยในลักษณะเชิงเส้นตรงหรือ linear ดังนั้นการใช้ linear regression ตรวจในมุมใดๆ ก็สามารถหา flare เหล่านั้นได้ อีกประการคือ flare ที่พบจะมีทั้งขนาดที่สั้นและขนาดที่ยาว ในที่นี้การหา flare ที่ยาวจะช่วยปรับปรุงคุณภาพในการโมเดลให้ดีขึ้น ซึ่ง flare ที่ยาวนั้นมักจะมีความยาวมากกว่า 80 % ของความยาวจากแนวรัศมีจากสถานี雷达 ดังนั้นเกณฑ์ทั้งสองส่วนนี้สามารถนำมาประยุกต์ใช้ในการหา flare ได้ การพัฒนาในส่วนนี้ได้ออกแบบโปรแกรมออกแบบหา flare ทั้งหมด 2 รอบ ซึ่งรอบแรกจะเป็นการหา flare ทั่วทุกมุมของข้อมูล雷达ก่อนแล้วทำการเก็บมุมที่ถูกระบุว่าเป็น flare และนำมุมเหล่านั้นมาหามุมข้างเคียงที่เพื่อค้นหาว่าเข้าข่ายเกณฑ์ที่ระบุว่าเป็น flare หรือไม่ ถ้าใช่ก็จะทำการ mark ในค่าแอลทริบิวต์ว่าเป็น flare ดังภาพด้านล่างจะเป็นโค้ดในส่วนการหา flare รอบแรก

```

def addRadialAngle(phLayer,radialAngle):
    #เพิ่มค่า radial angle ใน spatial index ไว้
    print('เพิ่ม radial angle ใน spatial index ไว้')
    #ตรวจสอบว่า layer ที่ให้มาเป็นพีลเยอร์
    idx = phLayer.fields().indexOf('radialAng')
    values = phLayer.uniqueValues(idx)
    #เพิ่ม spatial index ให้กับทุกฟีเจอร์ที่มี radial angle
    all_features = {}
    index = QgsSpatialIndex() + Spatial index
    for ft in phLayer.getFeatures():
        index.insertFeature(ft)
        all_features[ft.id()] = ft
    i=0
    #บันทึก radial angle ของแต่ละฟีเจอร์
    flare_ang=[]
    for ang in values:
        phLayer.selectByExpression('`radialAng` = {}'.format(ang))
        inds=phLayer.selectedFeatureIds()
        #บันทึก radial angle ของแต่ละฟีเจอร์ที่มี radial angle ใน spatial index ที่ได้มา
        val_list=[]
        #คำนวณหาค่าเส้นเชื่อมต่อที่จะใช้ในการสร้างพื้นที่สูญเสียใน spatial index
        for id in inds:
            val_list.append((id,all_features[id]['value']))
        dv=np.array(val_list) #แปลงเป็น array
        slope, intercept, r_value, p_value, std_err = linregress(dv[:,0], dv[:,1])
        ct=len(inds)
        #ตั้งค่าพื้นที่สูญเสียที่จะใช้ในการสร้างพื้นที่สูญเสียใน spatial index
        j=0
        for k,f in enumerate(phLayer.selectedFeatures()):
            if ((ct/dist)*100.0)>=PC_BinWithVal and r_value>=Corr:
                #กรณีที่ radial angle 999.0 ไม่เป็น flare ให้ตัดต่อ
                attrs=[1:-999.0,4:k,5:ct,6:(ct/dist)*100.0,7:str(slope),8:str(r_value),9:1]
                phLayer.dataProvider().changeAttributeValues({f.id():attrs})#f=flare
                if j==0:flare_ang.append(ang) #บันทึก radial angle ที่ได้มา
            else:
                attrs=[4:k,5:ct,6:(ct/dist)*100.0,7:str(slope),8:str(r_value),9:0]
                phLayer.dataProvider().changeAttributeValues({f.id():attrs})#0=ok
            j+=1
    return phLayer,flare_ang,values

```

ภาพที่ 4.3 โค้ดการหา flare รอบที่ 1

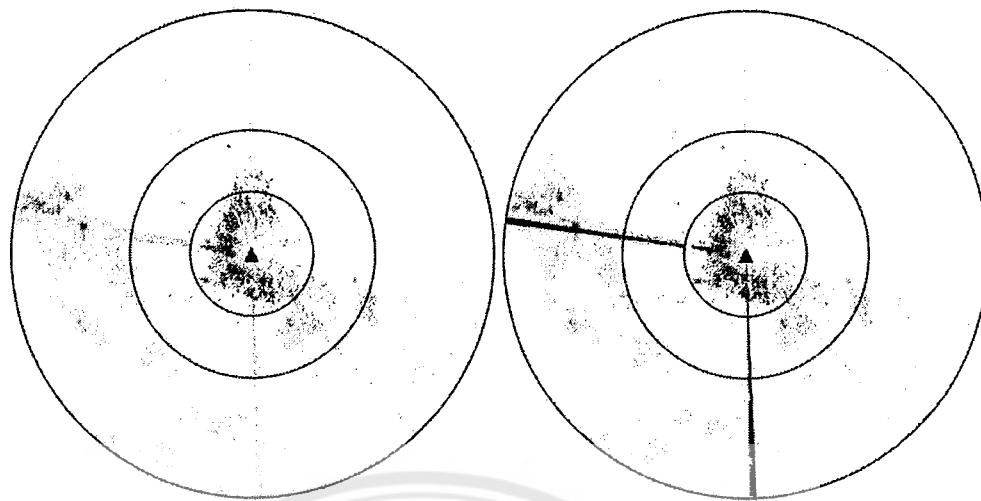
```

def radarFlareDetectionRound2(phLayer, PC_BinWithVal, Corr, flare_ang, values):
    print('... radar flare ตรวจสอบดี? ...')
    phLayer.startEditing()
    # แปลงพิกัดเป็น array
    ang_arr=np.array(values)
    # ตั้งชื่อ array ใหม่ Flare ตามเดิม
    flare_list=[]
    for ang in flare_ang:
        w_check=ang in ang_arr
        if w_check==False: continue # ตรวจสอบว่าจุดที่ให้มา flare หรือไม่
        id=np.where(ang_arr==ang) # หาจุดที่เก็บ flare ของบันทึกเดิม
        id=id[0]
        # กรณี flare ไม่ได้บันทึกบนกระดาษแล้ว
        if id>0 and id<(len(values)-1):
            pre_id=id-1 # จัดลักษณะ flare
            pot_id=id+1 # จัดลักษณะ flare
            flare_list.append(ang_arr[pre_id])
            flare_list.append(ang_arr[pot_id])
        # กรณีบันทึกอยู่ในกระดาษแล้ว
        if id==0: flare_list.append(ang_arr[id+1]) # ให้จัดลักษณะ +1
        # กรณีบันทึก flare ไม่บันทึกห่างจาก sweep ให้จัดลักษณะ -1
        if id==len(values)-1: flare_list.append(ang_arr[id-1])
        # flare_list ลักษณะจะมีตัวเลขแทน flare ที่ต้องคำนึงถึง linear regression ในระบบ
        # เช่น range ต้องห่างกับ flare ในระบบ จึงจะได้ปีกของเรือตัวสีฟ้าก้ามตัว
        # รูปต้องห่าง flare ด้วย
        for ang in set(flare_list): # ตั้ง set เพื่อจัดซ้ำตัวบันทึก unique ด้วย
            # ตรวจสอบว่าบันทึกที่ "002" หลังจาก flare ขาดหายไป
            phLayer.selectByExpression('`radialAng` = () AND `Flare1` != {} AND
            `PC_BinWithVal` >= () AND `Corr` >={}'.format(ang,1,PC_BinWithVal,Corr))
            for f in phLayer.selectedFeatures():
                phLayer.changeAttributeValue(f.id(),10,2) # 2=flare ตั้งค่าเป็นค่าเดิม "flare2"
                phLayer.changeAttributeValue(f.id(),9,2) # 2=flare ตั้งค่าเป็นค่าเดิม "flare1"
                # ปรับค่าให้เท่ากับ ให้แต่ละ dbZ ลงตัว "value" และตั้งค่าให้เป็น valueFlare
                phLayer.changeAttributeValue(f.id(),1,-999.00)

    phLayer.commitChanges()
    return phLayer

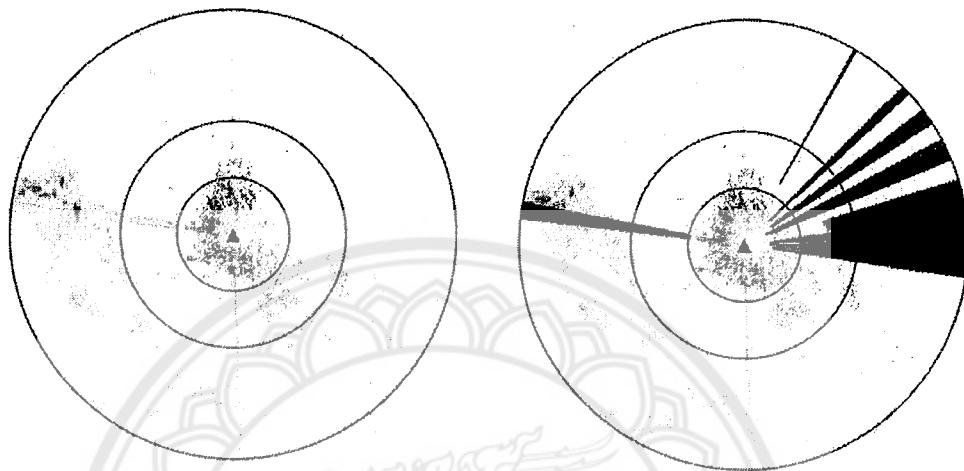
```

ภาพที่ 4.4 โค้ดการหา flare รอบที่ 2



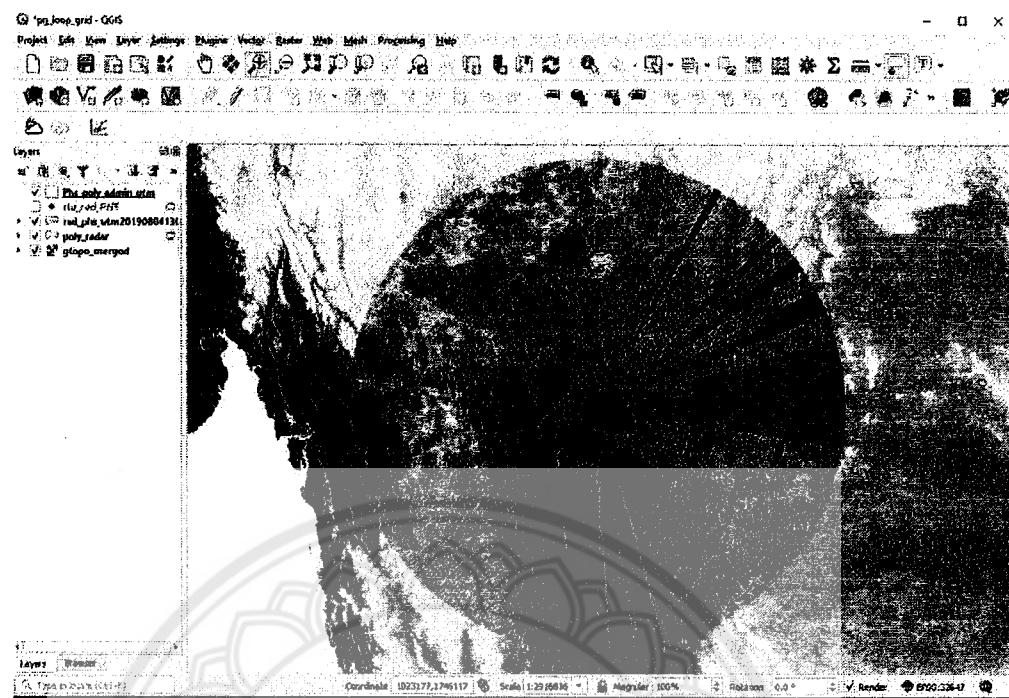
ภาพที่ 4.5 ผลการตรวจพบ radar flare ของสถานีพิษณุโลกเวลา 00:00 ของวันที่ 15 กรกฎาคม พ.ศ.2561 (ซ้าย)ภาพทั้งต้นที่พบว่ามี flare อยู่ทางตะวันตกและให้ข้อของสถานีเรดาร์ (ขวา) ผลการตรวจพบ flare สีแดงคือผลจากการบแกร สีน้ำเงินคือผลจากการบที่สอง
ภาพด้านบนแสดงให้เห็นถึงผลการตรวจพบ flare จากทั้งสองรอบ ในรอบแรกจะแสดงเป็นสีแดง ซึ่ง มุมในรอบแรกจะถูกเก็บเพื่อนำไปหาเฉพาะในการหา flare ในรอบที่สองเพื่อเพิ่มความเร็วในการ ค้นหา ซึ่งจากการทดลองจะพบว่า flare ในรอบแรกนั้นจะยังหาไม่พบทั้งหมด ทำให้จำเป็นต้องสร้าง กระบวนการกรองและค้นหา flare ที่เข้ากันที่อีกครั้ง

4.3 การวิเคราะห์การบดบังของลำบีมจากลักษณะภูมิประเทศ



ภาพที่ 4.6 ผลการวิเคราะห์การบดบังของลำบีมด้วยลักษณะภูมิประเทศ ของสถานีพิชณุโลกเวลา 2018/07/15/0000 (ซ้าย)ภาพต้นฉบับ (ขวา) ผลการวิเคราะห์การบดบัง

การวิเคราะห์การบดบังของลำบีมด้วยลักษณะภูมิประเทศนี้ได้กำหนดเปอร์เซ็นต์การบดบังที่ค่า 50 % นั่นคือ หากความสูงของภูมิประเทศเท่ากับหรือมากกว่าความสูงของศูนย์กลางลำบีม จะทำการ mark เ雷达标์ bin น้ำๆและ bin ที่อยู่ด้านหลังถัดไปจนถึง bin สุดท้ายของแนวรัศมีสูงสุดของการสังเกตภารณ์ ด้วย雷达标์ กำหนดให้เป็นส่วนที่ถูกบดบังในค่าเออทริบิวต์ของเวกเตอร์รูปปิด การบดบังนี้ไม่ได้นำไป เป็นปัจจัยที่ใช้ในการโมเสคด้วย เพียงแต่ต้องการวิเคราะห์ว่าคุณภาพของข้อมูลที่ใช้ในการโมเสค มี คุณภาพดีหรือไม่อย่างไร จากภาพด้านบนในส่วนของ雷达标์พิชณุโลกจะพบว่า มีการบดบังที่มาก ในทางตะวันออกซึ่งไม่ได้มีพื้นที่ลุ่มน้ำยม แต่เป็นพื้นที่ของลุ่มน้ำ่าน ส่วนทางตะวันตกของ雷达标์ พิชณุโลกที่เป็นพื้นที่ของลุ่มน้ำยมจะมีการบดบังพิยงเล็กน้อย บริเวณพื้นที่ที่ลำบีมถูกบดบังนี้จะมีค่า การสะท้อนของสัญญาณ雷达标์ที่อ่อนกว่าพื้นที่ที่ไม่ถูกบดบัง ดังนั้นในอนาคตส่วนที่ถูกบดบังนี้อาจหา วิธีการในการปรับแก้หรือใช้มุมยกที่สูงขึ้นไป ซึ่งยังต้องคำนึงถึงค่าผลลัพธ์สุดท้ายของการปรับแก้นี้ ด้วย แต่เนื่องจากงานวิจัยครั้งนี้ไม่ได้มุ่งที่จะปรับแก้ใดๆ กับลำบีมที่ถูกบดบัง ในอัลกอริทึมจึงไม่ได้ พัฒนากระบวนการปรับแก้ที่ในโค้ด PyQGIS



ภาพที่ 4.7 การบดบังของลำบีมด้วยภูมิประเทศสถานีเรดาร์พิษณุโลกแสดงใน QGIS ซึ่งอนทับกับข้อมูลเรดาร์วันที่ 4 สิงหาคม 2562 เวลา 13.00 น.

ภาพด้านบนแสดงถึงผลของการวิเคราะห์การบดบังของลำบีมด้วยภูมิประเทศ โดยใช้มุมยกแรกของสถานีเรดาร์พิษณุโลกที่มีมุม 0.5 องศา โดยเส้นทางแสดงถึงตำแหน่งที่มีการบดบังด้วยภูมิประเทศที่ใช้ข้อมูล DEM ในการคำนวณร่วมกับตำแหน่งความสูงของลำบีม จึงการคำนวณในครั้งนี้จะกำหนดให้บีมที่ถูกบดบังนั้นคือบีมที่มีความสูงของกึ่งกลางบีมต่ำกว่าหรือเท่ากับภูมิประเทศในแต่ละตำแหน่ง radar bin ที่กำลังพิจารณาให้เป็นลำบีมที่ถูกบดบัง แล้วแล้ว radar bin ที่เหลืออยู่ที่กำลังพิจารณานั้นเป็น radar bin ที่ถูกบดบังเข่นกัน อย่างไรก็ตามการการบดบังของลำบีมจะมีตำแหน่งที่คงที่ไม่เปลี่ยนแปลงตามห้วงเวลา เพราะภูมิประเทศไม่มีการเปลี่ยนแปลงอย่างฉับพลัน อย่างไรก็ตามหากมีการเปลี่ยนแปลงมุมยก โค้ดที่พัฒนานี้ก็ยังสามารถที่จะปรับเปลี่ยนได้โดยง่าย



ภาพที่ 4.8 ผลการวิเคราะห์การบดบังของลำบีมด้วยภูมิประเทคโนโลยีของสถานีเขียงรายและพิษณุโลกในลุ่มน้ำยมและน่าน พื้นที่บีมที่ถูกบดบัง 50% ของลำบีมโดยภูมิประเทคโนโลยีแสดงเป็นสีเทา ภาพด้านบนเมื่อนำเรดาร์ทั้งสองสถานีมาวิเคราะห์ดูการบดบังของลำบีมในลุ่มน้ำยมและน่าน โดยใช้การบดบัง 50 % ของลำบีมในมุมยกแรกของทั้งสองสถานีก่อตัวคือ มุมยกที่ 1.0 องศาและ 0.5 องศาของสถานีเขียงรายและพิษณุโลกตามลำดับ จะพบว่าลุ่มน้ำยมนั้นมีการบดบังเพียงเล็กน้อยในทางทิศตะวันตกของเรดาร์ ส่วนเรดาร์พิษณุโลกในทางทิศตะวันออกนั้นจะพบว่ามีการบดบังทำให้ลำบีมจากพื้นที่ภูเขาทางทิศตะวันออกของสถานี ทางแก้คืออาจหาเรดาร์ซ้ายนามาโมเสคบริเวณที่บดบังนี้หรือใช้มุมยกที่สูงกว่านี้ในมุมกดไปเฉพาะบริเวณนี้หรือที่เรียกว่า *rsndocappi*

```

def ExtractBeamBlockByCentroid(phLayer,rLayer):
    print('เริ่ม BeamBlock....')
    phLayer.startEditing()
    for f in phLayer.getFeatures():
        #รับค่าจาก centroid ของฟีล
        pts = f.geometry().centroid().asPoint() #get จุดศูนย์กลางของฟีลที่บันทึก
        #จัดทำบล็อกbeamblock
        #val คือbeamblock หรือพื้นที่ที่มีความสูงมากที่สุดใน sample
        val, res = rLayer.dataProvider().sample(QgsPointXY(pts[0],pts[1]), 1)

        #อัปเดตบล็อกbeamblock ในฟีลphLayer
        if np.isnan(val):
            phLayer.changeAttributeValue(f.id(),11,0)
        else:
            phLayer.changeAttributeValue(f.id(),11,val)
    phLayer.commitChanges()
    print('เสร็จแล้ว BeamBlock....')
    return phLayer

```

ภาพที่ 4.9 โค้ดการหาบริเวณที่ลำบีมของเรดาร์ถูกบดบังด้วยภูมิประเทศ

ภาพด้านบนเป็นโค้ดที่ได้จากการพัฒนาในการหาลำบีมของเรดาร์ถูกบดบังด้วยภูมิประเทศ ซึ่งจะใช้ DEM ที่อ่านเข้ามาเก็บไว้ใน memory จากนั้นจะทำการค้นหาทุกรูปปิดในแต่ละมุม โดยสกัดค่าพิกัดศูนย์กลางรูปปิดของค่าการสะท้อนเรดาร์หรือ bin นั้นๆ ออกมายื่นนำมาเปรียบเทียบกับความสูงที่พิกัดตรงกันโดยใช้ฟังก์ชัน QgsPointXY ของ PyQGIS ในการเบรี่ยนเทียบว่าความสูงของข้อมูลสองชุด เมื่อพบว่ามีการบดบังจะทำการอัพเดทในแอธิบิวต์ของ bin เหล่านั้นทันทีด้วยฟังก์ชัน changeAttributeValue ของ PyQGIS

4.4 การพัฒนาโค้ด PyQGIS เพื่อสร้างแผนที่โมเสคเดาร์ในลุ่มน้ำ

```
#ฟังก์ชันสำหรับจัดของการไม้เต็ม
+ def calParaGrid(xmin,xmax,ymin,ymax,ds_s):
+
+-----#
+ def createGrid(numCols,numRows,llx,lly,ll,ul,ur,lr,ds_s,pLayer,poly,pr,gridVal):
+
+-----#
+ def clipRadarByProvince(pLayer,prLayer,pcLayer,pr):
+
#-ดำเนินการ centroid ของกริด C ให้แก่ในพื้นที่กริด A หรือ B เพื่อสังเกตค่าพื้นที่มาสู่กริด C
+ def updateFieldOutputByIntersectCentroid(paLayer,pbLayer,pcLayer,rd):
+
#-
```

ภาพที่ 4.10 พังค์ชั่นหลักที่ใช้ในการโมเสค

ภาพด้านบนเป็นพังค์ชั่นหลักที่ได้พัฒนาในกระบวนการโมเสคผ่านกระบวนการค่าจากค่าสัญญาณการสะท้อนจากเรดาร์ตรวจอากาศ ซึ่งข้อมูลที่ได้นี้หลักจากผ่านการตรวจวัดค่า radar flare และผ่านการวิเคราะห์การบดบังของภูมิประเทศที่มีผลต่อจำเป็นในแต่ละมุมของมัทแอล ข้อมูลแต่ละสถานีจะพบว่า ยังมีขนาดที่ใหญ่กว่าขนาดพื้นที่ลุ่มน้ำที่ต้องการศึกษามาก ซึ่งเมื่อนำมาประมวลผลแล้วพบว่าใช้เวลาในการคำนวนค่อนข้างสูง ดังนั้นผู้วิจัยจึงพัฒนากระบวนการตัดเฉพาะพื้นที่ที่การสังเกตการณ์ของเรดาร์ครอบคลุมพื้นที่ลุ่มน้ำ โดยพัฒนาเป็นพังค์ชั่นใน PyQGIS ซึ่งจะต้องทำการคำนวนพารามิเตอร์โดยพัฒนาพังค์ชั่นที่ชื่อว่า calParaGrid เพื่อให้ได้ลักษณะทาง geometry ของลุ่มน้ำจากการคำนวนค่าที่เก็บไว้ใน shapefile ลุ่มน้ำ ทำให้สามารถสร้างกริดที่เป็นเวกเตอร์ที่ใช้ในการเก็บผลลัพธ์ในกระบวนการโมเสคนี้ได้โดยใช้พังค์ชั่น createGrid เพื่อคำนวนจำนวนกริดที่จะใช้ให้หักลุมพื้นที่ลุ่มน้ำ เมื่อได้ข้อมูลที่เรดาร์ที่ตัดของแต่ละสถานีแล้ว ก็นำเข้าสู่กระบวนการโมเสคโดยได้พัฒนาพังค์ชั่นที่ชื่อว่า updateFieldOutputByIntersectCentroid ซึ่งมีกระบวนการโมเสคที่ได้รับการพัฒนาอยู่ข้างในพังค์ชั่นนี้

```

def clipRadar(pLayer, prLayer, pclayer, pr):
    # ตั้งค่าไฟล์ชื่อพื้นที่ที่ต้องการใช้สำหรับพื้นที่เชิงเส้นที่ต้องการ
    # pLayer = ชั้นพื้นที่ชั้นเดียว, prLayer = ชั้นเดียวที่ต้องการ
    # pclayer = shapefile ที่ต้องเก็บเฉพาะพื้นที่ที่มีค่าพื้นที่ดินมากกว่าพื้นที่
    # pr = dataProvider ของ pclayer

    # ตั้งค่าพื้นที่เชิงเส้นที่ต้องการ extent เพื่อนำไป query ของพื้นที่เชิงเส้น
    poly_p = QgsFeature()
    offset = 2000
    ext = prLayer.extent()
    (xmin, xmax, ymin, ymax) = (ext.xMinimum() - offset, ext.xMaximum() + offset, \
                                 ext.yMinimum() - offset, ext.yMaximum() + offset)

    # สร้าง list ของจุดที่ต้องการตัดต่อให้มีรัศมีเพียง 2000
    points = [QgsPointXY(xmin, ymin), QgsPointXY(xmin, ymax), \
              QgsPointXY(xmax, ymax), QgsPointXY(xmax, ymin)]

    # ตั้งค่าพื้นที่เชิงเส้นที่ต้องการตัดต่อให้มีรัศมีเพียง 2000
    poly_p.setGeometry(QgsGeometry.fromPolygonXY([points]))

    # ตัดพื้นที่เชิงเส้นที่ต้องการตัดต่อให้มีรัศมีเพียง 2000
    inBound = poly_p.geometry().boundingBox()
    i = 0
    for ft in pLayer.getFeatures():
        if ft.geometry().intersects(inBound):
            poly_cri = QgsFeature()  # ตั้งค่าตัวแปร
            # ตัดส่วนที่ไม่ต้องการของพื้นที่เชิงเส้นที่ต้องการตัดต่อให้มีรัศมีเพียง 2000
            poly_cri.setGeometry(ft.geometry())
            # update feature
            poly_cri.setAttributes([i, ft['value'], ft['heightASL']])
            pr.addFeatures([poly_cri])  # ตั้งค่าพื้นที่เชิงเส้นที่ต้องการตัดต่อให้มีรัศมีเพียง 2000 ให้กับ dataProvider และ pr
            pclayer.updateExtents()  # ตั้งค่าพื้นที่เชิงเส้นที่ต้องการตัดต่อให้มีรัศมีเพียง 2000 ให้กับ map
            i += 1

    return pclayer

```

ภาพที่ 4.11 ฟังก์ชัน clipRadar

จากภาพด้านบน ฟังก์ชัน clipRadar จะทำการ offset 2000 เมตรจากค่า extent ของ shapefile ที่ใช้ การคำนวนจะทำการลูปข้อมูลทุก features ด้วยการใช้ getFeatures() ซึ่งค่าแอทริบิวท์จะถูกเก็บค่าการสะท้อนและความสูงของลำบีนเพื่อนำไปประมวลผลต่อไป

```

def calParaGrid(xmin, xmax, ymin, ymax, ds_s):
    print('ตั้งค่าพื้นที่เชิงเส้นที่ต้องการตัดต่อให้มีรัศมีเพียง 2000')
    # ตั้งค่าพื้นที่เชิงเส้นที่ต้องการตัดต่อให้มีรัศมีเพียง 2000
    numCols = int((xmax - (xmin / ds_s)) // ds_s) + 1  # จำนวนค่าในแนวนอน // ต้องการตรวจสอบ +1 ต้องการหักห้ามเศษส่วน
    print('จำนวนค่าในแนวนอน', numCols)
    numRows = int((ymax - (ymin / ds_s)) // ds_s) + 1  # จำนวนค่าในแนวตั้ง
    print('จำนวนค่าในแนวตั้ง', numRows)

    # กำหนดตัวต่อสุดท้าย
    llx = xmin - (xmin / ds_s)
    lly = ymin - (ymin / ds_s)

    # set dimension first polygon at lower left
    ll, ul, ur, lr = [llx, lly], [llx, lly + {ds_s * 2}], \
                     [llx + {ds_s * 2}, lly + {ds_s * 2}], [llx + {ds_s * 2}, ds_s]

    return numCols, numRows, llx, lly, ll, ul, ur, lr
    print('ตั้งค่าพื้นที่เชิงเส้นที่ต้องการตัดต่อให้มีรัศมีเพียง 2000')

```

ภาพที่ 4.12 ฟังก์ชัน calParaGrid

ภาพด้านบนเป็นโค้ดของฟังก์ชัน calParaGrid ซึ่งได้คำนวนโดยใช้ค่า extent ที่ได้จากฟังก์ชันก่อนหน้านี้ ทำการคำนวนจำนวนกริดที่จะต้องใช้โดยเริ่มนับจากไปเพื่อใช้ในการคำนวนในฟังก์ชัน createGrid

```

def createGrid(numCols,numRows,llx,lly,ll,ul,ur,lr,ds_s,pLayer,poly,pr,gridVal):
    print('เริ่มต้นการสร้างกริด')
    i=0
    for r in range(numRows):
        for c in range(numCols):
            ll[0],ll[1] = llx+(c*ds_s),lly+(r*ds_s)
            lr[0],lr[1] = ll[0]+ds_s, ll[1]
            ur[0],ur[1] = lr[0],lr[1]+ds_s
            ul[0],ul[1] = ll[0],ur[1]

            #list ของพิกัดของมุมกริดที่นำมาใช้ในการสร้างรูป多边形
            points = [QgsPointXY(ll[0],ll[1]),QgsPointXY(ul[0],ul[1]),\
                      QgsPointXY(ur[0],ur[1]),QgsPointXY(lr[0],lr[1])]

            #สร้างรูป多边形และกำหนดพื้นที่และ周長ให้กับ poly สำหรับ
            poly.setGeometry(QgsGeometry.fromPolygonXY([points]))

            #calculate area and perimeter from geometry
            #คำนวณพื้นที่และ周長ของรูป多边形ที่ได้มาแล้ว
            area=(poly.geometry().area())/1000/1000
            peri=poly.geometry().length()

            #บันทึกพื้นที่และ周長ของรูป多边形
            poly.setAttributes([i,area,peri,gridVal])

            pr.addFeatures([poly]) #เพิ่มรูป多边形ที่ได้มาลงในdataProvider หรือpr
            pLayer.updateExtents() #อัปเดต周界

            i+=1

    print('สร้างกริดสำเร็จ')
    return pLayer

```

ภาพที่ 4.13 ฟังก์ชัน createGrid

ภาพด้านบนเป็นโค้ดของฟังก์ชัน createGrid ในที่นี้การคำนวนจะอยู่บนพื้นที่ UTM ซึ่งจะทำการลูปตามจำนวนแถวและหลักที่ได้รับค่ามาจากฟังก์ชัน calParaGrid แล้วทำการสร้างรูปปีดด้วยการใช้พิกัดของแต่ละมุมกริดที่เป็น points เพื่อนำมาสร้างเป็น polygon feature ด้วย QgsGeometry แล้วทำการอัปเดตค่าแอลหริบิท์ของแต่ละรูปปีดที่ได้สร้างด้วยการ setAttributes ผลลัพธ์ของฟังก์ชัน createGrid จะนำไปใช้ในการเก็บค่าการสะท้อนที่ได้จากการประมาณค่าจากจุด centroid ของ radar bin แต่ละสถานีในกระบวนการโมเดลต่อไป

```

#loop วนลูปที่ index ของ idW
for feat in pclayer.getFeatures():
    inGeom = feat.geometry()
    idsListA = indexA.intersects(inGeom.boundingBox())
    idsListB = indexB.intersects(inGeom.boundingBox())

    pt_c=feat.geometry().centroid().asPoint()

    #ค่าระยะห่างของ bin
    dist_val=[]

    #list ของ bin ที่มีค่า distance มากกว่า 0
    w_list=[]

    #loop วนลูปที่ index ของ idB
    for id in idsListA:
        pt_a = all_featuresA[id].geometry().centroid().asPoint()
        distance = QgsDistanceArea()
        d = distance.measureLine(pt_c, pt_a) #ระยะห่าง C กับ A

        if d>rd: continue

        if all_featuresA[id]['value']>7 and all_featuresA[id]['value']<55 and all_featuresA[id]['heightaboveasl']<5000:
            #ค่า distance ของ bin ที่มีค่า distance มากกว่า 0
            dist_val.append(d,all_featuresA[id]['value'])

    #list ของ bin ที่มีค่า distance มากกว่า 0
    if d>0:
        w=i/d
        w_list.append(w)
    else:
        w_list.append(0)

```

ภาพที่ 4.14 โค้ดโมเดลส่วนที่ 1

ภาพด้านบนเป็นโค้ดของการโมเดลส่วนที่ 1 รายละเอียดโค้ดเต็มๆได้ในส่วนภาคผนวก จากโค้ดจะพบว่าอัลกอริทึมจะถูกพัฒนาให้ลุลปแต่ละ features ของข้อมูลเรดาร์ bin ที่ตัดขอบเขตให้พอดีกับคุณน้ำ โดยจะมีการกรองค่าการสะท้อนให้อยู่ในช่วง 7-55 dBZ และให้มีความสูงของลำบีมที่น้อยกว่า 5,000 เมตรเพื่อหลีกเลี่ยงสถานะของเงี้ยงที่อยู่ในก้อนเมฆ แล้วจะทำการคำนวณค่า IDW โดยใช้คำนวณค่าน้ำหนักตามระยะทางของจุด centroid ของกริดผลลัพธ์ กับจุด centroid ของ radar bin ที่อยู่ในรัศมีที่กำหนด ตรงส่วนนี้จะเห็นว่าได้ใช้เงื่อนไขโดยเรียก feature ที่ได้ทำการสร้าง spatial index ไว้แล้วก่อนการคำนวณในลูปนี้ ทำให้ความเร็วในการคำนวณมีอัตราเร็วขึ้นกว่าการที่ใช้การคำนวณแบบลูปธรรมดา ซึ่งเป็นสิ่งที่นักพัฒนาอัลกอริทึมใน PyQGIS ต้องคำนึงถึงให้มาก

```

if len(w_list)==0:continue #ถ้าไม่มีค่าที่จัด C ให้ต่อไป
else:
    #ต้องลบ掉ค่า distance ของ bin ที่ไม่ต้องคำนวณ
    w_check=0 in w_list
    dv=np.array(dist_val) #แปลง成 array
    if w_check==True:
        idx=w_list.index(0) #หาตัวเลขที่บันทึกไว้ในรายการ distance ที่มีค่าเป็น 0
        z_idw=dv[idx,1] #รับค่าที่บันทึกไว้ในรายการ distance ที่มีค่าเป็น 0
        pclayer.changeAttributeValue(feat.id(),3,str(z_idw))
        pclayer.changeAttributeValue(feat.id(),4,len(dist_val))
    else:
        wt=np.transpose(w_list)
        z_idw=np.dot(dv[:,1],wt)/sum(w_list) #find product เพื่อคำนวณค่าที่บันทึกไว้ในรายการ distance ที่มีค่าเป็น 0
        #แหล่ง https://www.pythontutorial.net/numpy/numpy-dot-product.html
        pclayer.changeAttributeValue(feat.id(),3,str(z_idw))
        pclayer.changeAttributeValue(feat.id(),4,len(dist_val))

pclayer.commitChanges() #ต้องให้เปลี่ยนค่าที่บันทึกไว้ในรายการ
print('บันทึกค่าที่บันทึกไว้ในรายการ.....')
return pclayer

```

ภาพที่ 4.15 โค้ดโมเดลส่วนที่ 2

โค้ดด้านบน เป็นส่วนถัดมาจากการส่วนแรกในกระบวนการโมเดล ซึ่งจะเป็นส่วนในการคำนวณค่า dBZ จากค่าการสะท้อนที่อยู่ข้างเคียงโดยการใช้ dot product กับอาเรย์ที่เก็บค่าระยะทางและน้ำหนักจากจุด centroid ของ bin ที่ตกในรัศมี ของการประมาณค่าตามระยะทางแบบ IDW และจะนำไปเก็บในกริดผลลัพธ์ที่ได้สร้างไว้ก่อนหน้า ทำจนกระทั่งเสร็จสิ้นทุกกริดผลลัพธ์แล้วรีเซอร์ฟผลลัพธ์กริดค่าการสะท้อนที่ได้จากการประมาณค่าเชิงพื้นที่ด้วยวิธีการ IDW นี้ กริดผลลัพธ์นี้จะถูกเก็บในรูปแบบเวกเตอร์ราย 15 นาที ต่อไปจะใช้กริดผลลัพธ์นี้เพื่อที่จะแปลงเป็นค่าฝนประมาณค่าตามความสัมพันธ์ของ Z-R ทั้งสามที่ได้เลือกมา

```
#-----#
def convertdbZ2RainRadar(a,b,pLayer):
    #แปลงค่าการสะท้อนของ dBZ เป็นฝนตกต่อชั่วโมง Z-R
    print("ทำการเปลี่ยนค่าที่ไม่ใช่ในฝนตกต่อชั่วโมงเป็น dBZ.....")
    pLayer.startEditing() #ต้องการให้เข้ากับไฟล์ข้อมูล

    #เลือกเฉพาะ dBZที่ไม่เท่า -999
    pLayer.selectByExpression('"Val" > {}'.format(-999))

    #-----#
    for ft in pLayer.selectedFeatures():
        for ft in pLayer.getFeatures():
            if ft['Val'] == -999:
                r=0.0
            else:
                z=10**float(ft['Val'])/10
                r=(z/a)**(1/b)

            pLayer.changeAttributeValue(ft.id(),5,r) #mm/hr

    pLayer.commitChanges() #ต้องการให้หยุดการแก้ไขข้อมูล
    print("ทำการเปลี่ยนค่าที่ไม่ใช่ในฝนตกต่อชั่วโมงเป็น dBZ.....")
    return pLayer
#-----#
```

ภาพที่ 4.16 แปลงข้อมูลผลลัพธ์การประมาณค่าการสะท้อนในหน่วย dBZ ไปเป็นฝนประมาณค่าด้วยความสัมพันธ์ Z-R

ภาพด้านบนจะทำการแปลงข้อมูลค่าการสะท้อนที่ได้จากการประมาณค่าเชิงพื้นที่ด้วย IDW ที่เก็บไว้ในแต่ละกริดของเวกเตอร์ เพื่อนำไปแปลงเป็นข้อมูลฝนประมาณค่าใช้ความสัมพันธ์จากค่า Z-R ทั้งสามที่ได้เลือกไว้ โดยค่าฝนประมาณค่าจะถูกอัพเดท ด้วยฟังก์ชัน changeAttributeValue ในชั้นข้อมูลกริดผลลัพธ์เดิมโดยใช้คอลัมน์ใหม่ที่ได้ออกแบบไว้ก่อนหน้า ซึ่งหน่วยที่ได้คือ mm/hr ต่อไปจะทำการแปลงผลลัพธ์จากเวกเตอร์ให้เป็นรัฐเตอร์เพื่อนำไปใช้ต่อในการประมาณผลแบบฝนสะสม ข้อดีของรัฐเตอร์คือจะทำให้การประมาณผลเร็วเมื่อจำนวนข้อมูลเพิ่มขึ้นตามทั่วเวลาที่ต้องการวิเคราะห์แต่ก่อนที่จะทำการแปลงเป็นข้อมูลแบบรัฐเตอร์ อัลกอริทึมจะแปลงระบบเวลาของไฟล์ข้อมูลจาก UTC ให้เป็น local time ก่อนเนื่องจากต้องนำข้อมูลผลลัพธ์นี้ไปหาความสัมพันธ์ด้วยการตรวจสอบ

กับฝนที่วัดได้จากภาคพื้นดินที่เก็บจากกล้องวัดฝนอัตโนมัติ หรือ rain gauge ที่มีการบันทึกข้อมูลโดยใช้เวลาท้องถิ่นหรือ localtime

```
def utc2local(f):
    #แปลงตัวฟังก์ชัน UTC เป็น local time เพื่อจะนำไปเทียบกับฟังก์ชัน
    print('แปลงตัวฟังก์ชัน UTC เป็น local time...', f)
    from_zone = tz.tzutc()
    to_zone = tz.tzlocal()
    #แปลงเวลาจาก UTC เป็นเวลาท้องถิ่น
    Y=f[0:4]
    m=f[4:6]
    d=f[6:8]
    H=f[8:10]
    M=f[10:12]

    utc_time = dt.strptime(Y+'-'+m+'-'+d+' '+H+':'+M, "%Y-%m-%d %H:%M")
    utc = utc_time.replace(tzinfo=from_zone)
    local_time = utc.astimezone(to_zone)

    #-Local radar time variables, used for save to file
    Y='%02d' % local_time.year
    m='%02d' % local_time.month
    d='%02d' % local_time.day
    H='%02d' % local_time.hour
    M='%02d' % local_time.minute
    fn_local=Y+m+d+H+M
    print(utc_time)
    print(local_time)
    print('แปลงตัวฟังก์ชัน UTC เป็น local time...', fn_local)
    print('*'*50)

    return fn_local
```

ภาพที่ 4.17 พังค์ชั่นในการแปลงเวลาของเรดาร์จาก UTC เป็น localtime

ภาพด้านบนแสดงโค้ดของพังค์ชั่นที่ถูกพัฒนาขึ้นเพื่อใช้ในการแปลงเวลาของเรดาร์จาก UTC เป็น local time โดยเวลาท้องถิ่นของประเทศไทยจะเร็วกว่าเวลามาตรฐานหรือ UTC อยู่ +7 ชั่วโมง ในที่นี้จะใช้โมดูลของ datetime และ dateutil ซึ่งจะมีพังค์ชั่น datetime และ tz ให้เลือกใช้ ชื่อไฟล์ที่ได้จะมีความละเอียดถึงระดับนาที เนื่องจากการตรวจวัดของเรดาร์จะทำการวัดทุกๆ 15 นาทีคือ นาทีที่ 00, 15,30 และ 45 ต่อไปจะทำการส่งออกข้อมูลไปเก็บไว้ในรูปแบบรัสเตอร์

```

def exportRainEstimates2Tif(outp,final_rainRad):
    # ส่งออกผลลัพธ์เป็นไฟล์ raster ในนาม geotif เพื่อไป validate กับสเกตช์
    processing.run("gdal:rasterize",
    {'INPUT':final_rainRad,
     'FIELD':'RainRad',
     'BURN':0,'UNITS':1,
     'WIDTH':2000,'HEIGHT':2000,
     'EXTENT':'522494.1237852003,678494.1237852003,
     1751182.977305875,2151182.977305875 [EPSG:32647]',
     'NODATA':-999,'OPTIONS':'',
     'DATA_TYPE':5,'INIT':None,
     'INVERT':False,
     'EXTRA':'',
     'OUTPUT':outp})

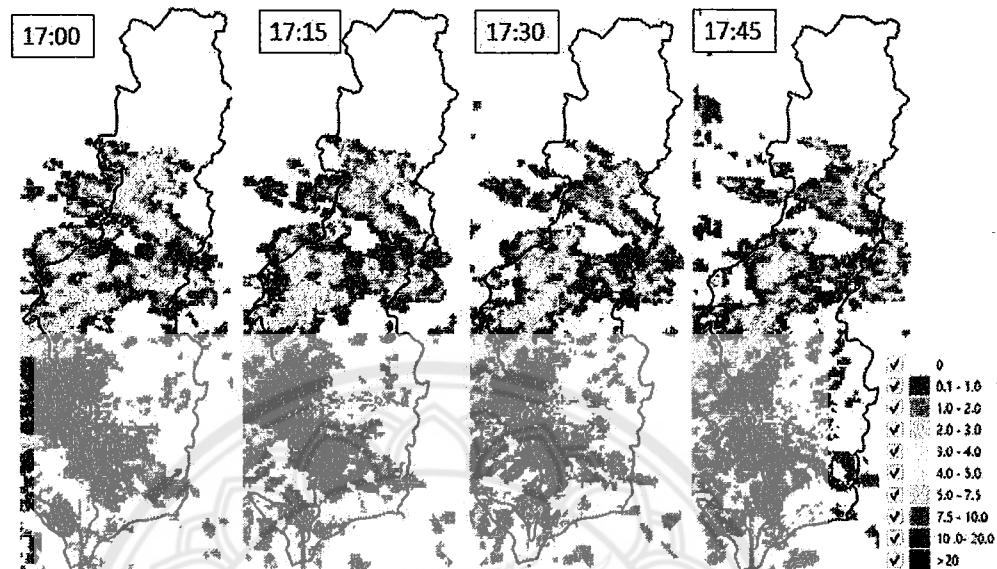
```

ภาพที่ 4.18 พังค์ชั้นในการส่งออกข้อมูลฝนประมาณค่าจากข้อมูลการสะท้อนของเรดาร์ออกไปเป็นชนิดข้อมูลแบบ raster

ภาพด้านบนแสดงพังค์ชั้นในการส่งออกข้อมูลฝนประมาณค่าจากข้อมูลการสะท้อนของเรดาร์ออกไปเป็นชนิดข้อมูลแบบ raster โดยต้องการให้เป็นนามสกุลแบบ geotif เพื่อที่จะสามารถนำไปวิเคราะห์ในซอฟต์แวร์ GIS ทั่วไปได้なくเห็นจากโปรแกรม QGIS โดยใช้เครื่องมือที่มีอยู่แล้วใน Processing Toolbox ซึ่งใช้ความสามารถของไลบรารี gdal โดยการกำหนดความละเอียดของจุดภาพที่ต้องการส่งออกเท่ากับ 2,000 เมตรซึ่งระบบขอบเขตหรือ extent ของ raster ที่ต้องการส่งออกเป็นระบบ UTM โดยอยู่ในโซนที่ 47 ซึ่งมีค่า EPSG เท่ากับ 32647 ค่าจุดภาพที่ไม่ค่าจะกำหนดให้เป็น -999 โดยในการส่งออกนี้จะใช้ข้อมูลในคอลัมน์ที่ชื่อ RainRad หรือฝนประมาณค่าด้วยความแม่นยำพื้นที่แบบ Z-R ซึ่งชื่อของไฟล์จะถูกกำหนดให้เป็น localtime ดังที่ได้จากพังค์ชั้น utc2local

ที่กล่าวมาทั้งหมดถึงตรงนี้นั้น จะเป็นในเรื่องของผลการพัฒนาอัลกอริทึมในการโมเดลข้อมูลเรดาร์ เพื่อให้ได้แผนที่ฝนประมาณค่าที่ครอบคลุมพื้นที่ลุ่มน้ำ โดยได้ดัดแปลงพัฒนานี้นั้นสามารถนำไปปรับใช้เพื่อทำการสร้างแผนที่ฝนประมาณค่าในแต่ละลุ่มน้ำของประเทศไทยได้ เนื่องจากหลักการในการโมเดลนี้ใช้หลักเดียวกัน จะแตกต่างในเรื่องภูมิศาสตร์ของขอบเขตลุ่มน้ำที่ต้องทำการปรับเปลี่ยนตัวโค้ดให้ตรงกับพิกัดที่ต้องการใช้หรือโซนของเส้นโครงแผนที่ที่ต้องการโมเดล นอกจากนี้โปรแกรมที่พัฒนาด้วย PyQGIS นี้ยังสามารถปรับแก้ความละเอียดเชิงพื้นที่ของข้อมูล raster ตามที่ต้องการได้ซึ่งต้องคำนึงถึงความละเอียดตั้งต้นของข้อมูลที่นำมาใช้ด้วย ซึ่งจะเกี่ยวเนื่องกับเรื่อง radar bin resolution ยกตัวอย่างเรดาร์ของเขียงรายมีความละเอียดของ bin gate ที่ 500 เมตร แต่ของพิษณุโลกใช้ที่ 1,000 เมตร นอกจากนี้ beam width หรือความละเอียดของมุมอัซิมูทที่ใช้ในแต่ละ

radar ray ก็ต้องคำนึงถึงจุดนี้ด้วย หากระยะทางห่างจากสถานีมากอาจมีความกว้างของ bin gate เกือบ 5,000 เมตรเมื่อใช้ beam width ที่ประมาณ 1 องศา

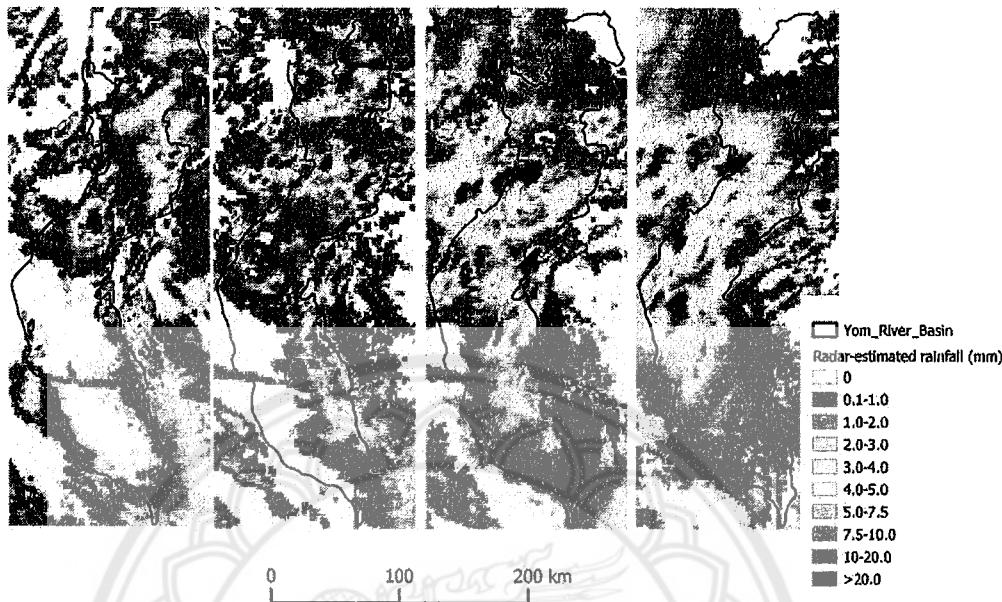


ภาพที่ 4.19 ฝนประมาณค่าจากการโมเสครเดาร์สถานีเขียงรายและพิษณุโลกในพื้นที่ลุ่มน้ำน่านของวันที่ 17 กรกฎาคม พ.ศ.2563

ภาพด้านบนแสดงการเคลื่อนตัวของพายุเขียวตูบุที่เคลื่อนจากทิศตะวันออกไปทางทิศตะวันตกแบบ instantaneous time นั่นคือเป็นผลจากการโมเสคราย 15 นาทีหลังจากของแต่ละข้อมูลเดาร์ โดยเป็นผลลัพธ์โดยใช้ความสัมพันธ์ Z-R แบบ Marshall-Palmer

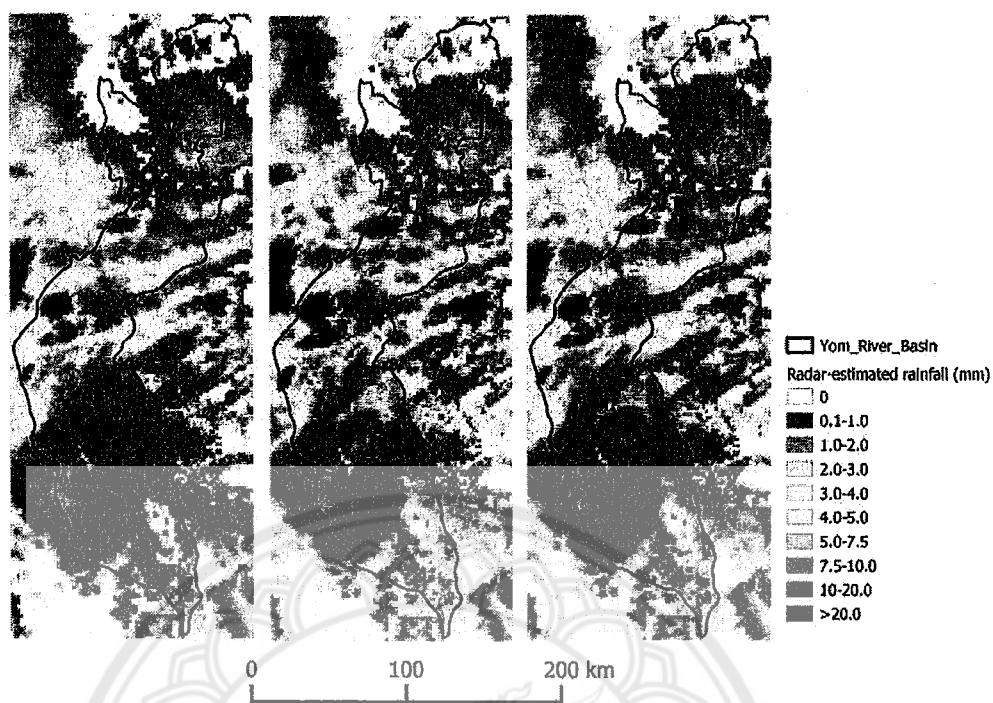
ผู้ใช้งานโค้ดที่พัฒนาจากงานวิจัยในครั้งนี้ อาจต้องการที่จะเพิ่มในส่วนของจำนวนสถานีเดาร์ที่ต้องการใช้ ก็สามารถที่ปรับเปลี่ยนโค้ดในฟังก์ชันโมเสคได้ นอกจากนั้นค่าพารามิเตอร์ทางกายภาพของลำบินหรือค่าการสะท้อนที่นำมาประมาณค่าเขิงพื้นที่ก็สามารถปรับเปลี่ยนให้เหมาะสมแก่ความต้องการได้อีกด้วย

เมื่อได้ข้อมูลภาพโมเสคราย 15 นาทีในรูปแบบข้อมูลแบบราสเทอร์แล้ว ก็นำมาสร้างเป็นฝนสะสมรายชั่วโมง จากนั้นก็สะสมให้เป็นฝนสะสมตามห้วงเวลาที่ต้องการในที่นี้คือ ราย 1, 3, 6, 12, 24 ชั่วโมง ดังผลจากภาพด้านล่าง



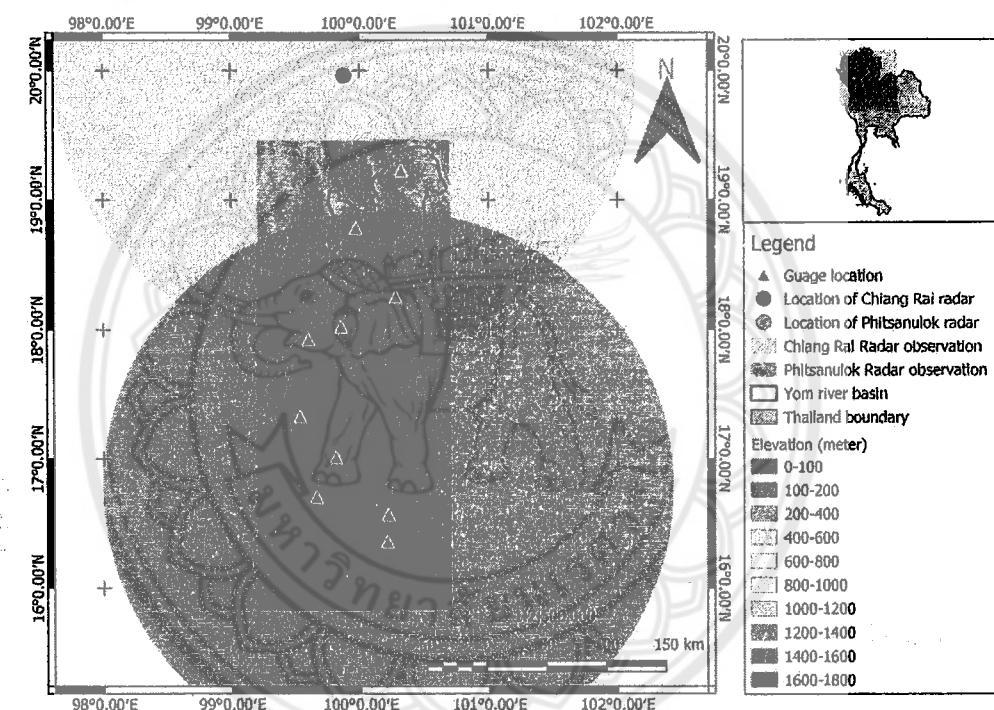
ภาพที่ 4.20 แผนที่ไม่เสียค่าฝันประมาณค่ารายชั่วโมงจากข้อมูลเรดาร์ในพื้นที่ลุ่มน้ำยม ในช่วงเวลา 15.00-18.00 น. ของวันที่ 17 กุมภาพันธ์ พ.ศ.2561 ในช่วงพายุเซนติลู

ภาพด้านบนแสดงผลของการไม่เสียค่าฝันประมาณค่ารายชั่วโมงจากข้อมูลเรดาร์ในพื้นที่ลุ่มน้ำยม ในช่วงเวลา 15.00-18.00 น. ของวันที่ 17 กุมภาพันธ์ พ.ศ. 2561 เพื่อให้เห็นพัฒนาการของการเคลื่อนที่ของกลุ่มฝนที่เกิดจากพายุเซนติลู โดยค่า Z-R ที่ใช้ในภาพด้านบนนี้เป็นของ Marshall-Palmer relationship จะพบว่ากลุ่มฝนนี้ได้เคลื่อนผ่านพื้นที่ของลุ่มน้ำยมในทางตอนบนและตอนกลางของพื้นที่ลุ่มน้ำ ซึ่งกลุ่มฝนประกอบไปด้วยฝนที่ค่อนข้างหนักซึ่งมีกลุ่มฝน convective ที่เป็นลักษณะของฝนฟ้าคะนองที่แสดงเป็นสีแดงในแผนที่ รวมกันกับฝนแบบ stratiform cloud นั่นคือฝนแบบแผ่นที่มักจะประกอบอยู่ในระบบฝนซึ่งให้ลักษณะฝนอ่อน จากผลการไม่เสียค่าฝันนี้จะเห็นประโยชน์ที่สามารถนำไปประยุกต์ใช้ในเรื่องการประมาณค่าการไหลของลำน้ำในพื้นที่ได้ ซึ่งในประเทศไทยยังไม่มีการมุ่งพัฒนาข้อมูลชุดนี้อย่างเป็นระบบ ถ้าหากนำโค้ดที่ได้จากการพัฒนาครั้งนี้ไปประมวลผลแบบใกล้เวลาจริง และนำผลลัพธ์ที่ได้เผยแพร่ให้ประชาชน นักวิจัยและนักวิชาการใช้ ณ ใกล้เวลาจริง จะช่วยในการพยากรณ์น้ำท่วม พยากรณ์ดินถล่มและภัยพิบัติที่เกี่ยวข้องกับพายุในช่วงมรสุมได้ ซึ่งจะช่วยในการตีอนภัยของชาติได้ไม่เพียงแต่ลุ่มน้ำในพื้นที่ศึกษาอีกแล้วนั้น ยังสามารถนำไปใช้ในลุ่มน้ำอื่นๆ ได้โดยไม่ต้องปรับใช้ให้เหมาะสมกับพื้นที่ ต่อไปจะแสดงผลที่จะตอบคำถามถึงความแตกต่างของการประมาณค่าฝนจากค่าการสะท้อนเรดาร์ด้วยค่า Z-R ที่แตกต่างกัน ดังผลที่แสดงในภาพด้านล่าง



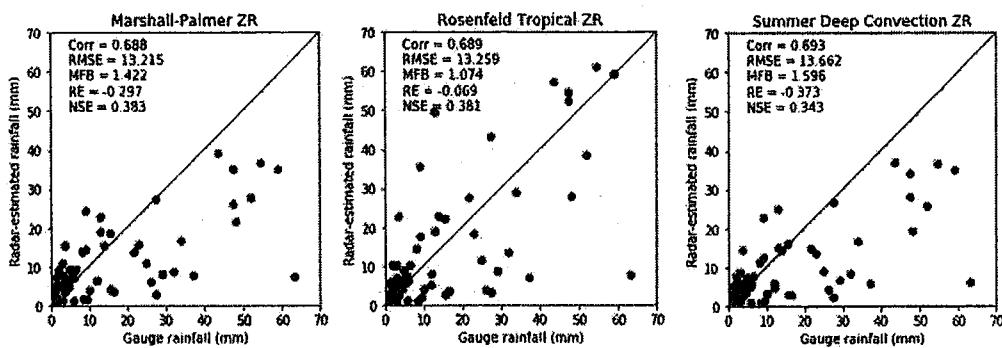
ภาพที่ 4.21 แผนที่ไม่เสียค่าฝันประมาณค่าจากค่าการสะท้อนเรดาร์ตรวจอากาศจากความสัมพันธ์ Z-R ของเวลา 19.00น. วันที่ 17 กรกฎาคม พ.ศ.2561 ในช่วงพายุเชิญติญ (a) Marshall-Palmer (b) Rosenfeld Tropical (c) Summer Deep Tropical Convection
จากภาพด้านบนแสดงให้เห็นถึงความแตกต่างของค่าฝันประมาณค่าจากค่าสะท้อนเรดาร์ตรวจอากาศ ของความสัมพันธ์ Z-R ทั้งสามแบบ จะพบว่า Rosenfeld Tropical จะมีความแตกต่างเชิงพื้นที่ของ การประมาณค่าฝนอยู่เมื่อเทียบกับความสัมพันธ์อีกสองแบบ จะสังเกตได้จากฝนที่ประมาณค่าใน บริเวณตอนกลางจะมีค่าสีแดงที่เป็นบริเวณกว้างกว่าของสองความสัมพันธ์ที่เหลือเนื่องจาก Rosenfeld Tropical นี้ได้จากการเก็บข้อมูลเพื่อหาความสัมพันธ์ของ Z-R ของลักษณะฝนในพื้นที่ เขตร้อน ส่วน Marshall-Palmer Z-R จะพบว่ามีการประมาณค่าฝนที่อ่อนกว่าของ Rosenfeld ซึ่ง คล้ายคลึงกับของ Summer Deep Tropical Convection อย่างไรก็ตามการตรวจสอบการประมาณ ค่าฝนจากความสัมพันธ์ Z-R นี้ต้องเทียบกับปริมาณฝนที่วัดได้จริงจากถังวัดฝนที่ติดตั้งอยู่บนพื้นโลก ซึ่งในทางสากลยอมรับว่าคือปริมาณฝนที่แท้จริง หรือ Ground Truth ที่จะใช้เปรียบเทียบว่าปริมาณ ฝนที่ประมาณค่าได้จากการทางการสำรวจระยะไกลจะใกล้เคียงมากน้อยเพียงใด

4.5 การตรวจสอบผลการประมาณค่าฝนเด ard ด้วยฝนที่วัดได้ที่สถานีวัดฝนภาคพื้นดิน ข้อมูลฝนจากกลังวัดฝนอัตโนมัติในระบบ automatic tipping bucket ได้ถูกนำมาตรวจสอบกับฝนที่ประมาณค่ามาจากการสะท้อนของเรดาร์ตรวจอากาศ โดยได้ทำการกรองสถานีที่มีความสมบูรณ์ทางด้านข้อมูลมากที่สุด รวมทั้งหมด 10 สถานีที่กระจายทั่วทั้งลุ่มน้ำยม เพื่อแสดงให้เห็นถึงความถูกต้องในการประมาณค่าโดยใช้ความสัมพันธ์ Z-R ทั้งสามแบบ เพื่อให้ทราบว่าความสัมพันธ์แบบไหนที่ให้ผลใกล้เคียงกับฝนที่วัดได้ภาคพื้นดินมากที่สุดในช่วงพายุเขียงตูน



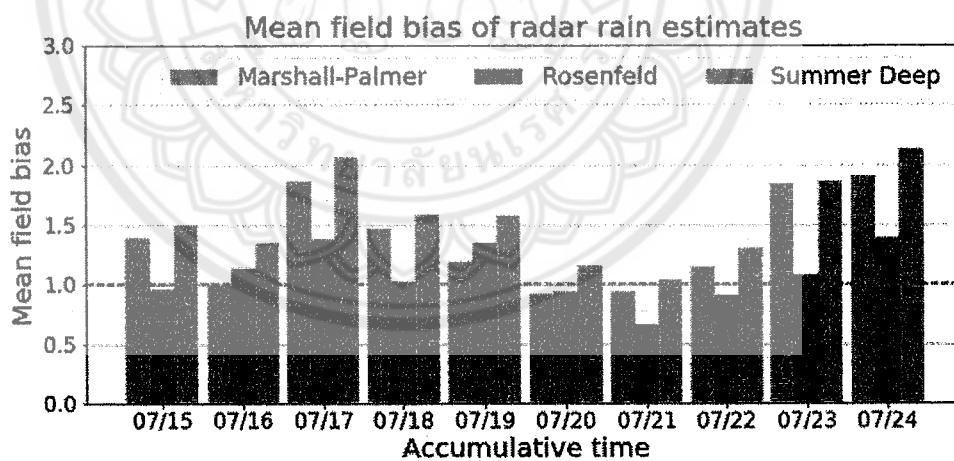
ภาพที่ 4.22 แผนที่แสดงการกระจายตัวของสถานีวัดฝนภาคพื้นดินจำนวน 10 สถานี ที่ใช้ในการตรวจสอบกับค่าฝนประมาณค่าที่ได้จากการโมเดลเรดาร์ตรวจอากาศสถานีเขียงรายและพิษณุโลก

จากการด้านบนจะพบว่าสถานีวัดฝนภาคพื้นดินนั้นมีการกระจายทั่วทั้งพื้นที่ลุ่มน้ำยม ซึ่งถือว่าเหมาะสมต่อการนำมาราดูตรวจสอบฝนประมาณค่าจากการสะท้อนเรดาร์เชิงพื้นที่ โดย 10 สถานีนี้ถือว่าเป็นสถานีที่สมบูรณ์ที่สุดที่มีข้อมูลครบถ้วนช่วงเวลา ไม่มีข้อมูลที่ขาดหายเนื่องมาจากสัญญาณขัดข้องโดย ทำให้มั่นใจได้ว่าผลการตรวจสอบความคลาดเคลื่อนของสองข้อมูลนั้นมีความน่าเชื่อถือ โดยสถานีเหล่านี้จากการออกแบบสนามเพื่อทำการสำรวจจับค่าพิกัดสถานี จะพบว่าเป็นสถานีที่ตั้งอยู่ในศูนย์ราชการนั่นคือ เทศบาลตำบลและองค์กรบริหารส่วนตำบลที่ให้ความอนุเคราะห์กรมอุตุนิยมวิทยาในการใช้สถานที่ติดตั้งสถานี โดยตั้งอยู่ในพื้นที่ราบโกลลุ่มชน



ภาพที่ 4.23 การตรวจสอบความถูกต้องของการประมาณค่าฝนจากค่าสะท้อนเรดาร์ในแต่ละ Z-R relationship ของฝนสะสมราย 24 ชั่วโมงในช่วงพายุเชิงตัญ (a) Marshall-Palmer ZR (b) Rosenfeld Tropical (c) Summer Deep Tropical Convection

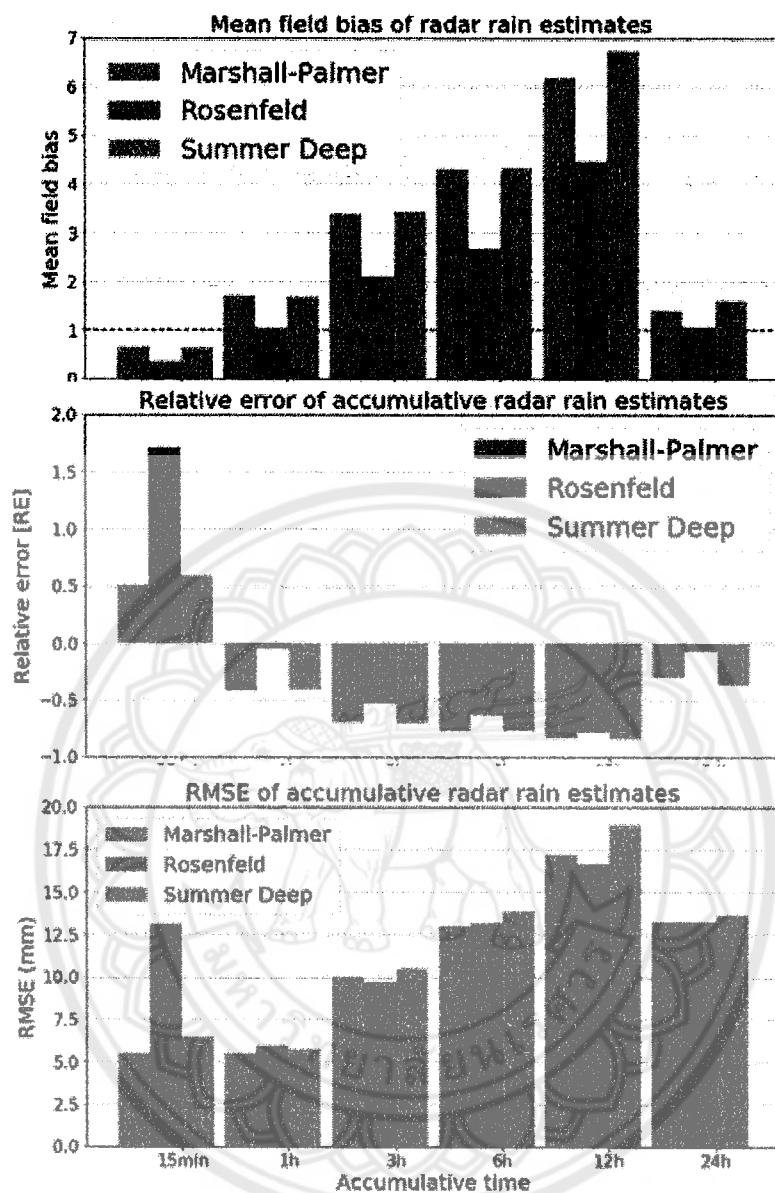
ภาพด้านบนเป็นผลของการตรวจสอบความถูกต้องด้วยฝนสะสมราย 24 ชั่วโมงของความสัมพันธ์ Z-R ทั้งสามแบบ โดยจะพบว่าส่วนใหญ่จะมีค่าที่ใกล้เคียงกัน โดยเฉพาะค่าสัมพันธ์และค่า RMSE อย่างไรก็ตามค่าที่แตกต่างกันที่เห็นได้อย่างชัดเจนคือ ค่า Mean Field Bias กับ Relative Error ของความสัมพันธ์แบบ Rosenfeld Tropical ที่จะให้ค่าที่ใกล้เคียงกับระหว่างค่าฝนจากเรดาร์และค่าฝนที่วัดจริงค่าพื้นดินเมื่อเปรียบเทียบกับความสัมพันธ์ Z-R ที่เหลือ



ภาพที่ 4.24 เปรียบเทียบค่า Mean Field Bias ของความสัมพันธ์ Z-R ทั้งสามในแต่ละวันในช่วงพายุเชิงตัญ

ภาพด้านบนจะพบว่าค่า Mean Field Bias ของความสัมพันธ์ Z-R ทั้งสามมีความผันแปรเปลี่ยนไปในแต่ละวัน ซึ่งแสดงให้เห็นถึงความสามารถของระบบฝนที่มีความแตกต่างกันในแต่ละวัน หากพิจารณาให้ละเอียด

จะพบว่าโดยส่วนใหญ่ค่า Mean Field Bias ของการประมาณค่าแบบ Rosenthal จะมีค่าเข้าใกล้ 1.0 มากกว่าการประมาณค่าด้วยความสัมพันธ์อีกสองแบบ ซึ่งค่าฝนประมาณค่าเรดาร์ของ Marshall-Palmer นั้นให้ผลที่ใกล้เคียงกับฝนภาคพื้นที่ดินกว่าความสัมพันธ์แบบ Summer Deep Tropical Convection การค้นพบตรงนี้จะช่วยให้กรมอุตุนิยมวิทยาได้พิจารณาทางเลือกที่จะ ความสัมพันธ์ของ Z-R ที่เหมาะสมกับระบบฝน ซึ่งกรมอุตุนิยมวิทยาปัจจุบันได้ใช้ความสัมพันธ์ Marshall-Palmer เป็นหลักซึ่งเป็นการประมาณฝนแบบ stratiform cloud และในช่วงมรสุมของ ประเทศไทยนั้นมีอีกหลายที่พัฒนามากจากแบบ convective แม้ว่าจะเป็นช่วงที่พายุกำลังจะ สายตัวเป็นดีเปรสชันหรือเม็กโนท์ที่มีความกดอากาศต่ำกว่าตามที่ ซึ่งหากมีการปรับใช้ ความสัมพันธ์ Z-R อย่างเหมาะสมจะทำให้การประมาณค่าฝนในลุ่มน้ำที่มีพื้นที่เสี่ยงต่อภัยพิบัติที่เป็น ผลกระทบตัวพายุได้ใกล้เคียงมากขึ้น คำถามต่อไปของการตรวจสอบความถูกต้องของการประมาณค่าฝนด้วยค่าการสะท้อนจากเรดาร์ ตรวจสอบภาคภูมิ ความถูกต้องมีความผันแปรอย่างไรในแต่ละช่วงเวลาของฝนสะสม ซึ่งจะได้นำเสนอ ในส่วนผลต่อไป



ภาพที่ 4-25 ผลการตรวจสอบความถูกต้องของการประมาณค่าฝนจากการประมาณค่าเรดารainแต่ช่วงเวลาของฝนสะสมของความสัมพันธ์ Z-R ทั้งสามรูปแบบ

ค่าสถิติที่ใช้ัดจะพบว่า Rosenfeld นั้นให้ค่าสถิติของการประมาณค่าฝนจากเรดาร์ที่ค่อนข้างดีในเกือบทุกช่วงเวลาของฝนสะสมเมื่อเปรียบเทียบกับอีกสองความสัมพันธ์ อย่างไรก็ตามพบว่าค่าสถิติของการประมาณค่าของ Rosenfeld ไม่สามารถประมาณฝนได้ถูกต้องมากนักในฝนราย 15 นาที เมื่อเทียบกับผลการตรวจสอบกับความสัมพันธ์ของ Marshall-Palmer และ Summer Deep Tropical Convection



ภาพที่ 4-26 การตรวจสอบความถูกต้องของฝนสะสมราย 24 ชั่วโมงโดยใช้ค่า mean field bias ของแต่ละสถานีของความสัมพันธ์ Z-R สามแบบ (a) Marshall-Palmer ZR (b) Rosenfeld Tropical (c) Summer Deep Tropical Convection

ภาพด้านบนพบว่าค่า mean field bias ของการประมาณค่าฝนด้วยค่าการสะท้อนของเรดาร์ด้วยอากาศพบว่ามีความแตกต่างกันเชิงพื้นที่ในแต่ละความสัมพันธ์ของ Z-R ทั้งสามแบบ หากค่า mean field bias เข้าใกล้ 1.0 แสดงให้เห็นว่าความแตกต่างระหว่างฝนประมาณค่ากับฝนที่ตรวจวัดได้จริงมีค่าใกล้เคียงกันดังปรากฏเป็นสีเขียวในแผนที่ด้านบน ซึ่งเมื่อพิจารณาความสัมพันธ์ของ Rosenfeld จะพบว่ามีค่า bias ที่น้อยกว่าเมื่อเทียบกับการใช้ความสัมพันธ์ Z-R อีกสองชนิด ที่เหลือ ซึ่งจะมีเพียงแต่ตอนบนของลุ่มน้ำที่มีค่า bias ที่สูงส่วนตอนกลางและตอนล่างนั้นมีค่า bias ที่น้อย ส่วนของ Marshall-Palmer และ Summer Deep Tropical Convection มีค่า bias ที่สูงในทางตอนบนและตอนล่างบางส่วนของลุ่มน้ำท่วม

บทที่ 5 สรุปผลการศึกษาและข้อเสนอแนะ

5.1 สรุปผลการศึกษา

งานวิจัยครั้งนี้มุ่งศึกษาและพัฒนาวิธีการทางภูมิสารสนเทศเพื่อประเมินค่าล้ำน้ำฝนเชิงพื้นที่และเวลาจากค่าการสะท้อนเรเดาร์ตรวจอากาศในช่วงฤดูมรสุม โดยได้ทำการพัฒนาวิธีการในระบบภูมิสารสนเทศใน PyQGIS ซึ่งเป็นภาษาไพธอนที่ใช้กับซอฟต์แวร์หัสเปิด QGIS ซึ่งได้พัฒนากระบวนการสร้างแผนที่โดยมุ่งสร้างโคลัตตันแบบเพื่อให้สามารถปรับใช้กับลุ่มน้ำในประเทศไทยได้ต่อไปในอนาคตเนื่องจากเป็นโคลัตต์ที่ใช้กับซอฟต์แวร์หัสเปิดซึ่งจะประหยัดต้นทุน โดยได้ใช้ลุ่มน้ำยมและลุ่มน้ำน่านในภาคเหนือของประเทศไทยเป็นพื้นที่ศึกษาตัวอย่าง

5.1.1 พัฒนาวิธีการทางภูมิสารสนเทศเพื่อประเมินค่าล้ำน้ำฝนเชิงพื้นที่และเวลาจากค่าการสะท้อนเรเดาร์ตรวจอากาศในช่วงฤดูมรสุม

การพัฒนาวิธีการในการโม่เสกค่าการสะท้อนของเรเดาร์ตรวจอากาศในระบบภูมิสารสนเทศหัสเปิดนั้น ใช้ภาษา PyQGIS ซึ่งเป็นภาษาไพธอนที่ใช้ในบริการในซอฟต์แวร์ QGIS ทำให้เกิดโคลัตตันแบบในการนำไปปรับใช้กับลุ่มน้ำอื่นๆ ของประเทศไทย ประสิทธิภาพของโคลัตตันแบบที่ได้พัฒนามานั้นสามารถประมวลผลข้อมูลได้แบบ batch files นั่นแสดงให้เห็นว่าโคลัตต์ที่พัฒนาได้มีความสามารถในการนำมายกตีใช้เพื่อการประมาณค่าฝนในลุ่มน้ำของประเทศไทยได้ในรูปแบบใกล้เวลาจริง หากมีการนำไปโคลัตต์ที่ได้พัฒนานี้ไปประมวลผลในระบบ server ที่มีประสิทธิภาพสูงจะสามารถนำผลิตภัณฑ์การโม่เสกในประมาณค่าที่ได้นี้ไปเตือนภัยให้กับประชาชนเพื่อลดภัยพิบัติในพื้นที่ที่มีความเสี่ยงสูงได้ ผลของการพัฒนาโคลัตตันแบบนี้สามารถที่จะนำไปวิเคราะห์ฝนสะสมในพื้นที่ลุ่มน้ำได้เพื่อเฝ้าระวังในช่วงการสลายตัวของพายุที่เข้าประเทศไทยในช่วงฤดูมรสุมได้

5.1.2 ศึกษาและวิเคราะห์รูปแบบเจิงพื้นที่ค่าเบี่ยงเบนของฝนประมาณค่าจากค่าการสะท้อนของเรเดาร์ตรวจอากาศที่สัมพันธ์กับค่าล้ำน้ำฝนที่วัดได้จากพื้นที่

ความสัมพันธ์ Z-R ทั้งสามแบบที่นำมาเพื่อหาว่าในช่วงพายุเขิญติลูนน์ ความสัมพันธ์แบบใดที่ให้ค่าฝนประมาณค่าที่เบี่ยงเบนไปจากค่าฝนที่ตรวจวัดได้จริงจากถังวัดฝนอัตโนมัติ พบร่วมกันว่าความสัมพันธ์แบบ Rosenfeld Tropical ZR ให้ค่าเบี่ยงเบนน้อยที่สุดในภาพรวมทั้งในแบบความเบี่ยงเบนเชิงพื้นที่และเชิงทั่วเวลาสะสม

5.2 ข้อเสนอแนะ

1. ควรพัฒนาโดยใช้วิธีการ CAPPI คือใช้หลายมุมยกเพื่อสร้างแผนที่โมเดลโดยมีระดับความสูงคงที่ซึ่งอาจจะใช้การประมาณค่าแบบสามมิติโดยใช้การถ่วงน้ำหนักแบบ Cressmann แทนการใช้แบบ IDW โดยให้คำนึงถึง effective radius ของค่า beam width
2. ควรพัฒนาวิธีการนับจำนวนก้อนเมฆ convective ใน PyQGIS เพื่อเป็นข้อมูลสนับสนุนการตัดสินใจในการประเมินแบบ objective ว่าควรจะใช้ Z-R แบบใดในแต่ละเวลาและเหตุการณ์ ผู้ซึ่งจากผลการวิจัยจะพบว่าการใช้ความสัมพันธ์ ZR แบบเดียวในทุกเหตุการณ์ ผู้นั้นจะไม่สามารถประมาณค่าฝนได้ใกล้เคียงกับฝนที่ตรวจวัดได้จริงในภาคพื้นดิน
3. ควรพัฒนาวิธีการสอบเทียบและปรับแก้ค่าความต่างของสัญญาณค่าการสะท้อนของเรดาร์ในแต่ละสถานีให้อยู่ในระดับที่ใกล้เคียงกันตามมาตรฐาน ซึ่งในการศึกษาครั้งนี้ไม่ได้มุ่งที่จะสอบเทียบสัญญาณเรดาร์ หากในงานวิจัยในอนาคตความมีการสอบเทียบซึ่งวิธีการสอบเทียบที่มีการใช้อย่างแพร่หลาย วิธีการหนึ่งคือการใช้ข้อมูลดาวเทียมเรดาร์ในการปรับเทียบกับสัญญาณเรดาร์ตรวจสอบจากสภาพพื้นดิน
4. ควรพัฒนาวิธีการปรับแก้ค่า bias ของผลิตภัณฑ์ฝนโมเดลในระดับลุ่มน้ำโดยใช้ PyQGIS โดยมุ่งพัฒนาโค้ดต้นแบบเพื่อการใช้งานแบบใกล้เวลาจริง เพื่อที่จะเผยแพร่ผลิตภัณฑ์โมเดลที่ได้สร้างนี้แบบเรียบไทม์ให้กับนักวิจัย นักวิชาการ หน่วยงานเอกชนและประชาชนที่สนใจในการใช้งานโดยทางหน่วยงานราชการผู้รับผิดชอบข้อมูลเรดาร์ควรหาช่องทางในการบริการข้อมูลที่มีค่าชุดนี้ ซึ่งจะทำให้เกิดบริการที่หลากหลายแบบโดยจะทำให้เกิดการกระตุ้นเศรษฐกิจของประเทศขึ้นมาจากการข้อมูลชุดนี้หากให้เอกชนนำไปเพิ่มมูลค่า เช่นการพัฒนาแอพลิเคชันในการเตือนภัย แอพลิเคชันเชิงพื้นที่เพื่อการใช้ชีวิตในเขตเมือง เป็นต้น

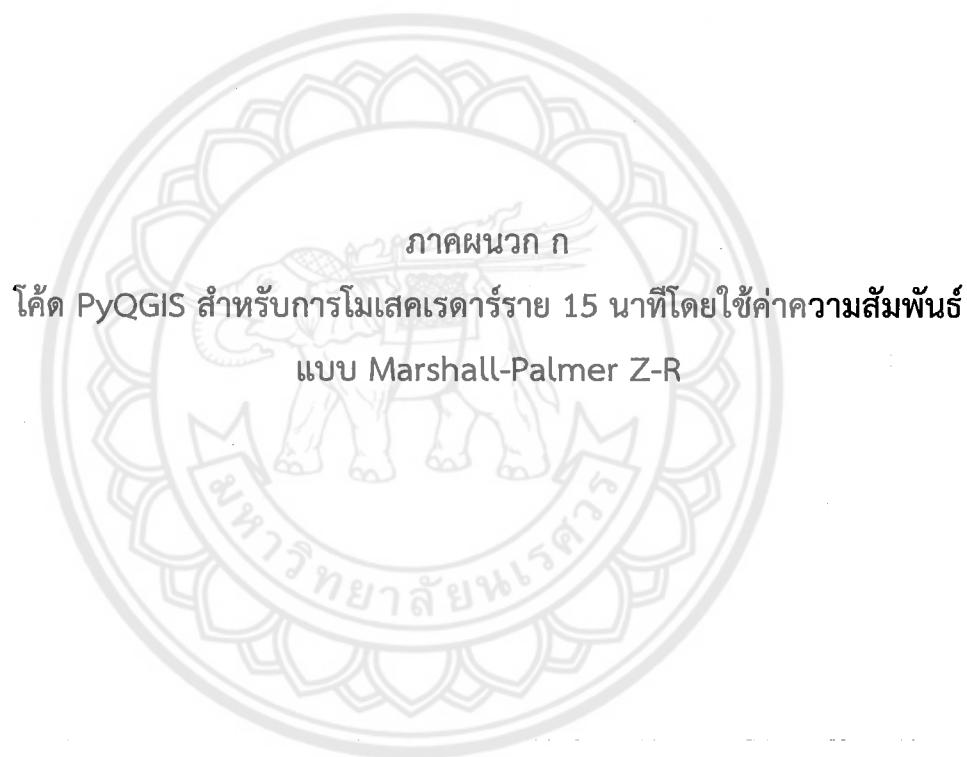
เอกสารอ้างอิง

- คณะกรรมการป้องกันและบรรเทาสาธารณภัยแห่งชาติ (2558). แผนป้องกันและบรรเทาสาธารณภัยแห่งชาติ พ.ศ. 2558. กรุงเทพฯ: กรมป้องกันและบรรเทาสาธารณภัย กระทรวงมหาดไทย.
- ภักดี จันทร์เกษ. 2557. รายงานโครงการประเมินปริมาณน้ำฝนด้วยเรดาร์ภาคตะวันออก. กรมฝนหลวงและการบินเกษตร. กระทรวงเกษตรและสหกรณ์. 171 หน้า
- Battan, L.J., Radar Observation of the Atmosphere, The University of Chicago Press, 1973.
- Chanraket, P., S. Kirtsael, C. Detyothin, A. Nakburee, K. Mongkala, 2015: characteristics of hailstorm over Northern Thailand during summer season. EnvironmentalAsia, 8(1), 101-114.
- Chumchean, S., R. Hanchoowong, W. Bunthai, 2010: Comparison between Rainmaking and Natural Rainfall in the Northeast of Thailand. Proc., 5th conference of Asia Pacific Association of Hydrology and Water Resources, Hanoi, Vietnam.
- Doviak, R.J., and D.S. Zrnic, Doppler Radar and Weather Observations, Academic Press, 1993.
- Djurcillov S. and Pang A., 2000: Visualizing Sparse Gridded Data Sets, IEEE Computer Graphics and Applications, IEEE Computer Society Press, 20(5), 52-57.
- Endo, N., J. Matsumoto, T. Lwin, 2009: Trends in precipitation extremes over Southeast Asia, SOLA, 5, 168–171.
- Fukutomi, Y., and T. Yasunari, 1999: East Asia and western North Pacific during early summer. J. Meteor. Soc. Japan, 77, 753–769.
- Graser, A., 2020, Free and Open Source GIS Ramblings. Retrieved from <https://anitagraser.com/pyqgis-101-introduction-to-qgis-python-programming-for-non-programmers/>
- Heistermann, M., S. Jacobi, T. Pfaff, 2013: Technical Note: An open source library for processing weather radar data (wradlib). Hydrology and Earth System Sciences 17, 863–871.

- Iwashima T, R. Yamamoto, 1993: A statistical analysis of the extreme events: Long-term trend of heavy daily precipitation. *Journal of Meteorological Society of Japan*, 71: 637–640.
- Klongvessa, P., M. Lu, S. Chotpantarat, 2017: Variations of characteristics of consecutive rainfall days over northern Thailand. *Theor Appl Climatol*, DOI 10.1007/s00704-017-2208-4.
- Houze, R. A., Jr., 2012: Orographic effects on precipitating clouds. *Rev. Geophys.*, 50, RG1001, 47pp., doi:10.1029/2011RG000365.
- IPCC. 2014: Climate Change 2014. Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, R.K. Pachauri and L.A. Meyer (eds.)]. IPCC, Geneva, Switzerland, 151 pp.
- Miller, M., 2020. Automating QGIS 3.xx with Python Taking your QGIS skills to the next level. Retrieved from <https://www.udemy.com/course/automating-qgis-3xx-with-python/>
- Mahavik, N., T. Satomura, S. Baimoung, 2013: Radar rainfall analysis in the middle of Indochina peninsula. *J. Disaster Res*, 8, 187-188.
- Mahavik, N. 2015: Study on rainfall over the middle of the Indo-China Peninsula during summer monsoon by producing gauge-calibrated ground-based radar data. Doctoral thesis. Kyoto University. 130pp.
- Mahavik, N., T. Satomura, S. Shige, B. Sysouphanthavong, S. Phonevilay, M. Wakabayashi, and S. Baimoung, 2014: Rainfall pattern over the middle of Indochina Peninsula during 2009–2010 summer monsoon. *HRL*, 8, 57–63.
- Mahavik, N., 2017: Bias Adjustments of Radar Rainfall during Seasonal March of the Summer Monsoon in the Middle of Thailand. *International Journal of Applied Environmental Sciences*, 12(4), 643-660.

- Mahavik, N., 2017: Extreme precipitation trends over the middle of Indochina Peninsula during the period from 1978-2007. International Journal of Earth Sciences and Engineering, 10(3): 595-603.
- Mahavik, N., S. Shige, K. M. Yamamoto, T. Hayashi, 2017: Diurnal propagating rain systems over the middle of Indochina Peninsula during summer monsoon observed by a composite map of gauge-calibrated ground-based radar data. Progress in Earth and Planetary Science. (submitted)
- Matyas, C. J., 2014: A Geospatial Analysis of Convective Rainfall Regions With in Tropical Cyclones After Landfall. International Journal of Applied Geospatial Research, 1(2), 69-89.
- NOAA, 2020. NOAA's Weather and Climate Toolkit(WCT). Retrieved from <https://www.ncdc.noaa.gov/wct/>
- Ogden, F. L., J. Garbrecht, P. A. DeBarry, L. E. Johnson, 2001. GIS and distributed watershed models II:modules, interfaces, and models. J. Hydrol. Eng., 6(6): 515-523.
- Okumura, K., T. Satomura, T. Oki, and W. Khantiyanan, 2003: Diurnal variation of precipitation by moving mesoscale systems: radar observations in northern Thai-land. Geophys. Res. Lett., 30.
- OpenSourceOptions, 2020, Programming for GIS and data science. Retrieved from <https://opensourceoptions.com/blog/category/python/pyqgis/>
- QGIS, 2020. Download QGIS for your platform. Retrieved from <https://qgis.org/en/site/forusers/download.html>
- QGIS project , 2020. PyQGIS Developer Cookbook. Retried from https://docs.qgis.org/3.10/en/docs/pyqgis_developer_cookbook/index.html
- Rinehart, R.E., Radar for Meteorologists, Rinehart Publishing, 1997.
- Satomura, T., 2000: Diurnal variation of precipitation over the Indo-China Peninsula: two-dimensional numerical simulation. J. Meteor. Soc. Japan., 78, 461–475.

- Satomura, T., M. Katsumata, S. Mori, S. Yokoi, J. Matsumoto, S. Ogino, and Y. Kamimura, 2013: To understand typhoons' behavior over Indochina. *JDR*, 8, 153–154.
- Takahashi, H. G., and T. Yasunari, 2008: Decreasing trend in rainfall over Indochina during the late summer monsoon: Impact of tropical cyclones. *J. Meteor. Soc. Japan*, 86, 429–438, doi:10.2151/jmsj.86.429.
- Tantanee, S. and Prakarnrat, S., 2008: Using GIS Buffer Technique to Improve Rainfall-Radar Reflectivity Relationship Estimation. *WSEAS TRANSACTIONS on FLUID MECHANICS*, 4(3), 304-313.
- UCAR Community Programs, 2020. Integrated Data Viewer (IDV). Retrieved from <https://www.unidata.ucar.edu/software/idv/>.
- Xie H., X. Zhou, E. R. Vivoni, J. M.H. Hendrickx, E. E. Small. 2005. GIS-based NEXRAD Stage III precipitation database: automated approaches for data processing and visualization,. *Computers and Geosciences*. 31: 65-76.
- Yokoi, S., and T. Satomura, 2008: Geographical distribution of variance of intraseasonal variations in western Indochina as revealed from radar reflectivity data. *J. Climate*, 21, 5154–5161.



ชื่อไฟล์ "9Mosaic15min_raster_MarshallPalmer_Yom.py"

20200615 ไฟล์นี้เป็นไฟล์โมเดลสำหรับแปลงไฟล์ที่ได้รับจากศูนย์จัดการข้อมูลภัยพิบัติ "095"

โมเดลสำหรับแปลงไฟล์ที่ได้รับจากศูนย์จัดการข้อมูลภัยพิบัติ "090"

จากไฟล์ 094 ที่ไม่เสียเวลาสร้างไฟล์นี้โดยใช้เวลาเพียง 1 นาที แต่ไฟล์นี้จะต้องใช้เวลาประมาณ 1 ชั่วโมง แต่ผลลัพธ์ที่ได้รับจะดีกว่ามาก
โดยคิดว่าถ้าหากจะทำเองต้องเสียเวลาอย่างน้อย 1 ชั่วโมง แต่ไฟล์นี้จะลดเวลาลงเหลือเพียง 1 นาที จึงแนะนำให้ใช้ไฟล์นี้

dbz7-55 ตั้งแต่กว่า 5 กม.

เวลาที่ได้รับมาจากศูนย์จัดการข้อมูลภัยพิบัติ "095"

zr ที่ใช้คือ z=200R1.6 marshallPalmer

```
import numpy as np
from scipy.stats import linregress
import os
import datetime
import fnmatch
from datetime import datetime as dt
from dateutil import tz
#-----
def copyRad(pLayer,phLayer,pr_phs):
    print('คัดลอกเรตาร์.....')
    i=0
    for f in pLayer.getFeatures():
        poly = QgsFeature()
        #สร้างชีนรูปปีกเรตาร์จากชุดข้อมูลด้านบน ดูค่าตัวแปร poly ด้านบน
        poly.setGeometry(f.geometry())
        #update feature
        poly.setAttributes([i,f['value'],f['radialAng'],f['heightASL']])
        phLayer.updateFeature(poly)
    print('คัดลอกเรตาร์.....')
    print('copyRad Done')
```

```

pr_phs.addFeatures([poly]) #อัพเดตค่ารูปปีชื่นที่กำลังทำงานผ่านdataProvider หรือ pr
phLayer.updateExtents() #อัพเดต map
i+=1

return phLayer
#-----#
def radarFlareDetectionRound1(phLayer,PC_BinWithVal,Corr,dist):
    #ทดสอบได้พังค์ชันนี้ไม่ได้ใช้ spatial index นะ
    print('ค้นหา radar flare ตามมุม รอบที่ 1.....')
    #หาค่ามุมทั้งหมดของเวدار์
    idx = phLayer.fields().indexOf('radialAng')
    values = phLayer.uniqueValues(idx)

    #สร้าง spatial index ให้ไว้เจอเรเพิ่มความเร็วในการประมวลผล
    all_features = {}
    index = QgsSpatialIndex() # Spatial index
    for ft in phLayer.getFeatures():
        index.insertFeature(ft)
        all_features[ft.id()] = ft

    # phLayer.startEditing()
    i=0

    #ลูปทีละมุมเพื่อกำหนนค่าสถิติ
    flare_ang=[]#ลิสเพื่อก็บนมุมที่เป็น flare
    for ang in values:
        # if i<10:print('angle:',ang)
        #อ่านเพิ่ม selectByExpression https://gis.stackexchange.com/questions/340961/pyqgis-using-variable-instead-of-number-in-selectbyexpression-query
        #อ่านเพิ่ม https://docs.qgis.org/testing/pdf/en/QGIS-testing-PyQGISDeveloperCookbook-en.pdf
        # req = QgsFeatureRequest().setFilterExpression(' "radialAng" == "ang" ') #ไม่ให้แบบนี้

        phLayer.selectByExpression("radialAng" = "{}".format(ang))
        inds=phLayer.selectedFeatureIds()

        #ลูปเพื่อคำนวนค่าสถิติของแต่ละมุม ด้วยการใช้ spatialIndex ที่ทำไว้ก่อนหน้านี้

```

```

val_list=[]

#เก็บค่าการสะท้อนของมุมที่ถูกเลือกด้วยการลูปทุกฟีเจอร์ในมุมนั้นๆด้วย spatial index
for id in inds:
    val_list.append((id,all_features[id]['value']))

dv=np.array(val_list) #แปลงเป็นอาเรย์
slope, intercept, r_value, p_value, std_err = linregress(dv[:,0], dv[:,1])
ct=len(inds)

#อัพเดทคอลัมน์ด้วยการฟีเจอร์ของลูปมุมที่เลือกด้วย spatial index
j=0
for k,f in enumerate(phLayer.selectedFeatures()):
    if ((ct/dist)*100.0)>=PC_BinWithVal and r_value>=Corr :
        #ตรวจสอบหากต่างจาก "082" เพราะอัพเดทค่า -999.0 ลงไปเมื่อเจอก flare ในรอบที่ 1 เลย
        attrs={1:-999.0,4:k,5:ct,6:(ct/dist)*100.0,7:str(slope),8:str(r_value),9:1}
        phLayer.dataProvider().changeAttributeValues({f.id():attrs})#1=flare
        if j==0:flare_ang.append(ang) #เก็บค่ามุมที่เป็น flare สำหรับฟีเจอร์แรกที่พบ

    else:
        attrs={4:k,5:ct,6:(ct/dist)*100.0,7:str(slope),8:str(r_value),9:0}
        phLayer.dataProvider().changeAttributeValues({f.id():attrs})#0=ok

    j+=1

# phLayer.commitChanges()
return phLayer,flare_ang,values
#-----
#refine เพื่อหา radar flare ขนาดใหญ่ที่ยังตกค้างจากการหารอบแรก
#ในรอบนี้จะใช้ "PC_BinWithVal">>=0.80 และ"Corr">>=0.5 และต้องเป็นมุมที่อยู่ซึ่ง Flare=1 ที่หาในรอบแรก
#ไปแล้ว
#ค่า "Corr">>=0.5 นี้เป็นตามสมมติฐานที่ว่าความสัมพันธ์แบบ linear ไม่เหมาะสมกับ flare ทำให้ค่า corr ต่ำ
#แม้ว่าค่า "Corr">>=0.5 จะต่ำ แต่ก็ยังแสดงให้เห็นว่ามีความสัมพันธ์ การใช้ nonlinear จะทำให้พบ
#ความสัมพันธ์

```

```

#ที่อาจจะให้ค่า R2 ที่สูงก็ตาม แต่จะทำให้การคำนวนใช้เวลาเพิ่มไปอีก จะทำให้ไม่เหมาะสมกับการใช้งานจริงของ
TMD

#นิสิตอาจารย์ใช้ nonlinear เก็บ radar flare ในรอบ refine นี้ก็ได้ ลองใช้  $y=x^2$  ก่อน
def radarFlareDetectionRound2(phLayer,PC_BinWithVal,Corr,flare_ang,values):
    print('ด้านหน้า radar flare ตามมุม รอบที่ 2.....')
    phLayer.startEditing()

#แปลงมุมทั้งหมดเป็นอาร์เรย์
ang_arr=np.array(values)

#สร้างลิสตามุมที่ทางเดียวของ Flare ในรอบแรก
flare_list=[]
for ang in flare_ang:
    w_check=ang in ang_arr
    if w_check==False: continue #ตรวจมุมที่กำลังจะป็นตัวไม่ใช่ flare ให้วนไป
    id=np.where(ang_arr==ang) #หาอินเด็กมุม flare ของมุมทั้งหมดที่มี
    id=id[0]
    #กรณีที่ flare ไม่ได้เป็นมุมแรกและมุมสุดท้าย
    if id>0 and id<(len(values)-1):
        pre_id=id-1 #อินเด็กซ์ก่อน flare
        pot_id=id+1 #อินเด็กหลัง flare
        flare_list.append(ang_arr[pre_id])
        flare_list.append(ang_arr[pot_id])

    #กรณีที่มุมของ flare เป็นมุมแรกของ sweep
    if id==0: flare_list.append(ang_arr[id+1]) #ให้อินเด็กซ์เป็น +1

    #กรณีที่มุมของ flare เป็นมุมสุดท้ายของ sweep ให้อินเด็กซ์เป็น -1
    if id==len(values)-1: flare_list.append(ang_arr[id-1])

#flare_list คือ ลิสตอมุมที่คาดว่าจะเป็น flare ที่ตัดจางจากการใช้ linear regression ในรอบแรก
#โดยเป็น range ที่อยู่ข้างๆกับ flare ในรอบแรก ซึ่งจะไปกรองออกสองค่าดังลูกปัดด้านล่าง

#ลูปเพื่อขับเดิน flare รอบ 2
for ang in set(flares_list): #ใช้ set เพื่อจะหาความที่เป็น unique จริงๆ

```

```
#https://gis.stackexchange.com/questions/315601/getting-particular-feature-using-expression-in-pyqgis
#https://docs.qgis.org/testing/pdf/en/QGIS-testing-PyQGISDeveloperCookbook-en.pdf
#ต้องส่วนนี้ที่ปรับปูงเพิ่มจากไฟล์ "082" เพื่อเก็บตก flare จากรอบเขต
phLayer.selectByExpression("radialAng" = {} AND "Flare1" != {} AND "PC_BinWithVal">>= {} AND
"Corr">>={}'.format(ang,1,PC_BinWithVal,Corr))

for f in phLayer.selectedFeatures():
    phLayer.changeAttributeValue(f.id(),10,2)#2=flare ที่ตอกค้างมาจากการรอบเขต "flare2"
    phLayer.changeAttributeValue(f.id(),9,2)#2=flare ที่ตอกค้างมาจากการรอบเขต "flare1"
    phLayer.changeAttributeValue(f.id(),1,-999.00) #เปลี่ยนแนวคิดใหม่ ให้แทนที่ค่า dbZ คือลักษณะ
"value" เลย และว่าจะใช้แทน valueFlare

    phLayer.commitChanges()
    return phLayer
#-----
def interpolateNearFlarebyIDW(phLayer,geom):
    print("ประมาณค่า dbz รอบข้าง flare.....")
    #เก็บบุบทั้งหมดใส่ลิสต์
    idx = phLayer.fields().indexOf('radialAng')
    values = phLayer.uniqueValues(idx)
    ang_list=list(values)
    ang_list.sort() #จัดเรียงมุนงนในลิสเพื่อจะนำไปหาค่าเด็กๆ

    #เก็บบุบที่พบ flare ทั้งหมดในลิส
    flare_ang_list=[]
    phLayer.selectByExpression("value" = '{}'.format(-999.0))
    for k,f in enumerate(phLayer.selectedFeatures()):
        flare_ang_list.append((f['radialAng']))
    flare_list=set(flare_ang_list)

    #อุปเพื่อหาค่าด้วย idw
    #นำบุบข้างๆที่พบ flare ไปเป็นค่า idw
    print('ประมาณค่า dbz ข้างๆ มุมที่เป็น radar flare ...')
    phLayer.startEditing()
```

```

for ang in flare_list:
    id=ang_list.index(ang) #หาค่ามุมจากค่าอินเด็กซ์ของๆตามกรอบด้านล่าง
    if id>0 and id<len(ang_list):
        pre_id=id-1
        pst_id=id+1

        if id==0:
            pre_id=0
            pst_id=id+1

        if id==len(ang_list):
            pre_id=id-1
            pst_id=id

    val_list=[ang_list[pre_id],ang_list[pst_id]] #มุมข้างๆ flare ในแต่ละครั้งของลูป

phLayer.selectByExpression("radialAng" = {} .format(ang))#ang คือ มุมที่พบ flare
#ลูปเช็คมาว่ามุมที่เป็น flare เพื่อหา centroid
for k,f in enumerate(phLayer.selectedFeatures()):
    #สร้างลิสต์เปล่าเพื่อบรรจุระยะทางและค่า dbz
    dist_val=[]
    #สร้างลิสต์ถ่วงค่าน้ำหนักตามระยะทางที่ห่างออกไป
    w_list=[]

    #หา centroid ของ f
    pt_f=f.geometry().centroid().asPoint()
    #หาระยะทางของ f กับจุดสถานีเรดาร์
    distance = QgsDistanceArea()
    d = distance.measureLine(pt_f, geom.asPoint()) #ระยะทางของจุด f กับ จุดสถานีเรดาร์ (m)

    #หา effective radius ตามระยะทางโดยเรดาร์ปัจจุบัน assume ที่ 1.0 องศา
    #-http://www.1728.org/angsiz.htm

```

```

re=np.tan(np.radians(1.0/2))*2*(d/1000) #หน่วย กม.

#สร้างรูปปีดจาก effective radius เพื่อนำไป intersect กับค่าศก.เรดาเรซที่ตกลงในเงื่อนไข
poly=QgsFeature()
rem=re*1000.0 #effective radius หน่วยเมตร

ll=pt_f[0]-rem,pt_f[1]-rem
ul=pt_f[0]-rem,pt_f[1]+rem
ur=pt_f[0]+rem,pt_f[1]+rem
lr=pt_f[0]-rem,pt_f[1]-rem

points = [QgsPointXY(ll[0],ll[1]),QgsPointXY(ul[0],ul[1]),\
          QgsPointXY(ur[0],ur[1]),QgsPointXY(lr[0],lr[1])]

poly.setGeometry(QgsGeometry.fromPolygonXY([points]))

#ลูปมุมซ้ายๆทั้งสองเพื่อนำมาตรวจสอบว่าเงื่อนไขแล้วถ้าตรง จะเก็บในลิสเพื่อทำ idw
#ตรงนี้ต้องใช้ QgsSpatialIndex จะทำให้การ intersects เร็วขึ้นมาก
for ba in val_list: #ตรงนี้จะซ้ำมาก เพราะต้องลูปทุกไฟล์ขอริวในทั้งสองบุน
    phLayer.selectByExpression("radialAng = {}".format(ba)) #ใช้ช้ากับลูปบวนได้หรือไม่
    for kk,fa in enumerate(phLayer.selectedFeatures()):
        if fa.geometry().intersects(poly.geometry()): #spatial intersect ภารกรของครั้งแรก
            pt_fa=fa.geometry().centroid().asPoint()

            #หาระยะทางของ fa รอบๆ กับจุดศก.เรดาเรซที่มี flare กรองครั้งที่สอง
            dist = QgsDistanceArea()
            dt = distance.measureLine(pt_f, pt_fa) #ระยะทางของจุด f กับ fa (m)
            if dt<rem and fa['value']>-999:

                #เพิ่มค่าระยะทางที่คำนวนได้ กับค่าการสะท้อน(value)ของ fa เข้าไปในลิส
                dist_val.append((dt,fa['value']))

#ถ้าระยะทางมากกว่า 0 แต่ยังอยู่ในรัศมีให้ถ่วงน้ำหนักตามระยะทางแล้วนำไปเก็บใน list
ของค่าถ่วงน้ำหนัก
if dt>0:

```

```

w=1/dt
w_list.append(w)

else:
    w_list.append(0)

if len(dist_val)==0:continue #ถ้าไม่มีจุดใดในรัศมี effective radius ของ pt_f เลย ให้วัน

w_check=0 in w_list
dv=np.array(dist_val) #แปลงเป็นอาเรย์
if w_check==True:
    idx=w_list.index(0) #หาตำแหน่งในลิสต์ว่าตรงไหนที่มีค่าน้ำหนักเป็น 0
    z_idw=dv[idx,1] ##กำหนดให้ค่าที่ตรงกับพิกัดของกริดเล็กน้อยเลย โดยไม่สนใจว่าจะมีค่าอื่นที่อยู่
    ในรัศมีนั้นหรือไม่
    phLayer.changeAttributeValue(f.id(),1,float(z_idw)) #1 หมายความว่าแทนแทนค่า dbz ใน
    value เลย แตกต่างกับ 082
else:
    wt=np.transpose(w_list)
    z_idw=np.dot(dv[:,1],wt)/sum(w_list) #ใช้ dot product เพื่อคำนวนหาค่าdbzด้วยวิธีการ idw จะ
    ได้ค่าเดียวกัน
    # อ่านเพิ่ม https://www.guru99.com/numpy-dot-product.html
    phLayer.changeAttributeValue(f.id(),1,float(z_idw)) #1 หมายความว่าแทนแทนค่า dbz ใน
    value เลย แตกต่างกับ 082

phLayer.commitChanges()

#-----
def ExtractBeamBlockByCentroid(phLayer,rLayer):
    print('สร้าง BeamBlock....')
    phLayer.startEditing()
    for f in phLayer.getFeatures():
        #สร้างจุดจาก centroid ของรูปปิด
        pts = f.geometry().centroid().asPoint() #get ค่าพิกัด centroid ของกริดแล้วสร้างให้เป็นจุด
        #นำค่ามาสร้าง BeamBlock

```

val, res = rLayer.dataProvider().sample(QgsPointXY(pts[0],pts[1]), 1) #val คือค่าbeamblockที่
เก็บกลับมาจากการสกัดด้วยจุด sample

```
#อัพเดทเฉพาะค่า beamblock ในไฟล์ pLayer
if np.isnan(val):
    phLayer.changeAttributeValue(f.id(),11,0)
else:
    phLayer.changeAttributeValue(f.id(),11,val)
```

```
phLayer.commitChanges()
print('เสร็จสิ้นการสกัดค่า BeamBlock....')
return phLayer

#-----
#อัพเดตคอลัมน์ RQI โดย 0=bin is clear,1=Flare,2=Beamblock,3=Flare+Beamblock
def updateRQI(phLayer):
    print('กำลังอัพเดตคอลัมน์ RQI...')
    phLayer.startEditing()
    #for az in values:
    phLayer.selectByExpression("Flare1" = {} AND "BeamBlock" = {}'.format(0,0))
    for f in phLayer.selectedFeatures():
        phLayer.changeAttributeValue(f.id(),12,0)

    phLayer.selectByExpression("Flare1" = {} AND "BeamBlock" = {}'.format(0,1))
    for f in phLayer.selectedFeatures():
        phLayer.changeAttributeValue(f.id(),12,2)

    phLayer.selectByExpression("Flare1" >= {} AND "BeamBlock" = {}'.format(1,0))
    for f in phLayer.selectedFeatures():
        phLayer.changeAttributeValue(f.id(),12,1)

    phLayer.selectByExpression("Flare1" >= {} AND "BeamBlock" = {}'.format(1,1))
    for f in phLayer.selectedFeatures():
        phLayer.changeAttributeValue(f.id(),12,3)
```

```

phLayer.commitChanges()
print('เสร็จสิ้นการอัปเดตคอลัมน์ RQI...')

return phLayer

#-----
#-----#
#ฟังก์ชันด้านล่างเป็นของ程式โมเดล
def calParaGrid(xmin,xmax,ymin,ymax,ds_s):
    print('คำนวนพารามิเตอร์เพื่อการโมเดล.....')
    #-คำนวนจำนวนกริดที่ต้องใช้ในแนวแก้วและหลัก
    numCols=int((xmax-(xmin-(xmin//ds_s)))/ds_s)+1 #จำนวนกริดในแนวหลัก // คือการหารเอาเศษ, +1
    คือต้องการให้ค่าลูมิเนซซ์ตัวเลข
    #  print('จำนวนกริดในแนวหลัก:',numCols)
    numRows=int((ymax-(ymin-(ymin//ds_s)))/ds_s)+1 #จำนวนกริดในแนวแก้ว
    #  print('จำนวนกริดในแนวแก้ว:',numRows)

    #-กำหนดค่าพิกัดมุมช้าย่าง
    llx=xmin-(xmin//ds_s)
    lly=ymin-(ymin//ds_s)

    #-set dimension first polygon at lower left
    ll,ul,ur,lr=[[llx,lly],[llx,lly+(ds_s*2)],\
                  [llx+(ds_s*2),lly+(ds_s*2)],[llx+(ds_s*2),ds_s]

    return numCols,numRows,llx,lly,ll,ul,ur,lr
    print('เสร็จสิ้นคำนวนพารามิเตอร์เพื่อการโมเดล.....')

#-----
def createGrid(numCols,numRows,llx,lly,ll,ul,ur,lr,ds_s,pLayer,poly,pr,gridVal):
    print('ทำการสร้างกริดผลลัพธ์การโมเดล.....')
    i=0
    for r in range(numRows):
        for c in range(numCols):
            #ตั้งค่าพิกัดของแต่ละมุมของรูปปีด

```

```

ll[0],ll[1] = llx+(c*ds_s),lly+(r*ds_s)
lr[0],lr[1] = ll[0]+ds_s, ll[1]
ur[0],ur[1] = lr[0],lr[1]+ds_s
ul[0],ul[1] = ll[0],ur[1]

#สร้าง list ของจุด หรือชุดข้อมูลจุดเพื่อนำไปสร้างเป็นรูปปีกกริดเล็ก
points = [QgsPointXY(ll[0],ll[1]),QgsPointXY(ul[0],ul[1]),\
           QgsPointXY(ur[0],ur[1]),QgsPointXY(lr[0],lr[1])]

#สร้างรูปปีกกริดจากชุดข้อมูลจุดด้านบน ดูค่าตัวแปร poly ด้านบน
poly.setGeometry(QgsGeometry.fromPolygonXY([points]))

#-calculate area and perimeter from geometry
#-คำนวนค่าเรขาคณิตของรูปปีกกริดได้แก่พื้นที่และความยาวเส้นรอบรูป
area=(poly.geometry().area())/1000/1000
peri=poly.geometry().length()

#อัพเดทค่าพิวาร์ของพีกกริดที่กำลังคำนวน
poly.setAttributes([i,area,peri,gridVal])

pr.addFeatures([poly]) #อัพเดทค่ารูปปีกกริดที่กำลังทำงานผ่านdataProvider หรือ pr

pLayer.updateExtents() #อัพเดท map

i+=1

print('เสร็จสิ้นการสร้างกริดผลลัพธ์.....')
return pLayer
#-----#
def clipRadarByProvince(pLayer,prLayer,pcLayer,pr):
    #ปรับปรุงคัดใหม่ส่วนนี้ทำให้เวลาไม่เก็บนาที คัดมาหากไฟล์ "075"
    #pLayer=ข้อมูลเรดาร์ที่จะตัด, prLayer=ขอบเขตจังหวัดแพร่
    #pcLayer=shapefileเพื่อเก็บเรดาร์ที่จะตัดตามขอบเขตแพร่
    #pr=dataProvider ของpcLayer

```

```

#สร้างรูปปีกข้อบอกราชจังหวัดเพื่อต่อไป query กรองเฉพาะรัฐ
poly_p=QgsFeature()
offset=2000
ext = prLayer.extent()
(xmin, xmax, ymin, ymax) = (ext.xMinimum()-offset, ext.xMaximum()+offset,\n
                             ext.yMinimum()-offset, ext.yMaximum()+offset)

#สร้าง list ของจุด หรือจุดข้อมูลๆเดียวสำหรับเป็นรูปปีก
points = [QgsPointXY(xmin,ymin),QgsPointXY(xmin,ymax),\n
           QgsPointXY(xmax,ymax),QgsPointXY(xmax,ymin)]

#สร้างชิ้นรูปปีกข้อบอกราชที่ศึกษาจากจุดข้อมูลๆเดียว
poly_p.setGeometry(QgsGeometry.fromPolygonXY([points]))

#จูบเดคร์เพื่อ intersect ผล spatial index กับ ข้อบอกราช
inBound = poly_p.geometry().boundingBox()
i=0
for ft in pLayer.getFeatures():
    if ft.geometry().intersects(inBound):
        poly_cri = QgsFeature() #เพิ่มตรงนี้
        #สร้างชิ้นรูปปีกจริงจากจุดข้อมูลๆเดียว คู่ค่าตัวแปร poly ด้านบน
        poly_cri.setGeometry(ft.geometry())
        #-update feature
        poly_cri.setAttributes([i,ft['value'],ft['heightASL']])
        pr.addFeatures([poly_cri]) #อัพเดทค่ารูปปีกชิ้นที่กำลังทำงานผ่าน dataProvider หรือ pr
        pcLayer.updateExtents() #อัพเดท map
        i+=1

return pcLayer

#-----
#-คำนวนจุด centroid ของกริด C ที่ตกในพื้นที่กริด A หรือ B เพื่ออัพเดทค่าพิวเดิมสูตร C
def updateFieldOutputByIntersectCentroid(paLayer,pbLayer,pcLayer,rd):
    print('กำลังคำนวนการอัพเดทค่าพิวเดิมกริด C ด้วยวิธี spatial intersection.....')
    i=0

```

#อ่านวิธีการใช้ spatial index เพื่อการเข้าถึงไฟล์ไฟล์

<https://gis.stackexchange.com/revisions/224954/4>

pcLayer.startEditing() #ตั้งค่าให้เริ่มการแก้ไขข้อมูล C

```
all_featuresA = {}
```

```
indexA = QgsSpatialIndex() # Spatial index
```

```
for ft in paLayer.getFeatures():
```

```
    indexA.insertFeature(ft)
```

```
    all_featuresA[ft.id()] = ft
```

```
all_featuresB = {}
```

```
indexB = QgsSpatialIndex() # Spatial index
```

```
for ft in pbLayer.getFeatures():
```

```
    indexB.insertFeature(ft)
```

```
    all_featuresB[ft.id()] = ft
```

#loopเพื่อคำนวนกริดผลลัพธ์ของ idw

```
for feat in pcLayer.getFeatures():
```

```
    inGeom = feat.geometry()
```

```
    idsListA = indexA.intersects(inGeom.boundingBox())
```

```
    idsListB = indexB.intersects(inGeom.boundingBox())
```

```
pt_c=feat.geometry().centroid().asPoint()
```

#สร้างลิสต์ถาวรเก็บระยะทางและค่า dbz

```
dist_val=[]
```

#สร้างลิสต์ถาวรค่าน้ำหนักตามระยะทางที่ห่างออกไป

```
w_list=[]
```

#loop idsListA เพื่อหาจุดที่อยู่ในแนวรัศมีในการทำ idw ค่า dbz

```
for id in idsListA:
```

```
    pt_a = all_featuresA[id].geometry().centroid().asPoint()
```

```
    distance = QgsDistanceArea()
```

```

d = distance.measureLine(pt_c, pt_a) #ระยะทางของจุด C กับ จุดA

if d>rd: continue

if      all_featuresA[id]['value']>=7      and      all_featuresA[id]['value']<=55      and
all_featuresA[id]['heightASL']<5000:
    #-เพิ่มค่าระยะทางที่คำนวนได้ กับค่าการสะท้อน(value)ของ B เข้าไปในลิส
    dist_val.append((d,all_featuresA[id]['value']))

#ถ้าระยะทางมากกว่า 0 แต่ยังอยู่ในรัศมีให้ถ่วงน้ำหนักตามระยะทางแล้วนำไปเก็บใน list ของค่าถ่วง
น้ำหนัก

if d>0:
    w=1/d
    w_list.append(w)
else:
    w_list.append(0)

#loop idsListB เพื่อหาจุดที่อยู่ในแนวรัศมีในการทำ idw ค่า dbz
for id in idsListB:
    pt_b = all_featuresB[id].geometry().centroid().asPoint()

    #-ทำเป็นพังค์ชั่นได้
    distance = QgsDistanceArea()
    d = distance.measureLine(pt_c, pt_b) #ระยะทางของจุด C กับ จุดA

    if d>rd: continue

    if      all_featuresB[id]['value']>=7      and      all_featuresB[id]['value']<=55      and
all_featuresB[id]['heightASL']<5000:
        #-เพิ่มค่าระยะทางที่คำนวนได้ กับค่าการสะท้อน(value)ของ B เข้าไปในลิส
        dist_val.append((d,all_featuresB[id]['value']))

#ถ้าระยะทางมากกว่า 0 แต่ยังอยู่ในรัศมีให้ถ่วงน้ำหนักตามระยะทางแล้วนำไปเก็บใน list ของค่าถ่วง
น้ำหนัก

```

```

if d>0:
    w=1/d
    w_list.append(w)
else:
    w_list.append(0)

if len(w_list)==0:continue #ถ้าไม่มีจุดใดในกริด C เลยให้กลับไปวนใหม่

#อัพเดทค่าที่ได้จากการประมาณค่าในคลัมบ์
w_check=0 in w_list
dv=np.array(dist_val) #แปลงเป็นอาเรย์
if w_check==True:
    idx=w_list.index(0) #หาตำแหน่งในลิสต์ว่าตรงไหนที่มีค่าน้ำหนักเป็น 0
    z_idw=dv[idx,1] ##กำหนดให้ใช้ค่าผันที่วัดได้จากสถานีที่ต้องกับพิกัดของกริดเล็กนั้นเลย โดยไม่สนใจ
    ว่าจะมีฝนที่วัดได้จากสถานีอื่นที่อยู่ในรัศมีนั้นถูกกีดกั้นตาม
    pcLayer.changeAttributeValue(feat.id(),3,str(z_idw))
    pcLayer.changeAttributeValue(feat.id(),4,len(dist_val))
else:
    wt=np.transpose(w_list)
    z_idw=np.dot(dv[:,1],wt)/sum(w_list) # ใช้ dot product เพื่อคำนวณหาค่าผันด้วยวิธีการ idw จะได้
    ค่าเดียวกัน
    # อ่านเพิ่ม https://www.guru99.com/numpy-dot-product.html
    pcLayer.changeAttributeValue(feat.id(),3,str(z_idw))
    pcLayer.changeAttributeValue(feat.id(),4,len(dist_val))

pcLayer.commitChanges() #ตั้งค่าให้หยุดการแก้ไขขั้นตอน C
print('เสร็จสิ้นการอัพเดทค่าพิกัดในกริดผลลัพธ์.....')
return pcLayer

#-----
def Timing_start():
    stt=datetime.datetime.now()
    print('*'*50)
    print ("started: {0}".format(datetime.datetime.now()))
    return stt

```

```

def Timing_used(stt):
    stp=datetime.datetime.now()
    print ("finished: {0}".format(datetime.datetime.now()))
    delTime=stp-stt
    print("ใช้เวลาวินาที:",delTime.seconds)
    print('***50)
    print("")

#-----



def run_main_flare_beamBlock(pLayer,rad_name,epsg,binSize,path_beamblock):
    #func นี้เรียกใช้กระบวนการหักหมุดของ radar flare และ beamblock
    #1. เริ่มกระบวนการ radar flare นำมายกไฟล์ "070"
    print('>>>> หา radar flare และ beamblock ประมาณค่าข้างเคียงด้วย idw....')

    #-----



    #1.1. ข่านshapefile ข้อมูลดินเจ้าร์เชียงราย
    # path_rad = "D:/tmp/PyQGIS/Plugin_practice/1data/rad_cri_utm201908041300.shp"
    # pLayer = QgsVectorLayer(path_rad, "Rad_"+rad_name, "ogr") #ค่าพิกัดขั้นข้อมูลที่จะใช้WGS84

    #-----



    #1.2. สร้างโดยคัดลอกข้อมูล 1.1 บางคอลัมน์ที่จะนำมาคำนวน
    # epsg = 32647 #กำหนดค่าพิกัดขั้นข้อมูลที่จะใช้WGS84
    uri = "Polygon?crs=epsg:" + str(epsg) + "&field=id:integer"&index=yes" # กำหนดค่าของข้อมูลขั้น
    เอกเตอร์
    opLayer = QgsVectorLayer(uri, 'rad_final_'+rad_name, 'memory') #สร้าง instance ของขั้นข้อมูล
    เอกเตอร์ตาม uri
    pr = opLayer.dataProvider() #เก็บค่า dataProvider ของขั้นข้อมูลรูปปิ๊ด

    #-เพิ่มหัวตารางโดยเก็บค่าบางคอลัมน์จากไฟล์ต้นฉบับบางส่วนและเพิ่มคอลัมน์ที่จะคำนวนสถิติเพื่อใช้ด
    radar flare
    pr.addAttribute(QgsField("value", QVariant.Double,'double', 10, 3),
                  QgsField("radialAng", QVariant.Double,'double', 10, 3),
                  QgsField("heightASL", QVariant.Double,'double', 10, 3),
                  QgsField("ID", QVariant.Int,len=3),
                  QgsField("numBinWithVal", QVariant.Double,'double', 10, 3),

```

```

QgsField("PC_BinWithVal", QVariant.Double,'double', 10, 3),
QgsField("Slope", QVariant.Double,'double', 10, 3),
QgsField("Corr", QVariant.Double,'double', 10, 3),
QgsField("Flare1", QVariant.Int),
QgsField("Flare2", QVariant.Int),
QgsField("BeamBlock", QVariant.Int,len=5),
QgsField("RQI", QVariant.Int,len=5])

opLayer.updateFields() #update attribute
#-----
#1.3ขนาดความละเอียดของกราฟิกวัดของเชิงราย ตรงนี้เพิ่มมาจาก 071
if binSize == 1000:
    #รีเซ็ตค่า
    #dist=240.0 #กิโลเมตร
    dist=237.0 #กิโลเมตร
else:
    dist=237.0*(1000/binSize) #กิโลเมตร

#-----
#1.4คัดลอกเริ่ดาร์
stt=Timing_start()
opLayer=copyRad(pLayer,opLayer,pr)
print('copyRad(pLayer,opLayer,pr)...'+rad_name)
Timing_used(stt)

#-----
#1.5ตรวจวัดRadar flare รอบที่ 1
PC_BinWithVal=85.0
Corr=0.85

stt=Timing_start()
opLayer,flare_ang,values=radarFlareDetectionRound1(opLayer,PC_BinWithVal,Corr,dist)

print('radarFlareDetectionRound1(opLayer,PC_BinWithVal,Corr,dist)...'+rad_name)
Timing_used(stt)

```

```

#-----
#1.6ตรวจวัดRadar flare รอบที่ 2
PC_BinWithVal=80.0
Corr=0.50

stt=Timing_start()
opLayer=radarFlareDetectionRound2(opLayer,PC_BinWithVal,Corr,flare_ang,values)
print('radarFlareDetectionRound2(opLayer,PC_BinWithVal,Corr,flare_ang,values)...'+rad_name)
Timing_used(stt)

#-----
#1.7การนำค่ากิริยาของ flare มาประมาณค่าด้วย idw เพื่อเติมค่า วิธีนี้ต้องใช้รัศมีตาม effective radius
# $\tan(\theta/2) = \text{size}/(2 * \text{dist})$  โดยที่ size คือด้านที่ต้องการหา ส่วน dist คือ ระยะของกิริยาน้ำที่กันสถานี
#โดยที่ theta คือ beamwidth ในที่นี่ให้ทดลอง 1 degree ก่อน ยิ่งใกล้ค่า size จะใหญ่ขึ้นตามระยะทาง
#หมายเหตุ: การเติม flare ด้วยการประมาณค่ารอบข้างเข้าไปจะประมวลผลช้ามากคือ มากกว่า 1000 วินที
#ดังนั้นเราในที่นี่ที่ภาคเหนือไม่แนะนำให้ใช้เนื่องจากมีการบดบังภูมิประเทศอย่างมาก ให้ได้แต่เพียงพื้นที่
#รอบเขื่อนพิชิตโดย
#ส่วนภาคเหนือต้องใช้การวิจัยอีกแบบ ที่จะสร้างภาพเฉพาะโดยใช้มุมยกที่สูงขึ้นไป หรือ pseudocappi
# geom = QgsGeometry(QgsPoint(sta_coord[0],sta_coord[1])) # นำพิกัดสถานีมาจากการประมาณดู
# คลาสของเรดาร์ shapefile
# sourceCrs = QgsCoordinateReferenceSystem(4326)
# destCrs = QgsCoordinateReferenceSystem(epsg)
# tr = QgsCoordinateTransform(sourceCrs, destCrs, QgsProject.instance())
# geom.transform(tr)

# uri = "Point?crs=epsg:" + str(epsg) + "&field=id:integer"&index=yes" # กำหนดค่าของข้อมูลขั้น
# เอกเทอร์รี
# ptLayer = QgsVectorLayer(uri, 'point_rad_CRI', 'memory') #สร้าง instance ของชั้นข้อมูลเวกเตอร์ตาม
# uri
# pr_pt = ptLayer.dataProvider()
# pt_rd=QgsFeature()
# pt_rd.setGeometry(QgsGeometry.fromPointXY(geom.asPoint()))
# pr_pt.addFeatures([pt_rd])
# ptLayer.updateExtents()

```

```
#เรียกฟังค์ชันประมาณค่า dbz ข้างเดียวกับเดาร์ flare ให้เข้าไปเติมบริเวณ flare ด้วย IDW
#  stt=Timing_start()
#  interpolateNearFlarebyIDW(phLayer,geom)
#  print(' interpolateNearFlarebyIDW(phLayer,geom)....')
#  Timing_used(stt)

#-----
#-----  

#2.1.เริ่มกระบวนการสกัดค่า beamblock ที่เป็นผลจากไฟล์ "072"
#2.1เปิดรasterเตอร์ผลลัพธ์การคำนวน beamblock
rLayer = QgsRasterLayer(path_beamblock , "BeamBlock_"+rad_name)
#-----  

#2.2สกัดค่า beamblock เข้าสู่วงเดือนร์เดาร์
stt=Timing_start()
opLayer=ExtractBeamBlockByCentroid(opLayer,rLayer)
print(' ExtractBeamBlockByCentroid(opLayer,rLayer)...'+rad_name)
Timing_used(stt)
#-----  

#2.3อัพเดทค่า RQI
stt=Timing_start()

opLayer=updateRQI(opLayer)

print(' updateRQI(opLayer)...'+rad_name)
Timing_used(stt)

print('เสร็จสิ้นการหา radar flare และ beamblock...'+rad_name)
return opLayer

##-----  

#-----  

#####
```

```

def
Mosaic_CRI_PHS(Flare_BB_CRI,Flare_BB_PHS,path_to_bound_layer,ds_s,epsg,rad_name1 ,rad_na
me2,file_time):
    #มาจากการไฟล์ "052"
    #กำหนดค่าพารามิเตอร์ของกริด C ผลลัพธ์ของการไม่เสีย
    uri = "Polygon?crs=epsg:" + str(epsg) + "&field=id:integer"&index=yes" # กำหนดค่าของข้อมูลชั้น
    เวกเตอร์
    pcLayer = QgsVectorLayer(uri, file_time, 'memory') #สร้าง instance ของชั้นข้อมูลเวกเตอร์ตาม uri
    pr_c = pcLayer.dataProvider() #เก็บค่า dataProvider ของชั้นข้อมูลรูปปั๊ด
    poly_c = QgsFeature() #สร้าง instance ของ feature ที่จะใช้สร้างรูปปั๊ดแต่ละชั้น
    #-กำหนดหัวใจวัด
    pr_c.addAttribute([QgsField("Area", QVariant.Double,'double', 10, 3),
                      QgsField("Perimeter", QVariant.Double,'double', 10, 3),
                      QgsField("Val", QVariant.Double,'double', 10, 3),
                      QgsField("NumPtsIntp", QVariant.Int,len=5),
                      QgsField("RainRad", QVariant.Double,'double', 10, 3)])
    pcLayer.updateFields() #update attribute
    #
    ##
    ##-----offset ตามค่าความยาวด้านข้างของกริด
    offset=ds_s*1 #จะใช้ค่านี้ในการเพิ่มพื้นที่ทั้งกริด A,B,C เพราะเราไม่ควรจะตัดให้พอดีเป๊ะกับพื้นที่ แต่ถ้าไม่
    เอากรีบเลี่ยง *0
    ##
    #-ตั้งค่าเวกเตอร์รูปปั๊ดจังหวัดเพื่อแสดงในແນ່ງທີ່
    prLayer = QgsVectorLayer(path_to_bound_layer, "Boundary_mosaic", "ogr") #ค่าพิกัดชั้นข้อมูลที่จะ
    ใช้WGS84 47n
    ext = prLayer.extent()
    (xmin, xmax, ymin, ymax) = (ext.xMinimum()-offset, ext.xMaximum()+offset,\n
                                 ext.yMinimum()-offset, ext.yMaximum()+offset)

    # QgsProject.instance().addMapLayers([prLayer])
    #
    #-สร้างกริด C
    #-กำหนดขอบเขตกริดC ผลลัพธ์ของการไม่เสีย ตามขอบเขตของ extent shapefile
    numCols,numRows,llx,lly,lu,ur,lr=calParaGrid(xmin, xmax, ymin, ymax,ds_s)

```

```

#-เรียกใช้ฟังค์ชันเพื่อสร้างกริดC
gridVal=-999
pcLayer=createGrid(numCols,numRows,llx,lly,ul,ur,lr,ds_s,pcLayer,poly_c,pr_c,gridVal)
#-----
#-สร้างชั้นข้อมูลเพื่อเก็บ雷达ที่ตัดของ CRI
#epsg = 32647 #กำหนดค่าพิกัดชั้นข้อมูลที่จะใช้ WGS84 47N UTM
uri = "Polygon?crs=epsg:" + str(epsg) + "&field=id:integer""&index=yes" # กำหนดค่าของชั้นข้อมูล
#----- เอกเตอร์
pcrLayer = QgsVectorLayer(uri, 'clip_rad_grid'+rad_name1, 'memory') #สร้าง instance ของชั้น
#----- เอกเตอร์
#----- เอกเตอร์
pr_cri = pcrLayer.dataProvider() #เก็บค่า dataProvider ของชั้นข้อมูลรูปปีด
poly_cri = QgsFeature() #สร้าง instance ของ feature ที่จะใช้สร้างรูปปีดแต่ละชั้น
#-add field to mem_layer to provide for feats.fields data
pr_cri.addAttribute([QgsField("value", QVariant.Double),
                     QgsField("heightASL", QVariant.Double)])
pcrLayer.updateFields() #update attribute

#----- เอกเตอร์
#----- เอกเตอร์
uri = "Polygon?crs=epsg:" + str(epsg) + "&field=id:integer""&index=yes" # กำหนดค่าของชั้นข้อมูล
#----- เอกเตอร์
pphsLayer = QgsVectorLayer(uri, 'clip_rad_grid'+rad_name2, 'memory') #สร้าง instance ของชั้น
#----- เอกเตอร์
#----- เอกเตอร์
pr_phhs = pphsLayer.dataProvider() #เก็บค่า dataProvider ของชั้นข้อมูลรูปปีด
poly_phhs = QgsFeature() #สร้าง instance ของ feature ที่จะใช้สร้างรูปปีดแต่ละชั้น
#-add field to mem_layer to provide for feats.fields data
pr_phhs.addAttribute([QgsField("value", QVariant.Double),
                     QgsField("heightASL", QVariant.Double)])
pphsLayer.updateFields() #update attribute

#-----
#----- เอกเตอร์
stt=Timing_start()
criLayer=clipRadarByProvince(Flare_BB_CRI,prLayer,pcrLayer,pr_cri)
print(' clipRadarByProvince(Flare_BB_CRI,prLayer,pcrLayer,pr_phhs)..... ')
Timing_used(stt)

```

```

stt=Timing_start()

phsLayer=clipRadarByProvince(Flare_BB_PHS,prLayer,pphsLayer,pr_ph)
print(' clipRadarByProvince(Flare_BB_CRI,prLayer,pphsLayer,pr_cri)..... ')
Timing_used(stt)

#-----
#กำหนดครึ่งที่จะใช้คำนวน IDW จากจุดศูนย์กลางของกริดผลลัพธ์
#ตรงนี้คือสิ่งที่แตกต่างจาก "050Mosaic..." 

# rd=ds_s*2 #ตัวเลขที่ใช้คูณ ถ้ามากจะทำให้การคำนวนจุดมาทำ idw มีเวลาใช้เวลานาน
rd=ds_s*1.0 #ตัวเลขที่ใช้คูณ ถ้ามากจะทำให้การคำนวนจุดมาทำ idw มีเวลาใช้เวลานาน
#เอาสัก 1 หรือ 2 เท่าพอดีๆ 0.5 หมายถึงเอาจุดเฉพาะในกริดมาประมาณค่า
#พบร่วมใช้ 0.5 ได้ผลดีกว่าค่าที่สูง คือ ไม่พับ overestimate หรือ underestimate มากนัก

#-อัพเดทค่าพิกัดในกริด C ด้วยการ intersect
stt=Timing_start()
updateFieldOutputByIntersectCentroid(crlayer,phsLayer,pcLayer,rd)
print(' updateFieldOutputByIntersectCentroid(pphsLayer,pcLayer,rd)..... ')
Timing_used(stt)
return pcLayer

#-----
def reprojectRad(path_rad,rad_name):
    #แปลงพิกัดเรดาร์ NAD83 เป็น WGS84 แล้ว UTM47N
    print('reprojection....'+rad_name)
    parameter = {'INPUT': path_rad, 'TARGET_CRS': 'EPSG:32647',
                 'OUTPUT': 'memory:Rad_Reprojected'}
    result = processing.run('native:reprojectlayer', parameter)
    path_rad=result['OUTPUT']

    return path_rad

#-----
def convertdbZ2RainRadar(a,b,pLayer):
    #แปลงค่าการสะท้อนของเรดาร์ dbZ เป็นฝนประมาณค่า rain rate ด้วยความสัมพันธ์ Z-R
    print('ทำการอัพเดทค่าพิกัดฝนประมาณค่าจากเรดาร์ในกริดผลลัพธ์.....')
    pLayer.startEditing() #ตั้งค่าให้เริ่มการแก้ไขข้อมูล

```

```

#เลือกเฉพาะค่า dbz ที่มากกว่า -999
# pLayer.selectByExpression("Val" > {'}.format(-999))

# for ft in pLayer.selectedFeatures():
for ft in pLayer.getFeatures():
    if ft['Val']==-999:
        r=0.0
    else:
        z=10**((float(ft['Val']))/10)
        r=(z/a)**(1/b)

    pLayer.changeAttributeValue(ft.id(),5,r) #mm/hr

pLayer.commitChanges() #ตั้งค่าให้หยุดการแก้ไขขั้นชั้นๆ
print('เสร็จสิ้นการอัปเดตค่าพีวีด์ฝนประมาณค่าจากเวดาไว้ในกริดผลลัพธ์.....')
return pLayer

#-----
def updateColumnFlare2Rainrad(pLayer,pfLayer):
    #อัปเดตcolumns เพื่อกำหนดค่ากริดที่ตกลอยู่ใน radar flare ให้เป็น nodata -999
    pLayer.startEditing() #ตั้งค่าให้เริ่มการแก้ไขขั้นชั้นๆ ค่า C

    all_features = {}
    index = QgsSpatialIndex() # Spatial index
    for ft in pfLayer.getFeatures():
        index.insertFeature(ft)
        all_features[ft.id()] = ft

    #loop เพื่อหาว่าผลลัพธ์ที่ไม่เสคนนี้มีกริดใดที่ตกลอยู่ใน radarflare
    for feat in pLayer.getFeatures():
        inGeom = feat.geometry()
        idsList = index.intersects(inGeom.boundingBox())
        #loop idsListB เพื่อหาจุดที่อยู่ในแนวรัศมีในการทำ idw ค่า dbz
        for id in idsList:
            if feat['Val']==-999 and all_features[id]['heightASL']<5000 and all_features[id]['Flare1']>0:

```

```

pLayer.changeAttributeValue(feat.id(),5,-999) #flare
break

pLayer.commitChanges() #ตั้งค่าให้หยุดการแก้ไขขั้นรื้อ模
return pLayer
#-----#
def utc2local(f):
    #แปลงชื่อไฟล์จาก UTC เป็น local time เพื่อจะนำไปเทียบกับฝนสถานี
    print('แปลงชื่อไฟล์จาก UTC เป็น local time...',f)
    from_zone = tz.tzutc()
    to_zone = tz.tzlocal()
    #แปลงเวลาจาก UTC เป็นเวลาท้องถิ่น
    Y=f[0:4]
    m=f[4:6]
    d=f[6:8]
    H=f[8:10]
    M=f[10:12]

    utc_time = dt.strptime(Y+'-'+m+'-'+d+' '+H+':'+M, "%Y-%m-%d %H:%M")
    utc = utc_time.replace(tzinfo=from_zone)
    local_time = utc.astimezone(to_zone)

    #-Local radar time variables, used for save to file
    Y='%'02d' % local_time.year
    m='%'02d' % local_time.month
    d='%'02d' % local_time.day
    H='%'02d' % local_time.hour
    M='%'02d' % local_time.minute
    fn_local=Y+m+d+H+M
    print (utc_time)
    print (local_time)
    print('แปลงชื่อไฟล์จาก UTC เป็น local time...',fn_local)
    print('***50)

    return fn_local

```

```

#-----
def exportRainEstimates2Tif(outp,final_rainRad):
    #ส่งออกผลลัพธ์ฝนประมาณค่าเป็น geotif เพื่อนำไป validate กับฝนสถานี
    processing.run("gdal:rasterize",
    {'INPUT':final_rainRad,
    'FIELD':'RainRad',
    'BURN':0,'UNITS':1,
    'WIDTH':2000,'HEIGHT':2000,
    'EXTENT':'522494.1237852003,678494.1237852003,1751182.977305875,2151182.977305875
[EPSG:32647]',
    'NODATA':-999,'OPTIONS':'',
    'DATA_TYPE':5,'INIT':None,
    'INVERT':False,
    'EXTRA':'',
    'OUTPUT':outp})
#-----

#####
#โค้ดหลัก
print('#'*50)
print('เริ่มการประมวลผลในสेकุนด์ 15 นาที.....')

#เตรียมชื่อไฟล์เดาร์เรียงราย
rad_cri
'D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/3data_processed/1radar_shp/1shp/1sontihn/1cri/' =
fn1 = []
fn1 += [f for f in os.listdir(rad_cri) if f.endswith('.shp')] #ลิสต์ชื่อไฟล์เดิม
ft1=[]
ft1+=[f[7:19]for f in fn1 ] #ตัดเฉพาะชื่อไฟล์ราย15นาที
#print(ft)

#เตรียมชื่อไฟล์เดาร์พิษณุโลก
rad_phn
'D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/3data_processed/1radar_shp/1shp/1sontihn/2phs/' =

```

```

fn2 = []
fn2 += [f for f in os.listdir(rad_ph) if f.endswith('.shp')] #ลิสต์ชื่อไฟล์เต็ม
ft2=[]
ft2+=[f[3:15]for f in fn2 ] #ลิสต์เวลาจากชื่อไฟล์ราย15นาที
#####
#ลูปหลักเพื่อเรียกฟังค์ชันย่ออย่างด้านบนมาไม่สุด
for f in ft1:
    # print('+',f)
    #ตรวจสอบว่ามีไฟล์ครบทั้งสองสถานีไหม ถ้าครบจะได้ทำโน้มสกัด
    if f in ft2:
        print('#'*25)
        print('เริ่มการประมวลผลไฟล์'+f+'....')
        #1.แปลงพิกัดและประมวลผล radar flare กับ beamblock
        #ตั้งค่าพารามิเตอร์ของเรดาร์เขียวราย
        pat = '*' + f + '*.shp'
        file = fnmatch.filter(fn1, pat)
        # print(file)

        path_rad =
"D:/Yang/1Research/0.2563.RadarMosGISYomNan/3data_processed/1radar_shp/1shp/1sontihn/1cri/"
+file[0]

        path_beamblock =
"D:/Yang/1Research/0.2563.RadarMosGISYomNan/3data_processed/3beamblockage/BeamBlockChi
angraiRaster.tif"

        rad_name='CRI'
        epsg = 32647 #กำหนดค่าพิกัดชั้นข้อมูลที่จะใช้WGS84
        binSize=500.0
        sta_coord=(99.88159181,19.96147083) #พิกัดสถานีเรดาร์ได้จาก shapefile

        #แปลง shapefile จาก EPSG:4269 - NAD83 ไปเป็น epsg:32647 UTM47N
        path_rad=reprojectRad(path_rad,rad_name)
        #ประมวลผล radar flare และ beamblock เขียวราย
        Flare_BB_CRI=run_main_flare_beamBlock(path_rad,rad_name,epsg,binSize,path_beamblock)

```

```
#-----  
#ตั้งค่าพารามิเตอร์ของเรดาร์พิชณ์โลก
```

```
pat = '*' + f + '*.shp'  
file = fnmatch.filter(fn2, pat)
```

```
path_rad  
= "D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/3data_processed/1radar_shp/1shp/1sontihn/2phs/  
" + file[0]
```

```
path_beamblock  
= "D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/3data_processed/3beamblockage/BeamBlockPhit  
sanulokRadarRaster.tif"
```

```
rad_name='PHS'  
epsg = 32647 #กำหนดค่าพิกัดชั้นข้อมูลที่จะใช้ WGS84  
binSize=1000.0  
sta_coord=(100.2179637,16.7754090) #พิกัดสถานีเรดาร์ได้จาก shapefile
```

```
#แปลง shapefile จาก EPSG:4269 - NAD83 ไปเป็น epsg:32647 UTM47N  
path_rad=reprojectRad(path_rad,rad_name)  
#ประมวลผล radar flare และ beamblock พิชณ์โลก  
Flare_BB_PHS=run_main_flare_beamBlock(path_rad,rad_name,epsg,binSize,path_beamblock)  
#-----
```

```
#####
#2. ไม่เสคเรดาร์  
path_to_bound_layer  
= "D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/1data_org/4shapefile_basemap/Yom_Basin_utm.s  
hp"  
ds_s=2000 #กริดผลลัพธ์ที่ต้องการตามระบบพิกัดห้ามต่ำกว่า 5000 เมตร  
epsg = 32647 #กำหนดค่าพิกัดชั้นข้อมูลที่จะใช้ WGS84  
rad_name1="CRI"  
rad_name2="PHS"  
  
stt=Timing_start()
```

```
mosaic_cri_phs=Mosaic_CRI_PHS(Flare_BB_CRI,Flare_BB_PHS,path_to_bound_layer,ds_s,epsg,rad
_name1,rad_name2,f)
#
mosaic_cri_phs=Mosaic_CRI_PHS(Flare_BB_CRI,Flare_BB_PHS,path_to_bound_layer,ds_s,epsg,rad
_name1,rad_name2)

print('Mosaic_CRI_PHS(Flare_BB_CRI,Flare_BB_PHS,path_to_bound_layer,ds_s,epsg,rad_name1 ,ra
d_name2,f).....')
Timing_used(stt)
#-----
#3.แปลงไฟ dbz เป็นไฟประมวลค่าด้วย z-r relationship MarshallPalmer
a=200.0
b=1.6

stt=Timing_start()
rainRad=convertdbZ2RainRadar(a,b,mosaic_cri_phs)
print('convertdbZ2RainRadar(a,b,mosaic_cri_phs).....')
Timing_used(stt)
#-----
#4.อัพเดทcolumns เพื่อกำหนดค่าก็อตที่ต่อกันใน radar flare ให้เป็น nodata -999
#ใช้Flare_BB_CRI และ Flare_BB_PHS เพื่อหา spatial index ในการ intersect กับ mosaic_cri_phs

#อ่านวิธีการใช้ spatial index เพื่อการเข้าถึงพีเจอร์ที่ไว้ชื่น
#https://gis.stackexchange.com/revisions/224954/4
#pLayer=rainRad
stt=Timing_start()
rainRad=updateColumnFlare2Rainrad(rainRad,Flare_BB_CRI)
print('updateColumnFlare2Rainrad(rainRad).....CRI')
Timing_used(stt)

stt=Timing_start()
final_rainRad=updateColumnFlare2Rainrad(rainRad,Flare_BB_PHS)
print('updateColumnFlare2Rainrad(rainRad).....PHS')
Timing_used(stt)
```

```
#-----  
#5. ส่งออกผลลัพธ์ฝนประมาณค่าเป็น geotif เพื่อนำไป validate กับผังสถานี  
outp_file=utc2local(f)+"."+tif" #ต้องนี้แตกต่างกับไฟล์ 090 เพราจะแปลงชื่อไฟล์เป็น local time  
  
outp='D:/Yang/1Research/0.2563.RadarMosGISYomNan/3data_processed/4mosaic_rainradar/1sonthi  
n/MarshallPalmerStratiformZR/15min/'+outp_file  
#     outp='D:/tmp/PyQGIS/Plugin_practice/1data/rainradarEstimates.tif'  
stt=Timing_start()  
exportRainEstimates2Tif(outp,final_rainRad)  
print('exportRainEstimates2Tif(outp).....')  
Timing_used(stt)  
#-----  
#####  
#6. แสดงผลในแผ่นที่  
QgsProject.instance().addMapLayers([final_rainRad])  
#-----  
print(' จบการประมวลผลไฟล์'+f+'....')  
print('#'*25)  
  
print('สิ้นสุดการประมวลผลในเดือนราย 15 นาที.....')  
print('#'*50)
```

ภาคผนวก ข

โค้ด Python สำหรับการคัดกรองข้อมูลฝันสถานีที่ไม่พบค่า None เลย
ในช่วงพายุเซัญเตัญจากสถานีทั้งหมด 43 สถานีในลุ่มแม่น้ำยม

ชื่อไฟล์ "1Gauge_QC_list_perfectStaion_manyDays"

20200612 QC ความสมบูรณ์ของสถานีฝนแบบหลายวัน เอามาจากไฟล์ "105"

โปรแกรมQC ความสมบูรณ์ของสถานีวัดฝน จกรองเฉพาะสถานีที่ไม่มี None เลยในช่วงวันที่ต้องการ
สองออกผลเป็นลิสต์สถานี จำนวนสถานีที่สมบูรณ์และไม่สมบูรณ์ในช่วงเวลาที่ต้องการ

ต่อไปจะนำลิสตรายชื่อสถานี ออกไปสร้างฝันราย 15 นาที รวมสถานีที่ผ่าน QC ในเวลาหนึ่ง

ข้อสังเกต โปรแกรมนี้จะกรองเอาแต่สถานีที่ไม่มี none ในช่วงเวลาที่ต้องการเพื่อให้มั่นใจว่าได้สถานีที่ไม่จำเป็น

บัญชา

แต่ก็ยังไม่ได้เขียนในโปรแกรมเพื่อกกรองสถานีที่ต้องการให้เหลือแค่สถานีที่ไม่มีค่าฝนหรือไม่ เพราะพบว่าจาก 14 สถานีที่ดินน้ำ มีถึง 4 สถานี

ที่ทุกวันในช่วงพายุฝนเชิงติดอยู่บนน้ำไม่มีค่าฝนเลย นั่นคือมีค่าเป็น 0 ทุกเร็คคอร์ด สรุปเหลือเพียง 10 สถานีเท่านั้น

```
import csv
import numpy as np
import os
import fnmatch
import re
```

```
def qc_gauge_completeness(st,ymd):
```

โปรแกรมQC ความสมบูรณ์ของสถานีวัดฝน จกรองเฉพาะสถานีที่ไม่มี None เลยในช่วงวันที่ต้องการ
สองออกผลเป็นลิสต์สถานี จำนวนสถานีที่สมบูรณ์และไม่สมบูรณ์ในช่วงเวลาที่ต้องการ

```
sta=[]
ko,kn=(0,0)
print('....เริ่มการตรวจสอบความสมบูรณ์ของฝนสถานีวันที่...'+ymd)
for sn in st: #ลูปชื่อสถานี
    fn=sn+'.csv'
    #      print('>>>',sn)
```

```
if os.path.isfile(pathGauge+fn): #ตรวจสอบว่ามีข้อมูลไฟล์สถานีไหม
    with open(pathGauge+fn, 'r') as f:
```

```

g = list(csv.reader(f, delimiter=','))
g=np.asarray(g)

#กรองเฉพาะวันเวลาที่ต้องการไปใช้งาน
pat=ymd[6:]+ '/' + ymd[4:6] + '/' + ymd[0:4]

#กรองเฉพาะที่มีเวลาตรงกับ pat
#1.ต้องจัดระเบียบไฟล์ก่อน โดยเอาวันเดือนปีของคอลัมน์2ไปแทนคอลัมน์1 ของ g
a=[]
a+=[re.sub(i[-9:], " ", i) for i in g[:,1]]
g[:,0]=a

#2.เอาช่วงปีเดียวที่เหลือวันเดือนปีของคอลัมน์2ไปแทนคอลัมน์2 ของ g (แทนที่เดิมเลย)
b=[]
b+=[re.sub(i[0:11], " ", i) for i in g[:,1]]
g[:,1]=b

#ใช้ pat (ymd) หาว่าช่วงเวลาที่ต้องการอยู่ในແດວใดบ้าง
data=g[np.where(np.isin(g[:,0], pat))]

#อ่าน https://stackoverflow.com/questions/51030608/finding-indices-of-values-in-2d-numpy-array
#อ่าน https://stackoverflow.com/questions/40976714/string-slicing-in-numpy-array

#ตรวจสอบความสมบูรณ์ของ data ว่ามี "None" ไหม ถ้าไม่มีให้เก็บในลิสเพื่อจะ write เป็น csv เก็บไว้เพื่อไปร่วมไฟล์ต่อ
if len(data[data[:,2]=='None'])==0:
    sta.append(sn) #write ออกไปเป็นไฟล์ของสถานีที่สมบูรณ์ นำไป validate
    ko +=1
    print('....สถานี ',sn,' สมบูรณ์ มีความยาว None',len(data[data[:,2]=='None']))
else:
    kn +=1
    print('....สถานี ',sn,' *ไม่* สมบูรณ์ มีความยาว None',len(data[data[:,2]=='None']))
print('....เสร็จสิ้นการตรวจสอบความสมบูรณ์ของฝ่ายสถานีวันที่...'+ymd)
return sta,ko,kn

```

```

#-----
#อ่านสถานี
gaugeName='D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/1data_org/2gauge/' #เป็นไฟล์เดอร์
ของไฟล์ข้อมูลสถานีพร้อมพิกัด
pathGauge='D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/1data_org/2gauge/1gauge_org/' ##
เป็นไฟล์เดอร์เก็บข้อมูลติดไฟล์สถานี
with open(gaugeName+'1สถานีวัดฝนกรมอุตุฯลุ่มน้ำท่วม2018.csv', 'r') as f:
    st = list(csv.reader(f, delimiter=','))

st=np.asarray(st[1:][:,0]) #ข้อมูลสถานีไม่เข้า header

#รันเวลาที่ต้องการQC
time=['20180715','20180716','20180717',
      '20180718','20180719','20180720','20180721','20180722','20180723','20180724'] #พากย์sonthai
res=[]#เก็บผลการqc
i=0
for ymd in time:
    i+=1
    print('***50')
    print('ตรวจสอบความสมบูรณ์ของสถานีวัดฝนวันที่',i,';',ymd )
    print('***50')
    st,ko,kn=qc_gauge_completeness(st,ymd) #st,ko,kn รายชื่อสถานีที่สมบูรณ์,จำนวนสถานีที่สมบูรณ์,
    จำนวนสถานีที่ไม่สมบูรณ์
    res.append((ymd,st,ko,kn)) #เก็บผลการqc

#เช็ค res[6][1] เป็น csv เพื่อนำไปใช้รวมฝน
gf = open(gaugeName+'list_passedQG_gauges.csv', 'w')
gaugeLists=res[6][1]
with gf:
    w = csv.writer(gf)
    w.writerow(gaugeLists)

print('***50')
print('เสร็จสิ้นการตรวจสอบความสมบูรณ์ของสถานีฝน.....')

```

ภาคผนวก ค

โค้ด PyQGIS ในการคำนวณการบดบังของลำบีมด้วยภูมิประเทศที่ได้จากการสำรวจพิษณุโลก ส่วนเรดาร์เขียงรายสามารถนำไปประยุกต์ตัดแบ่งโค้ดเองได้ เพราะโค้ดเหมือนกัน โดยต้องเตรียม DEM ตัดไว้ให้พอดีกับสถานีที่ต้องการหาค่าการบดบังของลำบีม

ชื่อไฟล์ "7BeamBlockAnalysisPHSbyTerrainComplete"

จากไฟล์068

20200528 การคำนวณการบดบังของบีมจากภูมิประเทศแบบสมบูรณ์ จะนำไปเป็นไฟล์ต้นแบบเพื่อทำ spatial intersect กับทุกการสแกน

การสร้างรูปปีกของเรดาร์ของมุมยกแรกในทุกอชิมุทและทุกระยะทางด้วยการคำนวณ radius effective ตามระยะทางและมุมของเรดาร์

ใช้ค่ามุมจากกราฟอ่านไฟล์ต้นฉบับเพื่อให้ได้อัชิมุของเรดาร์พิชณ์โลก และนำมามาหารสองกับมุมซึ่งเคียงเพื่อให้ได้ตระกูลางของแนวสแกน

1. กำหนดค่าพิกัดสถานีแบบ geo และแปลงเป็น utm
2. อ่านข้อมูล shape ของเรดาร์พิชณ์โลกเพื่อ get ค่าอัชิมุจริงที่จะนำไปคำนวณ beamwidth
3. สร้างรูปปีกผลลัพธ์ใน memory
4. สร้างคล้มเนื้อของรูปปีก
5. คำนวณพิกัดเพื่อให้ได้จุดของมุมรูปปีก
6. นำพิกัดของจุดเหล่านี้มาสร้างเป็นรูปปีก
7. คำนวณการบดบังสำหรับภูมิประเทศ (DEM)
7. อัพเดทตารางและແນ່ງທີ່
8. แสดงผลແນ່ງທີ່
9. สงผลออกเป็น shapefile และ วາສเตอร์ (แต่เราจะใช้วາສเตอร์ในการประมวลผลได้ดีต่อไป)

นำผลการคำนวณ beamblock นี้ไปใช้ทำการบดบังของแต่ละไฟล์ที่สแกนมาทุกๆ 15 นาที

```
import math
import numpy as np
print('คำนวณสำหรับภูมิประเทศบดบังโดยใช้โครงสร้างของเรดาร์จริงของเรดาร์พิชณ์โลก.....')
#-----
def CalcGeographicAngle(arith):
    #to convert arithmetic angle to geographic angle
    return (360 - arith + 90) % 360
#-----
def createPolyRadAlongAzimuth(numRange,dist,binSize,elevAng,pt_rd,pLayer,pr,h_rd,rad_alt):
    print('สร้างรูปปีกของกริดตามแนวอชิมัท....')
    #ค่าพารามิเตอร์สำหรับคำนวณความสูงของสำหรับบัน្តาที่เละเป็นกอลาง
    pi=3.14159265358979
```

```

re=6374000 # รัศมีโลก (เมตร)
rr=4.0/3.0*re # earth effective radius

#ลูปตามแนวมุมอิมพ์จากไฟล์ shape ตั้งฉบับ
k=0
for az in values:
    azimuth=CalcGeographicAngle(az)#แปลงเป็นมุมอิมพ์
    #    print('มุมอิมพ์ที่:',az)

    #หาความกว้างของลำบึมตามกรณี
    if k==0:
        beamWidth=((values[0]+values[1])/2) - (values[len(values)-1]+values[0])/2)%180
        #    print('>>0',k,beamWidth)

    if k==(len(values)-1):
        beamWidth=((values[k]+values[0])/2)-((values[k-1]+values[k])/2)
        #    print('>>L',k,beamWidth)

    if k>0 and k<(len(values)-1):
        beamWidth=((values[k]+values[k+1])/2)-((values[k-1]+values[k])/2)

    #เมื่อนำมาบังกรณี error ให้มุมตรงกลางมากำหนด beamwidthแทน
    if (beamWidth < -5.0 or beamWidth>5.0):
        beamWidth=(values[int(len(values)/2)+1]-values[int(len(values)/2)-1])/2
        #    print('>>+',k,beamWidth)

    #คำนวนมุมก่อนและหลังเพื่อนำไปหาค่าพิกัดปลายรูปนิ่ง
    angle1 = math.radians(azimuth-(beamWidth/2)) #แปลงมุมอิมพ์เป็นเรเดียนสำหรับมุมก่อนหน้า
    angle2 = math.radians(azimuth+(beamWidth/2)) #แปลงมุมอิมพ์เป็นเรเดียนสำหรับมุมด้านหลัง

    #ลูปตามแนวระยะทางที่กำหนด
    i=0
    for dt in range(int(dist)):
        #    print('ระยะ:',dt)

```

```

#ระยะทางตามแนวรัศมีในหน่วยเมตร
distance=(dt+1)*binSize #เมตร

if i==0:#กรณีเป็น bin ที่ 1
    dist_x1, dist_y1 = (distance * math.cos(angle1), distance * math.sin(angle1))
    dist_x2, dist_y2 = (distance * math.cos(angle2), distance * math.sin(angle2))
    xp1, yp1 = (pt_rd[0] + dist_x1, pt_rd[1] + dist_y1)
    xp2, yp2 = (pt_rd[0] + dist_x2, pt_rd[1] + dist_y2)

points = [QgsPointXY(pt_rd[0],pt_rd[1]),QgsPointXY(xp1,yp1),\
           QgsPointXY(xp2,yp2)]


else:
    dist_x1, dist_y1 = ((distance-binSize) * math.cos(angle1), (distance-binSize) * \
math.sin(angle1))
    dist_x2, dist_y2 = ((distance-binSize) * math.cos(angle2), (distance-binSize) * \
math.sin(angle2))
    xp1, yp1 = (pt_rd[0] + dist_x1, pt_rd[1] + dist_y1)
    xp2, yp2 = (pt_rd[0] + dist_x2, pt_rd[1] + dist_y2)

    dist_x3, dist_y3 = (distance * math.cos(angle1), distance * math.sin(angle1))
    dist_x4, dist_y4 = (distance * math.cos(angle2), distance * math.sin(angle2))
    xp3, yp3 = (pt_rd[0] + dist_x3, pt_rd[1] + dist_y3)
    xp4, yp4 = (pt_rd[0] + dist_x4, pt_rd[1] + dist_y4)
    points = [QgsPointXY(xp1,yp1), QgsPointXY(xp2,yp2),\
               QgsPointXY(xp4,yp4), QgsPointXY(xp3,yp3) ]


#คำนวน size ตาม effective radius ของมุมและระยะทาง
size=np.tan(np.radians(beamWidth/2))*2*(distance/1000) #หน่วย กม.
# print ("พิกัดที่คำนวนได้ตามมุม:",az,size,i+1,xfinal,yfinal )

 #-คำนวนความสูงของลำปิงในหน่วยเมตรcal beam height + rad altitude in meter.

```

```

beam_height= np.sqrt((distance)**2+rr**2+(2*(distance)*rr*np.sin(np.radians(elevAng)))-rr+h_rd+rad_alt #เมตร

#สร้างรีนวูปโดยจากจุดข้อมูลจุดด้านบน ดูค่าตัวแปร poly ด้านบน
poly = QgsFeature()
poly.setGeometry(QgsGeometry.fromPolygonXY([points])) #ตั้งค่าเรขาคณิต
poly.setAttributes([i,elevAng,az,distance,float(size),float(beam_height)]) #อัพเดทค่าในตาราง
pr.addFeatures([poly])
pLayer.updateExtents()
i+=1
k+=1
print('เสร็จสิ้นการสร้างรูปปีด....')
#-----
def findUniqueAngle(ppLayer):
    idx = ppLayer.fields().indexOf('radialAng')
    values = ppLayer.uniqueValues(idx)
    val=list(values)
    val.sort()
    list_ang=[]
    i=0
    #ต้องการนำมุมมาหารกันเพื่อให้มุมอยู่ตรงกลางรูปปีด
    for ang in val:
        if i<(len(val)-1):
            ang_cal=((val[i]%360)+(val[i+1]%360))/2
            #เขียนกำกับเพื่อแก้ปัญหา มุมของศูนย์ห้ามที่คำนวนแล้วน้อยกว่า 300.0 โดยเลข 300.0 เป็นเพียงเลข
            #ตัวอย่างตามความเป็นจริง
            if i==(len(val)-2) and ang_cal<300.0: ang_cal=(list_ang[i-1]+1)%360.0

        if i==(len(val)-1): #มุมศูนย์ห้าม จะถูกนำมาร่วมกับมุมแรกแล้วหารสอง
            ang_cal=(val[i]%360+(val[0]%360))/2

        list_ang.append(ang_cal)

    i+=1
#   print(list_ang)

```

```

return list_ang

#-----
def ExtractDEMByCentroid(pLayer):
    print('สกัดค่า DEM....')
    pLayer.startEditing()
    for f in pLayer.getFeatures():
        #สร้างจุดจาก centroid ของรูปปิ๊ด
        pts = f.geometry().centroid().asPoint() #get ค่าพิกัด centroid ของกริดแล้วสร้างให้เป็นจุด

        #นำค่าจุดมาสกัดความสูง DEM
        val, res = rlayer.dataProvider().sample(QgsPointXY(pts[0],pts[1]), 1) #val คือค่า dem ที่รีเซ็ตกลับ
        #มาจากการสกัดด้วยจุด sample

        #อัพเดทเฉพาะค่า DEM ในไฟล์ pLayer
        pLayer.changeAttributeValue(f.id(),6,val)

    pLayer.commitChanges()
    print('เสร็จสิ้นการสกัดค่า DEM....')

#-----
def CalBeamBlock(pLayer):
    print('คำนวนการรอบดบังของบีมจากภูมิประเทศ...')
    pLayer.startEditing()

    #ลูปแต่ละบีมเพื่อหาการรอบดบังของลำบีม
    i=0
    for az in values:
        #กรองฟีเจอร์ด้วยบีม
        pLayer.selectByExpression("radialAng" = "{}".format(az))

        #ลูปฟีเจอร์ตามบีมที่ได้กรองไว้เพื่อหาลำบีมที่ถูกบดบัง
        for k,f in enumerate(pLayer.selectedFeatures()):
            #print('ลำดับ',k)
            if f['heightASL']<=f['DEM_ASL']:
                dis_block=f['horizonDis']

```

```

i+=1
#กรองฟีเจอร์ด้วยมุมและระยะทางที่พบร่วมกับมุมและระยะทางที่พบร่วมกับบัง
pLayer.selectByExpression("radialAng" = {} AND "horizonDis">>= {}'.format(az,dis_block))
for fb in pLayer.selectedFeatures():
    pLayer.changeAttributeValue(fb.id(),7,1) #ลำบีมซึ่งที่ถูกบดบังโดยภูมิประเทศ

    break #เมื่ออัพเดทคล้มนี้เสร็จให้ออกลูปใบใหญ่ เพื่อคำนวนมุมถัดไป
pLayer.commitChanges()

print('จำนวนมุมทั้งหมดที่ถูกภูมิประเทศบดบัง:',i)
print('เสร็จสิ้นการคำนวนการบดบังของปีมีจากภูมิประเทศ...')

#-----
def updateColumnBeamBlock(pLayer):
    print('อัพเดทคล้มนี้ BeamBlock สำหรับบีมที่ไม่ถูกบดบัง...')
    pLayer.startEditing()
    for az in values:
        #กรองฟีเจอร์ด้วยมุม
        pLayer.selectByExpression("radialAng" = {}'.format(az))

        #ลูปฟีเจอร์ตามมุมที่ได้กรองไว้เพื่อหาลำบีมที่ถูกบดบัง
        for k,f in enumerate(pLayer.selectedFeatures()):
            if f['BeamBlock'] == NULL:
                pLayer.changeAttributeValue(f.id(),7,0) #ลำบีมซึ่งที่ไม่ถูกบดบังโดยภูมิประเทศ
        pLayer.commitChanges()
        print('เสร็จสิ้นการอัพเดทคล้มนี้ BeamBlock สำหรับบีมที่ไม่ถูกบดบัง...')

#-----
#1.แปลงพิกัดสถานีพิชณ์โดยจาก geo เป็น utm
geom = QgsGeometry(QgsPoint(100.2174233,16.7755399)) # นำพิกัดสถานีมาจากการเก็บ
geom = QgsGeometry(QgsPoint(100.2179637,16.7754090)) # นำพิกัดสถานีมาจากการแปลง
ของเรดาร์ shapefile
sourceCrs = QgsCoordinateReferenceSystem(4326)
destCrs = QgsCoordinateReferenceSystem(32647)
tr = QgsCoordinateTransform(sourceCrs, destCrs, QgsProject.instance())
geom.transform(tr)

```

```

epsg=32647

uri = "Point?crs=epsg:" + str(epsg) + "&field=id:integer""&index=yes" # กำหนดค่าของข้อมูลรูปแบบ
ptLayer = QgsVectorLayer(uri, 'sta_rad_PHS', 'memory') #สร้าง instance ของข้อมูลเวกเตอร์ตาม uri
pr_pt = ptLayer.dataProvider()
pt_rd=QgsFeature()
pt_rd.setGeometry(QgsGeometry.fromPointXY(geom.asPoint()))
pr_pt.addFeatures([pt_rd])
ptLayer.updateExtents()

#-----
#2.สร้างชั้นรูปปีกใน memory
uri = "Polygon?crs=epsg:" + str(epsg) + "&field=id:integer""&index=yes" # กำหนดค่าของข้อมูลรูปแบบ
plLayer = QgsVectorLayer(uri, 'poly_beamblock', 'memory') #สร้าง instance ของข้อมูลเวกเตอร์ตาม uri
pr = plLayer.dataProvider() #เก็บค่า dataProvider ของข้อมูลรูปปีก
#poly = QgsFeature() #สร้าง instance ของ feature ที่จะใช้สร้างรูปปีกเดลล์ชีน
#-add field to mem_layer to provide for feats.fields data
pr.addAttribute([QgsField("elevAngle", QVariant.Double,'double', 10, 3),
                QgsField("radialAng", QVariant.Double,'double', 10, 3),
                QgsField("horizonDis", QVariant.Double,'double', 10, 3),
                QgsField("size", QVariant.Double,'double', 10, 3),
                QgsField("heightASL", QVariant.Double,'double', 10, 3),
                QgsField("DEM_ASZ", QVariant.Double,'double', 10, 3),
                QgsField("BeamBlock", QVariant.Int,len=5)])#1=block,0=Unblock

plLayer.updateFields() #update attribute

#-----
#3.ตั้งค่าพารามิเตอร์เพื่อสร้างรูปปีก
#พิกัดสถานีเรดาร์
pt_rd=geom.asPoint()

#รัศมีเรดาร์
#dist=240.0 #กิโลเมตร
dist=237.0 #กิโลเมตร

```

```

beamWidth=1.0 #องศา

#มุมที่ต้องการ
#az=270.0
#azimuth=CalcGeographicAngle(az)#แปลงเป็นมุมอชิม็ท

#จำนวนอชิม็ท
numRange=360 #องศา

#ขนาดความละเอียดของกราฟระหว่างวัด
binSize=1000.0

if binSize == 1000:
    #รีเซ็ตเดคาร์ต
    #dist=240.0 #กิโลเมตร
    dist=237.0 #กิโลเมตร
else:
    dist=237.0*(1000/binSize) #กิโลเมตร

#มุมยก
elevAng=0.5 #องศา

#ค่าพารามิเตอร์สำหรับคำนวนความสูงของลำบึมจากระดับน้ำทะเลปานกลาง
h_rd=30.0 #-ความสูงของหอดูอยเดคาร์ตของกรมอุตุนิยมวิทยา (เมตร)
rad_alt=47 # ความสูงสถานีจากระดับน้ำทะเลปานกลาง (เมตร)
#-----
#4.หามุมตามเดคาร์ตันฉบับ ตรงนี้ที่แตกต่างจากไฟล์ "060"
path_to_rad_layer = "D:/tmp/PyQGIS/Plugin_practice/1data/rad_phis_utm201908041300.shp"
ppLayer = QgsVectorLayer(path_to_rad_layer, "rad_phis_utm201908041300", "ogr") #ค่าพิกัดชั้นข้อมูล
ที่จะใช้ WGS84

values=findUniqueAngle(ppLayer)
#-----
#5.สร้างชั้นรูปปีกเดคาร์ตตามแนวมุมอชิม็ทที่กำหนด

```

```
createPolyRadAlongAzimuth(numRange,dist,binSize,elevAng,pt_rd,pLayer,pr,h_rd,rad_alt)

#-----
#6. ถอดความสูง DEM เพื่อนำมาใส่ในชั้น pLayer
path_to_DEM = "D:/tmp/PyQGIS/Plugin_practice/1data/DEM_for_PHS_radar.tif"
rlayer = QgsRasterLayer(path_to_DEM, "DEM_PHS")

#ถอดความสูง
ExtractDEMByCentroid(pLayer)
#-----
#7. คำนวน beamblock
CalBeamBlock(pLayer)
#-----
#8. อัพเดตชั้น BeamBlock สำหรับบีมที่ไม่ถูกบดบัง
updateColumnBeamBlock(pLayer)
#-----
#9. ส่งออกผลการคำนวน beamblock ในรูป shapefile polygon
path_to_rad_layer = "D:/tmp/PyQGIS/Plugin_practice/1data/BeamBlockPhitsanulokRadar.shp"
QgsVectorFileWriter.writeAsVectorFormat(pLayer,path_to_rad_layer,'utf-8',driverName='ESRI Shapefile')

#-----
#10.rasterize beamblock เพื่อส่งออกเป็นรูป raster
outp = "D:/tmp/PyQGIS/Plugin_practice/1data/BeamBlockPhitsanulokRadarRaster.tif"
processing.run("gdal:rasterize",
{'INPUT':pLayer,'FIELD':'BeamBlock',
'BURN':0,
'UNITS':0,
'WIDTH':500,
'HEIGHT':500,
'EXTENT':'392807.3827114203,866803.0786619843,1618109.471501457,2092107.6119104286
[EPSG:32647]',
'NODATA':0,'OPTIONS':'',
'DATA_TYPE':5,
'INIT':None,
```

```
'INVERT':False,  
'EXTRA':'',  
'OUTPUT':outp})  
  
#-----  
#11.แสดงผลในแผนที่  
QgsProject.instance().addMapLayers([pLayer])  
QgsProject.instance().addMapLayers([ptLayer])  
  
#-----  
print('จบการคำนวน.....')
```







จากไฟล์ที่ชื่อ "6RainAcc24h_perfectStions.py"

20200612

ฝนสะสมราย24ชั่วโมงของสถานีที่มีข้อมูลฝนสมบูรณ์ ปรับปรุงจาก "110"

```

import csv
import numpy as np
import os
import fnmatch

gaugeName='D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/1data_org/2gauge/' #เป็นไฟล์เดอร์
ของไฟล์ของสถานีพร้อมพิกัด
pathGauge='D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/1data_org/2gauge/1gauge_org/' #
เป็นไฟล์เดอร์เก็บข้อมูลเดิมฝนแต่ละสถานี
outpRain24h='D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/3data_processed/2gauge/1sontihn/
24h/#เป็นไฟล์เดอร์เก็บข้อมูลผลลัพธ์ฝนราย24ชมของสถานีที่สมบูรณ์ที่ได้จากไฟล์1'

#สถานีนี้วัดฝนที่สมบูรณ์จากไฟล์1"
with open(gaugeName+'list_passedQG_gauges.csv', 'r') as f:
    st = list(csv.reader(f, delimiter=','))[0]
    print(st)

    print('เริ่มการสะสมฝน24ชั่วโมงจากการดึงข้อมูลฝนราย15นาทีของแต่ละสถานีฝน.....')
    #วันเวลาที่ต้องการรวมไฟล์ฝนราย24ชั่วโมง
    time=['20180715','20180716','20180717',
    '20180718','20180719','20180720','20180721','20180722','20180723','20180724'] #พากย์sontihn
    #ลูปเวลาเพื่อดึงตามเวลา 15 นาทีในแต่ละไฟล์
    for dd in time: #วัน
        #ลูป 1 รอบใน 1 วัน
        for n in range(1): #2 คือ 24/24 สะสมฝนทุกๆ12ชม.
            print('ทำการสะสมฝนราย24ชั่วโมง...'+dd+":"+str(n*24).format(n*24))
            start=n*24
            stop=start+24

```

```

gg=[] #ลิสเก็บชื่อไฟล์ที่หาได้ในแต่ละช่วงไม่กี่ที่ต้องการ ในที่นี้ทั้ง24ช่วงไม่
#ลูปเพื่อเปิดแต่ละสถานีเพื่อที่จะดึงตามเวลา 15 นาทีในแต่ละไฟล์
for s in st:#สถานีวัดฝนที่สมบูรณ์จากไฟล์1"
    gf=pathGauge+s+'.csv' #ข้อมูลไฟล์สถานีใหม่
    if os.path.isfile(gf): #ตรวจว่ามีข้อมูลไฟล์สถานีใหม่

        #เปิดไฟล์สถานี s
        with open(gf, 'r') as f:
            g = list(csv.reader(f, delimiter=','))
            g=np.asarray(g)

        #กรองเอาเฉพาะที่มีเวลาตรงกับ pat
        #1.ต้องจัดระเบียบไฟล์ก่อน โดยเอาวันเดือนปี+ช่วงไม่ของคล้มัน2ไปแทนคล้มัน1 ของ g
        a=[]
        a+=[re.sub(i[-6:], " ", i) for i in g[:,1]]
        g[:,1]=a

        sum_rr=0.0
        i=1
        for hh in range(start,stop): #ลูป12ช่วงไม่ที่ต้องการ

            pat=dd[6:]+"/"+dd[4:6]+"/"+dd[0:4]+" "+":{02d}.".format(hh) #ใช้คล้มัน1 ของ g
            if i==1:hh_start=dd+":{02d}.".format(start) #ชื่อไฟล์ช่วงไม่แรกของ24ช่วงไม่ที่กำลังจะมา
            tt=dd+":{02d}.".format(hh) #เวลา yyyymmddhh
            print('>>>','tt:',tt,'pat:',pat)

            #ใช้ pat (ymd) หาว่าช่วงเวลาที่ต้องการอยู่ในแกร่งได้บ้าง
            data=g[np.where(np.isin(g[:,1], pat))]
            rr=sum([float(data[i][2])for i in range(len(data))]) #รวมฝนในช่วงไม่ที่ตรงกับ pat
            sum_rr+=rr
            i+=1

            # print('....',len(data),rr,sum_rr)
            data=[str(s),hh_start,str(sum_rr)]

```

```
#เก็บข้อมูลฝนสะสมราย24 ชั่วโมงที่ดีงมาจากฝน15นาทีมาเก็บในลิสตันครบถ้วนสถานี
gg.append((data))

#เชฟ gg เป็น csv ตามชื่อไฟล์
#เชฟเป็นไฟล์ csv ของเวลาราย15นาทีนั้นๆ tt+'.csv'
gf = open(outpRain24h+hh_start+'.csv', 'w',newline='') #newline="" เพื่อป้องกันการเพิ่มบรรทัดตอน
เชฟ csv

with gf:
    w = csv.writer(gf)
    w.writerows(gg)

#    break
#    break

print('เสร็จสิ้นการสะสมฝน24ชั่วโมงจากการดึงข้อมูลฝนราย15นาทีของแต่ละสถานีฝน.....')
```

ภาคผนวก จ
ให้ด้วย PyQGIS ในสกัดค่าฝนประมาณค่าราย 24 ชั่วโมงตามตำแหน่งสถานีวัด
แผนแบบ Time series ในช่วงพายุเซิญติญ

ไฟล์ชื่อว่า "13extract_validate_radar_gauge24h_marshallPalmer.py"
20200613 การสกัดค่าฝน雷达และฝนสถานี Marshall Palmer ปรับปรุงจากไฟล์113

การสกัดค่าฝน雷达และฝนสถานีในแต่ละเวลา ภายใน 24 ชม. ปรับปรุงจากไฟล์113

ต่อไปจะใช้ผลลัพธ์จากไฟล์นี้ในการ validate ด้วยสถิติ bias,mse,rmse

```
import csv
import numpy as np

def transformGeo2UTM(lon,lat):
    #แปลงพิกัด geo เป็น utm47N
    geom = QgsGeometry(QgsPoint(float(lon),float(lat))) # นำพิกัดสถานีฝนมาสร้างเป็น geometry
    sourceCrs = QgsCoordinateReferenceSystem(4326)
    destCrs = QgsCoordinateReferenceSystem(32647)
    tr = QgsCoordinateTransform(sourceCrs, destCrs, QgsProject.instance())
    geom.transform(tr) #แปลง geo เป็น utm
    x_utm,y_utm=(geom.asPoint()[0],geom.asPoint()[1]) #get ค่า utm
    return x_utm,y_utm

def saveValidate(res,path_val,fileOutput):
    #เข้าไฟล์ผลลัพธ์การ validate เป็น csv
    val_res = open(path_val+fileOutput, 'w',newline="") #newline="" เพื่อป้องกันการเพิ่มบรรทัดตอนเข้า
    csv
    with val_res:
        w = csv.writer(val_res)
        w.writerows(res)

#-----
gaugeName='D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/1data_org/2gauge/' #เป็นไฟล์เดอร์
ของไฟล์ข้อมูลสถานีพร้อมพิกัด
```

```

path_rg='D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/3data_processed/2guage/1sontihn/24h/'  

#ไฟล์เดอร์เก็บฝนสถานี24ชม.  

path_rr='D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/3data_processed/4mosaic_rainradar/1so  

ntiinh/MarshallPalmerStratiformZR/24h/' #ไฟล์เดอร์เก็บฝน雷达ร์24ชม.  

path_val='D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/3data_processed/5outp_validations/1so  

ntiinh/MarshallPalmerStratiformZR/' #ไฟล์เดอร์เก็บผลลัพธ์สกัดข้อมูลrg+rr

#อ่านสถานีวัดฝนพังงาที่ในลุ่มน้ำยม  

with open(gaugeName+'สถานีวัดฝนกรมอุตุลุ่มน้ำยม2018.csv', 'r') as f:  

    st_all = list(csv.reader(f, delimiter=','))

#อ่านสถานีวัดฝนที่สมบูรณ์จากไฟล์105
with open(gaugeName+'list_passedQG_gauges.csv', 'r') as f:  

    st = list(csv.reader(f, delimiter=','))[0]

#สร้างลิสต์ฝน雷达ร์15นาที
fn_rr = []
fn_rr += [f for f in os.listdir(path_rr) if f.endswith('.tif')]

#สร้างลิสต์ฝนสถานี15นาที
fn_rg = []
fn_rg += [f for f in os.listdir(path_rg) if f.endswith('.csv')]

#-----
#แปลงพิกัดสถานีเป็นภาพพื้นดินจาก Geo เป็น UTM
st_all=np.asarray(st_all)
s_utm=[]
for s in st:
    #หาพิกัด geo(lon,lat) ในลิสต์สถานีทั้งหมด
    id=np.where(st_all==s)[0][0] #หาอินเด็กซ์แควร์ที่มีรหัสสถานีตรงกับ s
    lon,lat=(st_all[id][1],st_all[id][2]) #get ค่าพิกัดสถานีของ s จากไฟล์สถานีหลัก

    #แปลงพิกัด geo เป็น utm47N
    x_utm,y_utm=transformGeo2UTM(lon,lat)
    s_utm.append((s,x_utm,y_utm))

```

```

print('เริ่มการ validate.....')
res=[]
fn_rg=np.asarray(fn_rg)
for fr in fn_rr:
    rg_name=fr[0:10]+'.csv'
    rr_name=fr
    print('***50)
    print('+++',rg_name,rr_name)

#ถ้าพบไฟล์เดкар์ในฝนสถานี
if len(np.where(rg_name==fn_rg)[0])>0:

    #เปิดไฟล์เดкар์
    rlayer = QgsRasterLayer(path_rr+rr_name, rr_name)

    #เปิดไฟล์สถานีราย15นาทีแบบ csv
    with open(path_rg+rg_name, 'r') as f:
        rg = list(csv.reader(f, delimiter=','))

    #ลูปสถานีจากลิสไฟล์สถานีที่สมบูรณ์พร้อมพิกัดที่แปลงเป็น UTM แล้ว
    #เบริยบเทียบฝนสถานีกับไฟล์เดкар์ต่อละช่วงเวลา
    rg=np.asarray(rg)
    for s in s_utm:
        st,x_utm,y_utm=s

        #สกัดค่าฝนสถานีจาก rg ในสถานีนั้นๆ
        idg=np.where(rg==st)[0][0] #หาอินเด็กซ์แถวของฝนสถานีที่สมบูรณ์ที่มีรหัสสถานีตรงกับ s
        gauge=rg[idg][2]

        #สกัดค่าไฟล์เดкар์จาก rlayer ในสถานีนั้นๆ
        radar, result = rlayer.dataProvider().sample(QgsPointXY(x_utm,y_utm), 1) #radar คือค่าฝน
        เดкар์ที่รีเซิร์ฟกลับมาจากการสกัดตัวอย่าง
        print(st,"{:2f} , {:.2f}".format(float(gauge),float(radar)))

```

```
#เก็บผลลัพธ์การสกัด rr และ rg ของแต่ละสถานีไว้ในลิส res
res.append((st,"{:.2f}".format(float(gauge)), "{:.2f}".format(float(radar)))))

print('***50)
# break

#เชฟไฟล์ผลลัพธ์การ validate ออกໄປเป็น csv เพื่อคำนวนค่าสถิติ
fileOutput='validate_rr_rg_24h.csv'
saveValidate(res,path_val,fileOutput)

print('สิ้นสุดการ validate.....')
```





ภาคผนวก ฉบับ

โค้ด PyQGIS ตรวจสอบฟันประมวลค่าจากค่าการสะท้อนเรเดาร์ด้วยฟันที่วัด
ได้จากภาคพื้นดินราย 24 ชั่วโมง

ไฟล์ที่ใช้ว่า "14Validation_statistics_by_each_stations24h_marshallPalmer.py"
20200617 หาค่าสถิติ validation รายสถานี ราย24h MarshallPalmer
หาค่าสถิติ validation รายสถานี ราย24h ปรับปูงโดยตัดจาก 118 โดยข้อมูลจากไฟล์115
เพื่อนำไปวิเคราะห์และแสดงผลในรูปแผนที่

อ่านสถิติที่ใช้ในการหาbias http://scienceasia.org/2012.38.n4/scias38_373.pdf
Bias correction of radar rainfall estimates based on ageostatistical technique

```
import csv
import numpy as np
from scipy import stats

gaugeName='D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/1data_org/2gauge/' #เป็นไฟล์เดอร์
ของไฟล์ของสถานีพร้อมพิกัด

#ไฟล์เดอร์เก็บค่าการถักด้ rr+rg และเก็บผลลัพธ์การคำนวนค่าสถิติ validateation
path_val='D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/3data_processed/5outp_validations/iso
ntihn/MarshallPalmerStratiformZR/'

#อ่านสถานีวัดฝนพัฒนาในลุ่มน้ำ
with open(gaugeName+'1สถานีวัดฝนกรมอุตุลุ่มน้ำยม2018.csv', 'r') as f:
    st_all = list(csv.reader(f, delimiter=','))

#อ่านผลลัพธ์ถักด้ข้อมูลrr+rg
with open(path_val+'validate_rr_rg_24h.csv', 'r') as f:
    val= list(csv.reader(f, delimiter=','))

#แปลงลิสเป็นอาเรย์
r=np.asarray(val)
r = r.astype(np.float) #แปลงstring เป็น float เพื่อกำหนด

#กรองเอา rg>0.5 & rr>0.5
rr=r[:,1]>0.5 ]
```

```
r=rr[rr[:,2]>0.5]
```

```
def transformGeo2UTM(lon,lat):
    #แปลงพิกัด geo เป็น utm47N
    geom = QgsGeometry(QgsPoint(float(lon),float(lat))) # นำพิกัดสถานีผ่านมาสร้างเป็น geometry
    sourceCrs = QgsCoordinateReferenceSystem(4326)
    destCrs = QgsCoordinateReferenceSystem(32647)
    tr = QgsCoordinateTransform(sourceCrs, destCrs, QgsProject.instance())
    geom.transform(tr) #แปลง geo เป็น utm
    x_utm,y_utm=(geom.asPoint()[0],geom.asPoint()[1]) #get ค่า utm
    return x_utm,y_utm

def statistics(r):
    """
    สถิติ validation เปรียบเทียบระหว่าง gauge และ การประมาณค่าฝนด้วย radar
    """
    yhat=r[:,2] #estimator=rr
    Y=r[:,1] #expected=rg

    #SSE:sum of squared errors (or MSE:mean square error)
    sse=np.mean((np.mean(yhat) - Y) ** 2)

    #Variance
    var = np.var(yhat)

    #Bias of estimator
    bias = sse - var

    #ค่าสถิติ regression line
    slope, intercept, r_value, p_value, std_err = stats.linregress(yhat,Y)

    #MFB mean field bias
    mfb=np.sum(Y)/np.sum(yhat)

    #MAE
```

```

mae=np.sum(yhat-Y)/len(yhat)

#NSE The Nash-Sutcliffe-Efficiency (NSE, Nash and Sutcliffe, 1970)
nse= 1-(np.sum((yhat-Y)**2)/np.sum((Y-np.mean(Y))**2))

#Relative error(RE)
ree=np.sum(yhat-Y)/np.sum(Y)

#RMSE
rmse=np.sqrt((np.sum((yhat-Y)**2))/len(yhat))

return sse,var,bias,r_value,r_value**2,mfb,mae,nse,ree,rmse

def saveValidate(res,path_val,fileOutput):
    # เชื่อมผลลัพธ์การ validate เป็น csv
    val_res = open(path_val+fileOutput, 'w',newline='') #newline="" เพื่อป้องกันการเพิ่มบรรทัดตอนเชื่อม
    with val_res:
        w = csv.writer(val_res)
        w.writerows(res)

    # หารายชื่อสถานีฝนภาคพื้นดิน
    st_all=np.asarray(st_all)
    st=set(r[:,0]) # หารายชื่อสถานีแบบ unique
    res=[]

    res.append(("code_st","x_utm","y_utm","sse","var","bias","r_value","r_square","mfb","mae","nse","re","rms
e"))

    for s in st:
        print('***50')
        print('+++สถานี',s)
        ids=np.where(st_all==str(int(s)))[0][0] # หาอินเด็กซ์แล้วที่มีรหัสสถานีตรงกับ s
        lon,lat=(st_all[ids][1],st_all[ids][2]) #get ค่าพิกัดสถานีของ s จากไฟล์สถานีหลัก
        x_utm,y_utm=transformGeo2UTM(lon,lat)

        # break

```

```

id=np.where(r[:,0]==s) #หาว่าสถานี s มีชื่ออยู่ใน r ไหนบ้าง
re=r[id[0]] #ตั้งชื่อผลลัพธ์ของสถานี s

#คำนวนค่า validation statistics
sse,var,bias,r_value,r_square,mfb,mae,nse,ree,rmse=statistics(re)
res.append((int(s),x_utm,y_utm,sse,var,bias,r_value,r_square,mfb,mae,nse,ree,rmse))

print('+++ค่าสถิติตรวจสอบความถูกต้องของการโมเดล+++')
print("Residual sum of squares: {:.2f}".format(sse))
print("Variance: {:.2f}".format(var))
print("Bias: {:.2f}".format(bias))
print("correlation coefficient: {:.2f}".format(r_value))
print("coefficient of determination (r_squared): {:.2f}".format(r_value**2))
print("Mean Field Bias: {:.2f}".format(mfb))
print("Mean Absolute Error:{:.2f}".format(mae))
print("NSE:{:.2f}".format(nse))
print("RE:{:.2f}".format(ree))
print("RMSE: {:.2f}".format(rmse))
print("***50)

# break

#exportไฟล์ผลลัพธ์การ validate ออกเป็น csv เพื่อใช้ pyqgis พลิกด้วย命令ที่
fileOutput='validate_rr_rg_24h_each_stations.csv'
saveValidate(res,path_val,fileOutput)

```



ภาคผนวก ช

โค้ด PyQGIS การคำนวณค่าสถิติการตรวจสอบฟันประมาณค่าจากการ
สะท้อนเรดาร์ด้วยฟันที่วัดได้จากภาคพื้นดินราย 24 ชั่วโมง



""
ไฟล์ชื่อว่า "44Overall_statistics_validate_date_radar_gauge24h_all_files_marshallPalmer.py"
20200619 คำนวนค่าสถิติ validation ระหว่างฝนสถานีกับฝนเดาร์ราย24h MarshallPalmer ราย24h

ต่อไปจะคำนวนค่าสถิติแต่ละสถานี เพื่อนำไปแสดง spatial bias และผลลัพธ์กราฟดูค่าสถิติรายวัน

อ่าน http://scienceasia.org/2012.38.n4/scias38_373.pdf
Bias correction of radar rainfall estimates based on geostatistical technique

""
import csv
import numpy as np
from scipy import stats

print('.....เริ่มการคำนวนค่าสถิติ validation ระหว่างฝนสถานีกับฝนเดาร์ราย24h...')

def statZR(r):
 ""
 คำนวนสถิติในแต่ละวัน
 ""
 yhat=r[:,3] #estimator=rr
 Y=r[:,2] #expected=rg

 #SSE:sum of squared errors (or MSE:mean square error)
 sse=np.mean((np.mean(yhat) - Y) ** 2)

 #Variance
 var = np.var(yhat)

 #Bias of estimator
 bias = sse - var

 #ค่าสถิติ regression line
 slope, intercept, r_value, p_value, std_err = stats.linregress(yhat,Y)

```

#MFB mean field bias
mfb=np.sum(Y)/np.sum(yhat)

#Relative error(RE)
re=np.sum(yhat-Y)/np.sum(Y)

#NSE The Nash-Sutcliffe-Efficiency (NSE, Nash and Sutcliffe, 1970)
nse= 1-(np.sum((yhat-Y)**2)/np.sum((Y-np.mean(Y))**2))

#RMSE
rmse=np.sqrt((np.sum((yhat-Y)**2))/len(yhat))

return sse,var,bias,r_value,mfb,re,nse,rmse

def saveValidate(res,path_val,fileOutput):
    #เชฟผลลัพธ์การ validate เป็น csv
    val_res = open(path_val+fileOutput, 'w',newline='') #newline="" เพื่อป้องกันการเพิ่มบรรทัดตอนเชฟ
    CSV
    with val_res:
        w = csv.writer(val_res)
        w.writerows(res)

    #-----
    path_val='D:/Yang/1Reseach/0.2563.RadarMosGISYomNan/3data_processed/5outp_validations/1so
    ntihn/MarshallPalmerStratiformZR/' #ไฟล์เดอร์เก็บผลลัพธ์สกัดข้อมูลrg+rr

    #อ่านผลลัพธ์สกัดข้อมูลrr+rg
    with open(path_val+'validate_rr_rg_date_24h.csv', 'r') as f:
        val= list(csv.reader(f, delimiter=','))

    #แปลงลิสเป็นอาเรย์
    r=np.asarray(val)
    r = r.astype(np.float) #แปลงstring เป็น float เพื่อการคำนวน

    #กรองเฉพาะrg>0.5 & rr>0.5

```

```

rr=r[:,2]>0.5 ]
r=rr[rr[:,3]>0.5]

dd=set(r[:,0])
res=[]
for d in dd: #ค้นหาฝนในแต่ละวัน
    print('#'*50)
    print('---วันที่:',d)

    id=np.where(d==r)
    rd=r[id[0]]
    #ถ้าพบไฟล์เดียวของวันที่กำลังค้นหา
    if len(rd)>0:
        sse,var,bias,r_value,mfb,re,nse,rmse=statZR(rd)
        res.append((int(d),sse,var,bias,r_value,mfb,re,nse,rmse))
        #ค่าสถิติการตรวจสอบรวม

        print('+++'+ค่าสถิติตรวจนับความถูกต้องของการไม่เสีย+++')
        print("Residual sum of squares: {:.2f}".format(sse))
        print("Variance: {:.2f}".format(var))
        print("Bias: {:.2f}".format(bias))
        print("correlation coefficient: {:.2f}".format(r_value))
        print("coefficient of determination (r_squared): {:.2f}".format(r_value**2))
        print("Mean Field Bias: {:.2f}".format(mfb))
        print("Relative Error: {:.2f}".format(re))
        print("NSE: {:.2f}".format(nse))
        print("RMSE: {:.2f}".format(rmse))
        print('#'*50)

#เข้าไฟล์ผลลัพธ์การ validate รายวัน
fileOutput='validate_rr_rg_date_24h_all_stations.csv'
saveValidate(res,path_val,fileOutput)

print('.....เสร็จสิ้นการคำนวนค่าสถิติ validation ระหว่างฝนสถานีกับฝนเดาร์ราย24h...')

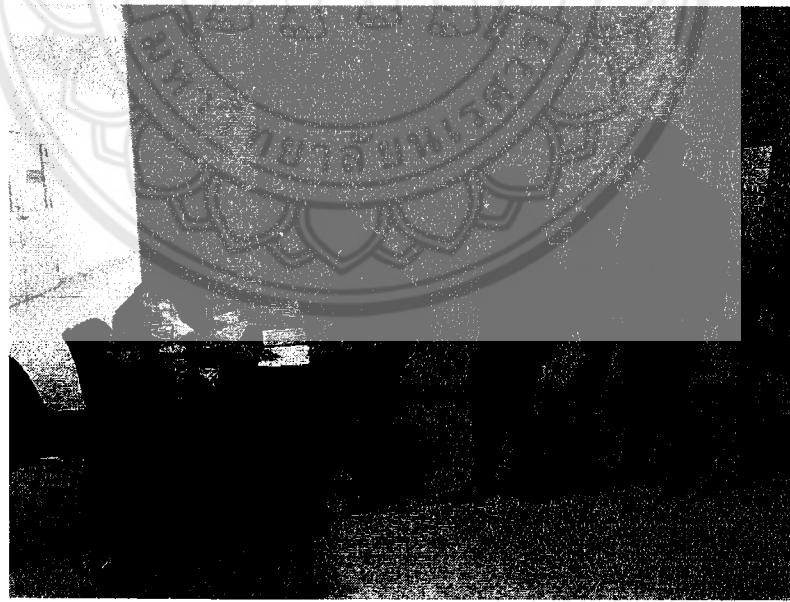
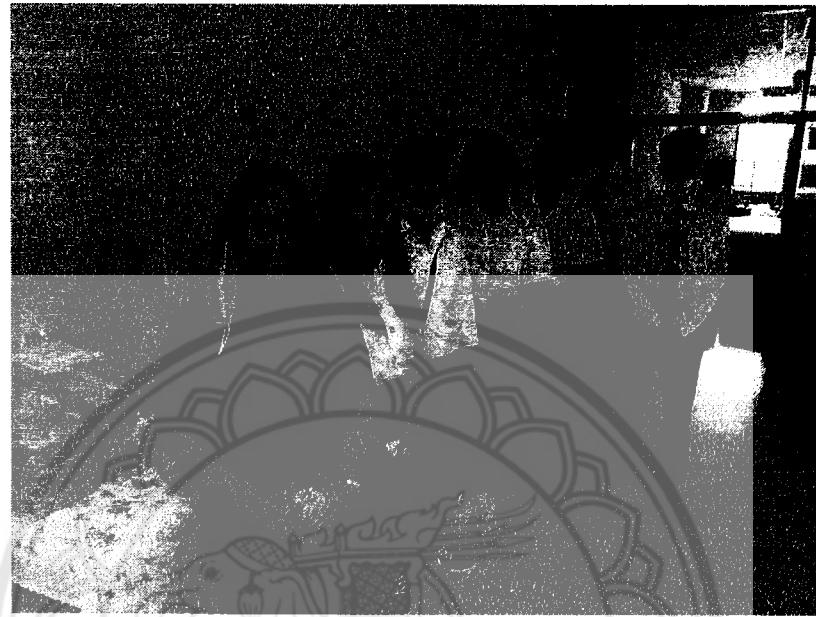

```

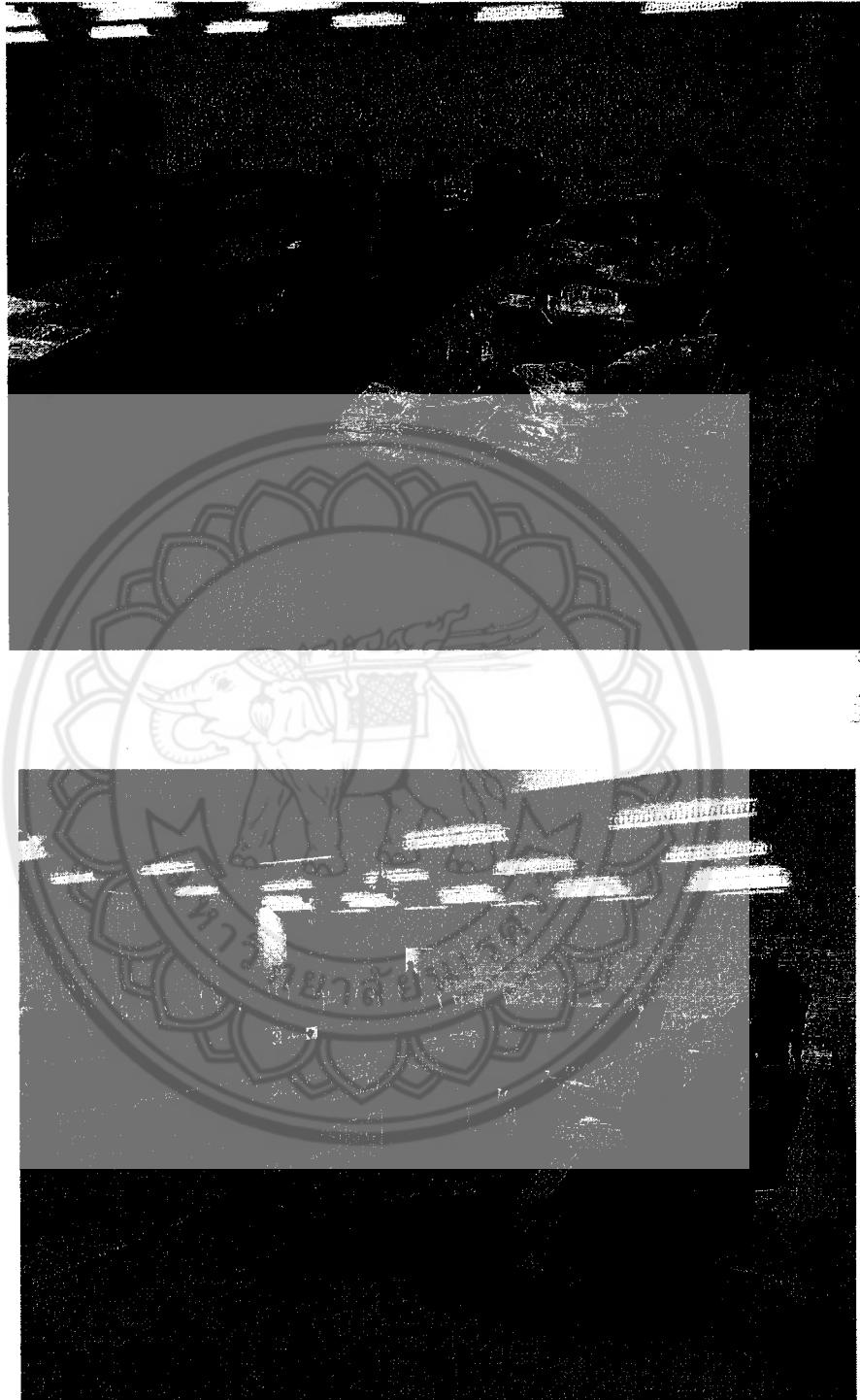
รายละเอียดการจัดงานอบรมเผยแพร่องค์ความรู้งานวิจัย

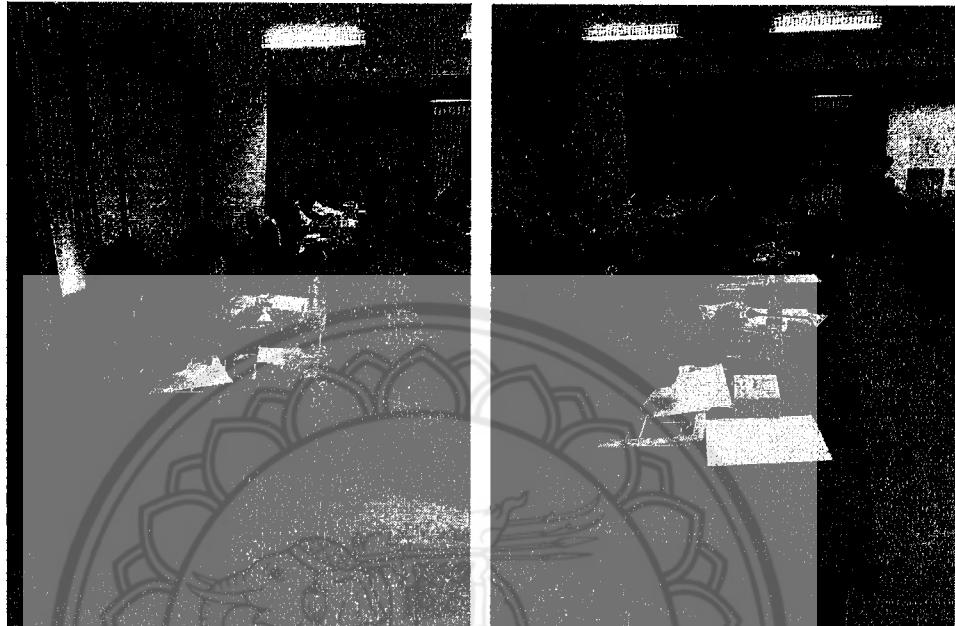














บันทึกข้อความ

ส่วนราชการ คณะเกษตรศาสตรฯ ภาควิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม โทร. 2757

ที่ อว 0603.07.01/

วันที่ กรกฏาคม 2563

เรื่อง ขออนุมัติจัดโครงการอบรมปิดโครงการวิจัยงบแผ่นดินรหัสโครงการ R2562B031

เรียน คณบดีคณะเกษตรศาสตรฯ

ตามที่มหาวิทยาลัยนเรศวรได้อนุมัติจัดสรรงบประมาณแผ่นดิน ประจำปีงบประมาณ 2562 ให้แก่สัญญาเลขที่ R2562B031 โครงการวิจัยเรื่อง “การวิเคราะห์รูปแบบบนพื้นฐานข้อมูลภูมิสารสนเทศตรวจสอบจากสภาพพื้นดินรายละเอียดสูงในลุ่มน้ำยมและน่าน” โดยมี ผศ.ดร.นรรพล มหาวิศ ผู้อำนวยการวิชาชีวทรัพยากรธรรมชาติและสิ่งแวดล้อม เป็นหัวหน้าโครงการนั้น บัดนี้โครงการได้บรรลุวัตถุประสงค์ที่ตั้งไว้แล้ว

ดังนั้น ข้าพเจ้า ผศ.ดร.นรรพล มหาวิศ จึงโปรดอนุมัติโครงการอบรมเผยแพร่องค์ความรู้ด้านอุดมวิทยาและการใช้ข้อมูลเดิร์ตตรวจสอบภาคในทางเกษตรกรรม ในวันศุกร์ที่ 31 กรกฎาคม 2563 ตั้งแต่เวลา 09.00 – 16.00 น. ณ ห้อง AG2203 คณะเกษตรศาสตรฯ มหาวิทยาลัยนเรศวร โดยขอเบิกค่าใช้จ่ายดังนี้

| | |
|--|------------|
| 1. ค่าอาหารผู้เข้าร่วมโครงการ (200บาทx30 คน) | 6,000 บาท |
| 2. ค่าอาหารว่าง (100บาทx2มื้อx30คน) | 6,000 บาท |
| 3. ค่าวัสดุสำนักงาน | 6,400 บาท |
| 4. ค่าวิทยากร (600บาทx2คนx3ชั่วโมง) | 3,600 บาท |
| รวมเป็นเงิน | 20,000 บาท |

หมายเหตุ ถ้าจ่ายได้ทุกรายการ

ทั้งนี้โดยเบิกจ่ายจากงบประมาณแผ่นดิน ประจำปีงบประมาณ 2562 โครงการวิเคราะห์รูปแบบบนพื้นฐานข้อมูลภูมิสารสนเทศตรวจสอบจากสภาพพื้นดินรายละเอียดสูงในลุ่มน้ำยมและน่าน

จึงเรียนมาเพื่อโปรดพิจารณาอนุมัติ

(ผศ.ดร.นรรพล มหาวิศ)

หัวหน้าโครงการฯ

โครงการอบรมเผยแพร่องค์ความรู้ด้านอุตุนิยมวิทยาและเรดาร์ตรวจสอบอากาศในทางเกษตรกรรม

สาขาวิชาภูมิศาสตร์ ภาควิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม

คณะเกษตรศาสตร์ ทรัพยากรธรรมชาติและสิ่งแวดล้อม มหาวิทยาลัยนเรศวร

ชื่อโครงการ :

อบรมเผยแพร่องค์ความรู้ด้านอุตุนิยมวิทยาและเรดาร์ตรวจสอบอากาศในทางเกษตรกรรม

หลักการและเหตุผล

ตามที่มหาวิทยาลัยนเรศวรได้อนุมัติจัดสรรงบประมาณแผ่นดิน ประจำปีงบประมาณ ประจำปีงบประมาณ 2562 ให้แก่สัญญาเลขที่ R2562B031 โครงการวิจัยเรื่อง “การวิเคราะห์รูปแบบพื้นฐานข้อมูลภูมิสารสนเทศตรวจสอบจากภาคพื้นดินรายละเอียดสูงในคุณน้ำยมและน่าน” โดยมี ผศ. ดร.นัฐพล มหาวิศว์ สังกัดสาขาวิชาภูมิศาสตร์ ภาควิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม เป็นหัวหน้าโครงการนั้น บัดนี้โครงการได้นำรุกวัตถุประสงค์ที่ตั้งไว้ในสัญญาแล้ว เพื่อให้มีการเข้าใจและทราบถึง องค์ความรู้ทางด้านอุตุนิยมวิทยาและการใช้ข้อมูลดาวเทียมในการวิเคราะห์สภาพอากาศ คณะผู้วิจัยจึงได้ จัดการอบรมเพื่อเผยแพร่องค์ความรู้ในชื่อโครงการอบรมเผยแพร่องค์ความรู้ด้านอุตุนิยมวิทยาและเรดาร์ ตรวจสอบอากาศในทางเกษตรกรรม โดยเชิญผู้เชี่ยวชาญจากการมอุตุนิยมวิทยามาเป็นวิทยากรในการบรรยาย ให้กับผู้เข้าประชุม

วัตถุประสงค์

เพื่อเผยแพร่องค์ความรู้ด้านอุตุนิยมวิทยาและเรดาร์ตรวจสอบอากาศในทางเกษตรกรรม

รูปแบบการจัดประชุม

บรรยายองค์ความรู้ทางทฤษฎีและอภิปราย รวมถึงแลกเปลี่ยนประสบการณ์ระหว่างวิทยากร หน่วยงานและผู้ที่สนใจที่เข้าร่วมอบรม

ผู้เข้าประชุม

เจ้าหน้าที่หน่วยงานที่เกี่ยวข้องจากภาครัฐในพื้นที่จังหวัดพิษณุโลกและสุโขทัยและนิสิตที่สนใจ จำนวน 30 คน

ผลผลิตและตัวชี้วัดความสำเร็จ

ผลผลิต

ผู้เข้าประชุมมีความเข้าใจด้านอุตุนิยมวิทยาและการใช้ข้อมูลเรดาร์ในการวิเคราะห์สภาพอากาศ ด้านเกษตรกรรม

ตัวชี้วัดความสำเร็จ

(1) ร้อยละของผู้เข้าประชุม : ไม่น้อยกว่าร้อยละ 80

(2) ร้อยละของผู้เข้าประชุมที่ผ่านการประเมิน : ไม่น้อยกว่าร้อยละ 80

ประโยชน์ที่คาดว่าจะได้รับ

(1) ผู้เข้าประชุมเข้าใจด้านอุตุนิยมวิทยาการเกษตร

(2) ผู้เข้าประชุมเข้าใจสภาพอากาศผ่านกรณีตัวอย่างในการใช้ข้อมูลเรดาร์เพื่อเฝ้าระวังด้าน การเกษตร

หน่วยงานและผู้รับผิดชอบโครงการ

ภาควิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม

คณะเกษตรศาสตร์ ทรัพยากรธรรมชาติและสิ่งแวดล้อม มหาวิทยาลัยนเรศวร

ผศ.ดร.นัฐพล มหาวิค

ระยะเวลาในการจัดอบรม

วันที่ 31 กรกฎาคม 2563

การฝึกอบรม

การเผยแพร่องค์ความรู้ด้านอุตุนิยมวิทยาและเรดาร์ตรวจสอบภาคในทางเกษตรกรรม

สาขาวิชาภูมิศาสตร์ ภาควิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม

คณะเกษตรศาสตร์ฯ มหาวิทยาลัยนเรศวร

วันศุกร์ที่ 31 กรกฎาคม 2563

08.30-09.00 น. ลงทะเบียน (ณ ห้อง AG2203 คณะเกษตรศาสตร์ฯ มหาวิทยาลัยนเรศวร)

09.00-10.45 น. การบรรยายเรื่อง การตรวจสอบภาคด้วยเรดาร์ตรวจสอบภาคพื้นดิน และการประยุกต์ใช้ในการเฝ้าระวังทางด้านเกษตรกรรม

วิทยากร: คุณนนท์ธรวช เนื่องจำนง ผู้เชี่ยวชาญเรดาร์ตรวจสอบภาคสถานี อุตุนิยมวิทยาจังหวัดพิษณุโลก

- หลักการทฤษฎีเรดาร์ตรวจสอบภาค
- ผลิตภัณฑ์จากข้อมูลเรดาร์
- หลักการพยากรณ์เพื่อการเตือนภัยด้วยเรดาร์ตรวจสอบภาค
- ตัวอย่างกรณีศึกษางานภาคปฏิบัติที่เกี่ยวข้องทางด้านเกษตรกรรม

10.45-11.00 น. พักรับประทานอาหารว่าง

11.00-12.00 น. การบรรยายเรื่อง การตรวจสอบภาคด้วยเรดาร์ตรวจสอบภาคพื้นดิน

วิทยากร: คุณนนท์ธรวช เนื่องจำนง

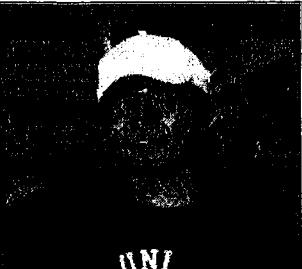
- การปฏิบัติงานและกรณีศึกษาตัวอย่างภาคเหนือตอนล่างในด้าน การเกษตรกรรม
- ภารกิจเฝ้าระวังเตือนภัย ภัยแล้ง ฝนจางร่องมรสุม ฝนจากพายุ พายุฤดูร้อน ลูกเห็บ พื้นที่ภาคเหนือตอนล่าง
- ปัญหาและโจทย์งานวิจัยที่ต้องการพัฒนา (ถาม-ตอบอภิปราย)

12.00-13.00 น. พักรับประทานอาหารกลางวัน

| | |
|----------------------------|--|
| 13.00-14.30 น. | การบรรยายเรื่อง หลักการอุตุนิยมวิทยาและการปฏิบัติงานในด้านพยากรณ์อากาศทางด้านเกษตรกรรม วิทยากร: อาจารย์ประพุติ ยอดไพบูลย์ ผู้อำนวยการสถานีอุตุนิยมวิทยาจังหวัดพิษณุโลก <ul style="list-style-type: none">■ หลักการและการเก็บข้อมูลทางอุตุนิยมวิทยา■ เครื่องมือทางอุตุนิยมวิทยา■ หลักการพยากรณ์เพื่อการเตือนภัยในงานอุตุนิยมวิทยา■ ตัวอย่างกรณีศึกษางานภาคปฏิบัติที่เกี่ยวกับด้านการเกษตร |
| 14.30-14.45 น. | พักรับประทานอาหารว่าง |
| 14.45-16.00 น. การเกษตร | การบรรยายเรื่อง ลักษณะสภาพอากาศภาคเหนือตอนล่างที่เกี่ยวกับด้าน วิทยากร: อาจารย์ประพุติ ยอดไพบูลย์ <ul style="list-style-type: none">■ การปฏิบัติงานและกรณีศึกษาตัวอย่างภาคเหนือตอนล่างที่เกี่ยวกับด้านการเกษตร■ ภารกิจเฝ้าระวัง เตือนภัยและพยากรณ์ไฟป่า ภัยแล้ง ฝนจารจร มองร่อง มองสุม ฝนจากพายุ พายุฤดูร้อน ลูกเห็บ พื้นที่ภาคเหนือตอนล่างที่เกี่ยวกับด้านการเกษตร■ ปัญหาและโจทย์งานวิจัยที่ต้องการพัฒนา (ถาม-ตอบอภิปราย) |
| 16.00-16.30 น. | สรุปผลการฝึกอบรมและมอบประกาศนียบัตร |



ประวัติและผลงานทางวิชาการ [update APRIL 28 , 2020]

| | |
|---|--|
|  | Asst. Prof. Dr. Nattapon Mahavik (ผศ.ดร.นัฐพล มหาวิค) |
| UNI | <ul style="list-style-type: none"> ■ D.Sc. in Atmospheric Science, Kyoto University, 2015 , Japan [MEXT] ทุนรัฐบาลญี่ปุ่น ■ M.Sc. in Photogrammetry and Geoinformatics, Stuttgart University of Applied Sciences, 2009, Germany [DAAD] ทุนรัฐบาลเยอรมัน ■ วทบ. ภูมิศาสตร์ (เกียรตินิยม), มหาวิทยาลัยเชียงใหม่, 2545 |
| Contact | +66(55)962752, nataponm@nu.ac.th |
| Research interests | <ul style="list-style-type: none"> ■ Ground-based radar rainfall estimation ■ Remote sensing in Urban Heat Island ■ Rainfall estimation from satellite products |

Awards and Honour

- 2018 Shunji Murai Award for best paper on Asian Conference on Remote Sensing 2018 for the title *“The convective cloud properties extraction from weather radar reflectivity during SONCA tropical storm over the Lower Northern Thailand”*
- 2010-2013 Japanese government scholarship (monbukagakusho: MEXT) ทุนนักศึกษาวิจัยปริญญาเอก
- 2007-2009 German Academic Exchange Service (DAAD) ทุนปริญญาโท
- 1999-2001 The Shell Company of Thailand Limited

Work experiences

| Date | Position | Organization |
|-----------------------|--|--|
| 25 May 2009 - Present | Lecturer in Geography and Geoinformatics | Department of Natural Resources and Environment, Naresuan University |

Specialization skills

- Ground-based weather radar and Satellite rainfall product processing
- Fortran, Python Scripting Language, GrADS Scripting Language, NCL Scripting Language, R Scripting Language
- Linux systems (Centos, Ubuntu)
- ArcGIS, QGIS, ENVI, ERDAS IMAGINE, MATLAB
- Drone photogrammetry (DJI, Syma) for mapping

ผลงานทางวิชาการ

บทความวิจัย/บทความวิชาการที่ตีพิมพ์
ระดับนานาชาติ

1. Mahavik, N., Tantanee, S., (2020). Precipitating clouds analysis based on the developed radar mosaic products over the Chao Phraya River basin, *International Journal of Geoinformatics*. (in press, Scopus).
2. Mahavik, N., Tantanee, S., (2020). Developed Radar Quality Index of Mosaicked Weather Radars over The Chao Phraya River Basin, Thailand. (Submitted in Scopus)
3. Mahavik, N., (2020). Seasonal characteristics of precipitating cloud properties and structure derived from TRMM: 16 years over the inland of the Indochina Peninsula (Submitted in Scopus).
4. Mahavik, N., Tantanee, S., (2019). Convective systems observed by ground-based radar during seasonal march of Asian summer monsoon in the middle of Indochina Peninsula, *Engineering and applied science research*, 46(4), 318-330. (Scopus).
5. Mahavik, N., Tantanee, S., (2019). Spatio-Temporal Analysis of Convective Cloud Properties Deriving from Weather Radar Reflectivity during the Decaying Stage of Tropical Storm over the Lower Northern Thailand, *International Journal of Geoinformatics*, 15(3), 91-102. (Scopus).
6. Seejata, K., Yodying, A., Wongthadom, T., Mahavik, N., Tantanee, S., (2018). Assessment of flood hazard areas using Analytical Hierarchy Process over the Lower Yom Basin, Sukhothai Province. *Procedia Engineering*, 212, 340-347. (Scopus)
7. Mahavik, N., (2017). Spatial Seasonal Distribution of Climatological Precipitation over the Middle of the Indochina Peninsula. *Applied Environmental Research*, 39(3), 63-76.
8. Mahavik, N., (2017). Extreme precipitation trends over the middle of Indochina Peninsula during the period from 1978-2007. *International Journal of Earth Sciences and Engineering*, 10(3), 595-603.
9. Mahavik, N., (2017). Bias Adjustments of Radar Rainfall during Seasonal March of the Summer Monsoon in the Middle of Thailand. *International Journal of Applied Environmental Sciences*, 12(4), 577-594.
10. Mahavik, N., T. Satomura, S. Shige, B. Sysouphanthavong, S. Phonevilay, M. Wakabayashi, S. Baimoung (2014). Rainfall Pattern over the Middle of Indochina Peninsula during 2009–2010 Summer Monsoon. *Hydrological Research Letter*, 8, 57-63.
11. Mahavik, N., T. Satomura, S. Baimoung (2013). Radar rainfall analysis in the middle of Indochina peninsula. *J. Disaster Res*, 8, 187-188. (Scopus)

ตีพิมพ์ในรายงานสืบเนื่องจากการประชุมวิชาการ (Proceedings) ระดับนานาชาติ

1. Khaing, T.W., S. Tantanee, W. Pratoomchai, N. Mahavik, 2019: *Coupling Flood Hazard with Vulnerability Map for Flood Risk Assessment: A Case Study of Nyaung-U Township in Myanmar*, GMSARN 2019, Env-08.
2. Tobgay, S., Mahavik, N., 2019: Identifying habitats of red panda (*AILURUS F. FULGENS*) in Sakteng wildlife sanctuary, Bhutan in Maxent using climate change scenarios. *Proceeding of Asian Conference on Remote Sensing 2019*, MoB1-5.
3. Yodying, A., Seejata, K., Chatsudarat, S., Chidburee, P., Mahavik, N., Kongmuang, C., Tantanee, S. 2019: Flood hazard assessment using Fuzzy Analytic Hierarchy Process: a case study of Bang Rakam model in Thailand. *Proceeding of Asian Conference on Remote Sensing 2019*, TuD2-4.
4. Seejata, K., Yodying, A., Chatsudarat, S., Chidburee, P., Mahavik, N., Kongmuang, C., Tantanee, S. 2019: Assessment of flood hazard using geospatial data and Frequency Ratio model in Sukhothai province, Thailand. *Proceeding of Asian Conference on Remote Sensing 2019*, MoD1-4.
5. Chatsudarat, S., Yodying, A., Seejata, K., Chidburee, P., Mahavik, N., Kongmuang, C., Tantanee, S. 2019: The Study of landslide susceptibility using GIS based on rainfall deriving from TRMM over Northern Thailand. *Proceeding of Asian Conference on Remote Sensing 2019*, TuD2-4, MoD2-7.
6. Mahavik, N., Tantanee, S. 2018: Quality Assessment of mosaicked weather radars over the Chao Phraya river basin, Thailand. *Proceeding of GMSARN Int. Conf. on Energy, Environment, and Development in GMS*, SD44, 1-4.
7. Mahavik, N., Tantanee, S. 2018: The convective cloud properties extraction from weather radar reflectivity during SONCA tropical storm over the Lower Northern Thailand. *Proceeding of Asian Conference on Remote Sensing 2018*, 3, 1604-1612. (Shunji Murai Award)
8. Mahavik, N., Tantanee, S. 2018: Comparison of spatial error structures between TRMM products and APHRODITE over the Indochina Peninsula. *Proceeding of Asian Conference on Remote Sensing 2018*, 3, 1754-1763.
9. Mahavik, N., T. Satomura, B. Sysouphanthavong, S. Phonevilay, M. Wakabayashi, and S. Baimoung, 2013: Rainfall pattern observed by radar calibrated by rain gauges in Indochina. *Proceeding of The Third International MAHASRI/HyARC Workshop on Asian Monsoon and Water Cycle*, 44-49.

ตีพิมพ์ในรายงานสืบเนื่องจากการประชุมวิชาการ (Proceedings) ระดับชาติ

1. อุไรรัตน์ อุ่นเมือง, ศศิธร ฉัตรสุดารัตน์, กมลฉัตร ศรีจะตะ, อภิษฐาน ยอดยิ่ง, นภาพร สนองบุญ, จรัส ดาว คงเมือง, ศรินทร์ทิพย์ แทนранานี, น้ำฝน มหาวิค. (2562). การพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไ米ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจาก Google และอากาศยานไร้คนขับ. เรื่องเต็มการประชุมวิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์และสิ่งแวดล้อม ครั้งที่ 4, 12-13 ธันวาคม 2562, มหาวิทยาลัยพะเยา, หน้า 279-288.
2. ณัฐร์ษา สุยะตา, อุไรรัตน์ อุ่นเมือง, ศศิธร ฉัตรสุดารัตน์, กมลฉัตร ศรีจะตะ, อภิษฐาน ยอดยิ่ง, จรัส ดาว คงเมือง, นภาพร สนองบุญ, น้ำฝน มหาวิค. (2562). การคาดการณ์การเปลี่ยนแปลงพื้นที่ปลูกข้าวโพดที่ส่งผลกระทบต่อทัศนคติชุมชนด้วยแบบจำลอง MOLUSCE : กรณีศึกษา อาเภอปัว จังหวัดน่าน. เรื่องเต็มการประชุมวิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์และสิ่งแวดล้อม ครั้งที่ 4, 12-13 ธันวาคม 2562, มหาวิทยาลัยพะเยา, หน้า 289-298.
3. พิจิตรา พะยอม, อุไรรัตน์ อุ่นเมือง, ศศิธร ฉัตรสุดารัตน์, กมลฉัตร ศรีจะตะ, อภิษฐาน ยอดยิ่ง, จรัสดาว คงเมือง, นภาพร สนองบุญ, น้ำฝน มหาวิค. (2562). การเปรียบเทียบการกระจายตัวของจุดความร้อนที่ตรวจวัดได้ระหว่าง MODIS C6 กับ VIIRS ในพื้นที่อำเภอลานสัก จังหวัดอุทัยธานี. เรื่องเต็มการประชุมวิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์และสิ่งแวดล้อม ครั้งที่ 4, 12-13 ธันวาคม 2562, มหาวิทยาลัยพะเยา, หน้า 450-458.
4. เมทีนี เมืองมา, อุไรรัตน์ อุ่นเมือง, ศศิธร ฉัตรสุดารัตน์, กมลฉัตร ศรีจะตะ, อภิษฐาน ยอดยิ่ง, จรัสดาว คงเมือง, นภาพร สนองบุญ, น้ำฝน มหาวิค. (2562). การทดสอบดัชนีสกัดพื้นที่น้ำผิวดินจากข้อมูลภาพดาวเทียมและพื้นที่น้ำท่วมน้ำจากในบริเวณลุ่มน้ำยมตอนล่าง. เรื่องเต็มการประชุมวิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์และสิ่งแวดล้อม ครั้งที่ 4, 12-13 ธันวาคม 2562, มหาวิทยาลัยพะเยา, หน้า 428-436.
5. ธัญญารัตน์ สืตื้อ, อุไรรัตน์ อุ่นเมือง, ศศิธร ฉัตรสุดารัตน์, กมลฉัตร ศรีจะตะ, อภิษฐาน ยอดยิ่ง, จรัส ดาว คงเมือง, นภาพร สนองบุญ, น้ำฝน มหาวิค. (2562). การทดสอบดัชนีชี้ภัยแล้งด้วยข้อมูลผลิตภัณฑ์ฝนดาวเทียมTRMM ในพื้นที่ภาคเหนือตอนล่าง. เรื่องเต็มการประชุมวิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์และสิ่งแวดล้อม ครั้งที่ 4, 12-13 ธันวาคม 2562, มหาวิทยาลัยพะเยา, หน้า 468-476.
6. ณปภัช ร่มรัตน์ไตร, อุไรรัตน์ อุ่นเมือง, ศศิธร ฉัตรสุดารัตน์, กมลฉัตร ศรีจะตะ, อภิษฐาน ยอดยิ่ง, จรัส ดาว คงเมือง, นภาพร สนองบุญ, น้ำฝน มหาวิค. (2562). การศึกษาปรากฏการณ์ภาวะความร้อนที่สัมพันธ์กับการขยายตัวของเมืองในพื้นที่จังหวัดเชียงใหม่. เรื่องเต็มการประชุมวิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์และสิ่งแวดล้อม ครั้งที่ 4, 12-13 ธันวาคม 2562, มหาวิทยาลัยพะเยา, หน้า 477-485.

7. ศีตลา บัวขาว น้ำโพล มหावิค และกัมปนาท ปิยะรำงษัย. (2561). ความถูกต้องของการจำแนกพื้นที่ป่าลูกอ้อระหว่างเทคนิคการจำแนกแบบละเอียดกว่าจุดภาพและการจำแนกแบบกำกับดูแล. การประชุมวิชาการนำเสนอผลงานวิจัยระดับชาติ เครือข่ายบันทึกศึกษามหาวิทยาลัยราชภัฏภาคเหนือ ครั้งที่ 18 และลำปางวิจัยครั้งที่ 4, 86-97.
8. มินตรา รองจันทร์และน้ำโพล มหावิค. (2560). การวิเคราะห์หาวันเริ่มต้นและวันสิ้นสุดของฤดูมรสุม ในพื้นที่ภาคเหนือของประเทศไทย. เรื่องเต็มการประชุมวิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์ และสิ่งแวดล้อมมนเรศวร ครั้งที่ 2, 15 ธันวาคม 2560, มหาวิทยาลัยนเรศวร, หน้า 170-179.
9. ศศิธร ฉัตรสุดารัตน์และน้ำโพล มหावิค. (2560). การศึกษาและวิเคราะห์การปลดปล่อยกําชเรือน กระเจในพื้นที่ป่าลูกอ้อ โดยเทคนิคการจำแนกเชิงวัดจากข้อมูลภาพถ่ายดาวเทียม Landsat-8 กรณีศึกษา อำเภอไทรโยง จังหวัดกำแพงเพชร. เรื่องเต็มการประชุมวิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์ และสิ่งแวดล้อมมนเรศวร ครั้งที่ 2, 15 ธันวาคม 2560, มหาวิทยาลัยนเรศวร, หน้า 72-82.
10. อミニตรา เจริมวัดและน้ำโพล มหावิค. (2560). การศึกษาและวิเคราะห์การตรวจพบไฟป่าของข้อมูล ดาวเทียมกรณีศึกษาอุทยานแห่งชาติในเขตจังหวัดแม่ย่องสอนและเชียงใหม่. เรื่องเต็มการประชุม วิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์ และสิ่งแวดล้อมมนเรศวร ครั้งที่ 2, 15 ธันวาคม 2560, มหาวิทยาลัยนเรศวร, หน้า 180-189.
11. ศีตลา บัวขาว น้ำโพล มหावิค และกัมปนาท ปิยะรำงษัย. (2560). การตรวจสอบความถูกต้องของ การจำแนกพื้นที่ป่าลูกอ้อ ด้วยวิธีการจำแนกแบบละเอียดกว่าจุดภาพ โดยใช้ข้อมูลดาวเทียมแลนด์ เชท 8. นเรศวรวิจัย ครั้งที่ 13, 138-147.
12. กมลนัตร ศรีจะตะและน้ำโพล มหावิค. (2559). การศึกษาความหนาแน่นของเมืองจากข้อมูล แบบจำลองความสูงไลดาร์ : กรณีศึกษาเขตดาวทวนและเขตสตริป รัฐเพนซิลเวเนีย ประเทศ สหรัฐอเมริกา. เรื่องเต็มการประชุมวิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์ และ สิ่งแวดล้อมมนเรศวร ครั้งที่ 1, 3 พฤศจิกายน 2559, มหาวิทยาลัยนเรศวร, หน้า 158-165.
13. เนตรนภา หงษ์ทองและน้ำโพล มหावิค. (2559). การศึกษาเปรียบเทียบเทคนิคการจำแนกเชิงจุดภาพ ของพื้นที่การเกษตรจากข้อมูลภาพถ่ายดาวเทียมไทยโซต : กรณีศึกษา อำเภอพร้าว จังหวัดเชียงใหม่. เรื่องเต็มการประชุมวิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์ และสิ่งแวดล้อมมนเรศวร ครั้งที่ 1, 3 พฤศจิกายน 2559, มหาวิทยาลัยนเรศวร, หน้า 92-98.
14. พิมพ์ผกา อ่องแก้วและน้ำโพล มหावิค. (2559). การศึกษาปฏิทินการเพาะปลูกพืชจากข้อมูลภาพ ดาวเทียมหลายช่วงเวลา : กรณีศึกษา จังหวัดอุตรดิตถ์. เรื่องเต็มการประชุมวิชาการ ทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์ และสิ่งแวดล้อมมนเรศวร ครั้งที่ 1, 3 พฤศจิกายน 2559, มหาวิทยาลัยนเรศวร, หน้า 85-91.
15. ทับทิม วงศ์ทະดำเนและน้ำโพล มหावิค. (2559). การวิเคราะห์พื้นที่เสี่ยงน้ำท่วมโดยกระบวนการจำดับ ชั้นเชิงวิเคราะห์ บริเวณลุ่มน้ำยมตอนล่าง จังหวัดสุโขทัย. เรื่องเต็มการประชุมวิชาการ

ทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์ และสิ่งแวดล้อมนเรศวร ครั้งที่ 1, 3 พฤศจิกายน 2559,
มหาวิทยาลัยนเรศวร, หน้า 115-122.

16. คิริภัทร เอี่ยมละอ้อและน้ำพลด มหาวิค. (2559). พื้นที่เสี่ยงต่อการเกิดไฟป่าจากฐานข้อมูลตำแหน่งไฟป่าปี 2547-2557 : กรณีศึกษา อุทยานแห่งชาติน้ำหนาว. เรื่องเต็มการประชุมวิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์ และสิ่งแวดล้อมนเรศวร ครั้งที่ 1, 3 พฤศจิกายน 2559, มหาวิทยาลัยนเรศวร, หน้า 174-182.
17. อภิชญา ยอดยิ่งและน้ำพลด มหาวิค. (2559). การเปรียบเทียบปริมาณน้ำฝนรายวันระหว่างผลิตภัณฑ์น้ำฝนจากข้อมูลดาวเทียมกับข้อมูลน้ำฝนภาคพื้นดิน พื้นที่ภาคเหนือของประเทศไทย. เรื่องเต็มการประชุมวิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์ และสิ่งแวดล้อมนเรศวร ครั้งที่ 1, 3 พฤศจิกายน 2559, มหาวิทยาลัยนเรศวร, หน้า 99-106.

การนำเสนอภาคราย (Oral presentation)

1. Mahavik, N., Tantanee, S. 2018: Quality Assessment of Mosaicked Weather Radars Over the Chao Phraya River Basin, Thailand. *The Grand GMSARN International Conference 2018 on “Energy, Environment, and Development in GMS”*, 28-30 November 2018 at Ramada Encore Hotel Nanning, Nanning City, Guangxi Province, China.
2. Mahavik, N., Tantanee, S. 2018: The convective cloud properties extraction from weather radar reflectivity during SONCA tropical storm over the Lower Northern Thailand. *Asian Conference on Remote Sensing 2018*, 15-19 October 2018, Renaissance Hotel Kuala Lumpur, Kuala Lumpur, Malaysia.
3. Mahavik, N., Tantanee, S. 2018: Comparison of spatial error structures between TRMM products and APHRODITE over the Indochina Peninsula. *Asian Conference on Remote Sensing 2018*, 15-19 October 2018, Renaissance Hotel Kuala Lumpur, Kuala Lumpur, Malaysia.
4. Seejata, K., Yodying, A., Wongthadam, T., Mahavik, N., Tantanee, S., (2017). Assessment of flood hazard areas using Analytical Hierarchy Process over the Lower Yom Basin, Sukhothai Province. *7th International Conference on Building Resilience Using scientific knowledge to inform policy and practice in Disaster Risk Reduction* 27-29 November, 2018- Swissotel Le Concorde, Bangkok, Thailand.

5. Mahavik, N., S. Shige, T. Hayashi, M. K. Yamamoto, 2015: Orogenic propagating rain systems over the middle of Indochina observed by gauge-calibrated ground-based radar data. MAHASRI workshop, 4 March 2015, Nagoya, Japan.
6. Mahavik, N., T. Satomura, B. Sysouphanthavong, S. Phonevilay, M. Wakabayashi, and S. Baimoung, 2013: Calibration of rainfall observed by weather radars and rainfall pattern study in Indochina peninsular. IMPAC-T final workshop, 11-12 November 2014, Sendai, Japan.
7. Mahavik, N., T. Satomura, B. Sysouphanthavong, S. Phonevilay, M. Wakabayashi, and S. Baimoung, 2013: Rainfall pattern in the middle of Indochina Peninsular during 2009-2010 summer monsoon. IGU regional conference, 4-9 August 2013, Kyoto, Japan.
8. Mahavik, N., T. Satomura, B. Sysouphanthavong, S. Phonevilay, M. Wakabayashi, and S. Baimoung, 2013: Rainfall pattern in the middle of Indochina peninsular using Vientiane radar. MAHASRI/HyARC Workshop, 14 March 2013, Nagoya, Japan.
9. Mahavik, N., T. Satomura, B. Sysouphanthavong, S. Phonevilay, M. Wakabayashi, and S. Baimoung, 2013: Rainfall pattern in the middle of Indochina peninsular. IMPAC-T workshop, 28-29 January 2013, Bangkok, Thailand.

การนำเสนอภาคปีสเตอร์ (Poster presentation)

1. Mahavik, N., T. Satomura, B. Sysouphanthavong, S. Phonevilay, M. Wakabayashi, and S. Baimoung, 2013: Rainfall Estimated by Radar Reflectivity Calibrated by Rain Gauges and Rainfall Patterns in Indochina. GCOE final symposium, 2-3 December 2013, Kyoto, Japan.
2. Mahavik, N., T. Satomura, B. Sysouphanthavong, S. Phonevilay, M. Wakabayashi, and S. Baimoung, 2013: Application of Ground Based Radar and TRMM Rainfall for Hydrological Analysis, Part of Calibration of rainfall observed by weather radars and rainfall pattern study in Indochina peninsular. IMPAC-T final workshop, 11-12 November 2013, Sendai, Japan.

3. Mahavik, N., T. Satomura, B. Sysouphanthavong, S. Phonevilay, M. Wakabayashi, and S. Baimoung, 2013: Rainfall pattern in the middle of Indochina Peninsular during 2009-2010 summer monsoon. DACA-13, 8-12 July 2013, Davos, Switzerland.
4. Mahavik, N., T. Satomura, B. Sysouphanthavong, S. Phonevilay, M. Wakabayashi, and S. Baimoung, 2013: Rainfall pattern in the middle of Indochina Peninsula during 2009-2010 summer monsoon. JPGU 2013, 19-24 May 2013, Chiba, Japan.
5. Mahavik, N., T. Satomura and S. Baimoung, 2012: Z-R parameter variations using conventional weather radar in the middle of Indochina peninsular. Meteorological Society of Japan (MSJ), 3-6 October 2012, Hokkaido, Japan.
6. Mahavik, N., T. Satomura, B. Sysouphanthavong, S. Phonevilay, M. Wakabayashi, and S. Baimoung, 2012: Radar rainfall analysis in the middle of Indochina peninsular. International symposium on GCOE ARS, 3-4 August 2012, Kyoto, Japan.
7. Mahavik, N., T. Satomura and S. Baimoung, 2012: Calibration of weather radar for analyzing precipitation distribution in Thailand. Meteorological Society of Japan (MSJ), 25-28 May 2012, Tsukuba, Japan.
8. Mahavik, N., T. Satomura and S. Baimoung, 2011: Investigation ZR relationship by using weather radar in Thailand. IMPAC-T Joint conference with SEA water 9th conference, 1-3 December 2012, Bangkok, Thailand.
9. Mahavik, N., T. Satomura, B. Sysouphanthavong, S. Phonevilay, M. Wakabayashi, and S. Baimoung, 2011: Precipitation characteristics in Thailand using weather radars and surface meteorological observations. Meteorological Society of Japan (MSJ), 16-19 May 2011, Tokyo, Japan.

ประสบการณ์ในงานวิจัย

1. 2558-2559 (หัวหน้าโครงการ) โครงการพัฒนากระบวนการทำแผนที่ข้อมูลน้ำฝนเชิงเลขจากเรดาร์ตรวจอากาศภาคพื้นดินด้วยซอฟต์แวร์ระบบภูมิสารสนเทศรหัสเปิด ทุนพัฒนาบุคลากร มหาวิทยาลัยนเรศวร (งบประมาณ 220,000 บาท)
2. 2560-2561(หัวหน้าโครงการ) โครงการพัฒนาวิธีการทำแผนที่โมเดลน้ำฝนประมาณค่าจากข้อมูลเรดาร์ตรวจอากาศภาคพื้นดินบริเวณลุ่มน้ำเจ้าพระยา ทุนงบประมาณแผ่น วช. มหาวิทยาลัยนเรศวร (งบประมาณ 583,000 บาท)

3. 2560-2561(หัวหน้าโครงการ) โครงการความผันแปรโครงสร้างภัยภาระบนจากข้อมูลดาวเทียม TRMM บริเวณอินโดจีนตอนกลาง ทุนงบประมาณแผ่น วช. มหาวิทยาลัยนเรศวร (งบประมาณ 599,500.00)
4. 2562-2563(หัวหน้าโครงการ) โครงการการวิเคราะห์รูปแบบฝนบนพื้นฐานข้อมูลภูมิสารสนเทศ เรดาร์ตรวจอากาศภาคพื้นดินรายละเอียดสูงในลุ่มน้ำยมและน่าน ทุนงบประมาณแผ่น วช. มหาวิทยาลัยนเรศวร (งบประมาณ 460,000 บาท)
5. 2560-2564(ผู้ร่วมวิจัย) โครงการ Advancing Co-design of Integrated Strategies with Adaptation to Climate Change in Thailand (ADAP-T) สนับสนุนโดย Science and Technology Research Partnership for Sustainable Development (SATREPS), JST-JICA (6,000,000 เยน)
6. 2553-2557 (ผู้ร่วมวิจัย) โครงการ Integrated Study Project on Hydro-Meteorological Prediction and Adaptation to Climate Change in Thailand (IMPAC-T) สนับสนุนโดย JICA/JST
7. 2559-2561 (ผู้ร่วมวิจัย) โครงการศึกษาความเป็นธรรมทางสุขภาพ สำนักงานกองทุนสนับสนุนการสร้างเสริมสุขภาพ (สสส.)

การเข้ารับฝึกอบรม (Academic training)

1. Technical Training on Forest Stand Height Estimation Using Synthetic Aperture Radar Data organized by SERVIR-Global, 13-15, January 2020, Asian Disaster Preparedness Center, SM Tower, Bangkok, Thailand,
2. Aerosol Modeling Workshop organized by National Astronomical Research Institute of Thailand (NARIT), University of Phayao (UP) & Hydro and Agro Informatics Institute (HAI), , February 19 – 21, 2019, Kantary Hills Hotel, Chiang Mai, Thailand.
3. Lidar data for biomass estimation by Silva Carbon supported by USAID, 5-11 February 2017, The Bazaar Hotel, Bangkok, Thailand
4. Landuse and Landcover change using remote sensing by SERVIR Mekong and ADPC supported by USAID, October 2015, Pak Nai Lert Hotel, Bangkok, Thailand.
5. Capacity building and brainstorming by SERVIR Mekong and ADPC supported by USAID, November 2015, Pak Nai Lert Hotel, Bangkok, Thailand.
6. IHP 22nd Precipitation measured by Satellite, 18 November – 1 December 2012 at Nagoya university, Nagoya, Japan.

7. Climate change downscaling approaches and applications, 9-14 November 2011 at United Nations University, Tokyo, Japan.
8. TRMM and GSMP training, 29 August-2 September 2011, by JAXA at Kyoto university and visited JAXA, Kyoto-Chiba-Tokyo, Japan

วิทยากร/ผู้ทรงคุณวุฒิ

1. 19 มกราคม 2563. วิทยากรอบรมโครงการการประมวลผลข้อมูลเรดาร์ตรวจอากาศภาคพื้นดินด้วยซอฟต์แวร์ IDV, การประชุมวิชาการนิสิตนักศึกษาภูมิศาสตร์และภูมิสารสนเทศแห่งประเทศไทย ครั้งที่ 12. อาคารເອກາະສົດ ມາຮວຍລ້ຽນເຮົວວຽງຈັນ ຈັງຫວັດພິບປຸລືໂລກ
2. 12-13 ธันวาคม 2562. ผู้วิพากษ์ในงานประชุมวิชาการทรัพยากรธรรมชาติ สารสนเทศภูมิศาสตร์และสิ่งแวดล้อม ครั้งที่ 4, 12-13 ธันวาคม 2562, ມາຮວຍລ້ຽນພະເຍາ.
3. 4-5 พฤศจิกายน 2562. วิทยากรอบรมโครงการการประมวลผลข้อมูลเรดาร์ตรวจอากาศด้วยภาษา Python. ຈຳກັດວຽກ Python 2019 Thailand. ໃນ ຈຸ່າລັງກຽມທ້າວີທາລະຍ. ກຽມທ່ານຄຣ
4. 6 ตุลาคม 2562. วิทยากรอบรมโครงการ อบรมการทำแผนที่อากาศيانรีັກນັບເພື່ອການຈັດການຕ້ານເກະທຽບຮົມ. ມາຮວຍລ້ຽນຮາມຄລ້ານນາວີທາເຂດພິບປຸລືໂລກ.
5. 20 กันยายน 2562. วิทยากรอบรมโครงการการประมวลผลข้อมูลเรดาร์ตรวจอากาศเชิงภูมิสารสนเทศด้วยภาษา Python. ກາຄວິชาภົມີສາສົງ. ຄະນະມນຸຍາສາສົງແລະສັງຄມສາສົງ ມາຮວຍລ້ຽນຮາມຄລ້ານນາວີທາເຂດພິບປຸລືໂລກ.
6. 19 กันยายน 2562. วิทยากรอบรมโครงการการประมวลผลข้อมูลเรดาร์ตรวจอากาศเชิงภูมิสารสนเทศด้วยภาษา Python. ກາຄວິชาภົມີສາສົງ. ຄະນະສັງຄມສາສົງ ມາຮວຍລ້ຽນເຊິ່ງໃໝ່.
7. 5-9 กันยายน 2562. วิทยากรอบรมโครงการการประมวลผลข้อมูลเรดาร์ตรวจอากาศด้วยภาษา Python. ກຣມອຸດຸນີມວິທາ. ກຽມທ່ານຄຣ.
8. 30 สิงหาคม 2562. ຜູ້ทรงคุณวุฒิร่วมเสวนາໂຄງການທຶນທາງອນໄຕຕະຫຼາດສາທິປະໄຕ. ກາຄວິชาภົມີສາສົງ. ຄະນະສັງຄມສາສົງ ມາຮວຍລ້ຽນເຊິ່ງໃໝ່.
9. 10 กรกฎาคม 2562. วิทยากรอบรมเรื่อง “ຂໍ້ມູນດາວເຖິມ TRMM ເພື່ອການປະເມີນຄ່າຝານ. ອາກະ CITCOM. ມາຮວຍລ້ຽນເຮົວວຽງຈັນ ພິບປຸລືໂລກ.
10. 4 กรกฎาคม 2562. ວິທາກຣອບຣມໜ້າຂ້ອງການປະເມີນຄ່າຝານ. ຈຳກັດວຽກ Python. ຈຳກັດວຽກ Python. ປະເທດຕະຫຼາດສາທິປະໄຕ. ມາຮວຍລ້ຽນເຮົວວຽງຈັນ ພິບປຸລືໂລກ.
11. 31 มกราคม 2562. ຜູ້ทรงคุณวุฒิວິພາກຍໍ່ຫັກສູງຕະຫຼາດສາທິປະໄຕ. ກາຄວິชาภົມີສາສົງ. ຄະນະມນຸຍາສາສົງ ແລະສັງຄມສາສົງ ມາຮວຍລ້ຽນຮາມຄລ້ານນາວີທາເຂດພິບປຸລືໂລກ.
12. 6 พฤศจิกายน 2561. ວິທາກຣອບຣມໜ້າການການປະເມີນຄ່າຝານ. ກາຄວິชาພົມີສາສົງ. ຄະນະມນຸຍາສາສົງ ແລະສັງຄມສາສົງ ມາຮວຍລ້ຽນຮາມຄລ້ານນາວີທາເຂດພິບປຸລືໂລກ.
13. 29 สิงหาคม 2561. ວິທາກຣສ່ວນາໂຄງກາງງານສ່ວນາຫ້າຫລວງ-ຊຸມນ້ຳນາງນອນ. ກາຄວິชาພົມີສາສົງ. ຄະນະສັງຄມສາສົງ ມາຮວຍລ້ຽນເຊິ່ງໃໝ່.

14. 24 กรกฎาคม 2561. วิทยกรอบรมโครงการประกวดผลข้อมูลเรคาร์ดตรวจสอบภาษาไทย งานFoss4G 2018 Thailand. ณ มหาวิทยาลัย AIT. กรุงเทพมหานคร

