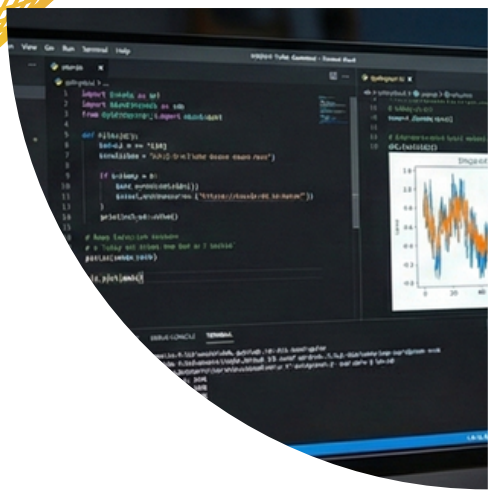
เอกสารประกอบการฝึกอบรมเชิงปฏิบัติการ
หลักสูตรการประยุกต์ใช้ภาษาไพธอน (Python)
เพื่องานทางอุทกวิทยา

สำนักบริหารจัดการน้ำและอุทกวิทยา

25-27 มีนาคม พ.ศ.2569



จัดทำโดย

นายภควัต ลำจวน วิศวกรโยธาชำนาญการพิเศษ

สำนักออกแบบวิศวกรรมและสถาปัตยกรรม

คำนำ

ในยุคปัจจุบัน เทคโนโลยีดิจิทัลและการวิเคราะห์ข้อมูลขนาดใหญ่ (Big Data) ได้เข้ามามีบทบาทสำคัญอย่างยิ่งต่อการบริหารจัดการทรัพยากรน้ำ การปฏิบัติงานด้านอุทกวิทยามีความจำเป็นต้องจัดการกับข้อมูลที่มีปริมาณมหาศาลและมีความซับซ้อน ไม่ว่าจะเป็นข้อมูลระดับน้ำ ปริมาณน้ำฝน หรือข้อมูลอนุกรมเวลา (Time Series) ที่ต่อเนื่องยาวนาน ซึ่งวิธีการประมวลผลแบบดั้งเดิมอาจไม่เพียงพอต่อการตอบสนองความต้องการที่รวดเร็วและแม่นยำในสถานการณ์ปัจจุบัน

เอกสารประกอบการฝึกอบรมหลักสูตร "การประยุกต์ใช้ภาษาไพธอน (Python) เพื่องานทางอุทกวิทยา" ฉบับนี้ จัดทำขึ้นเพื่อมุ่งเน้นถ่ายทอดองค์ความรู้และทักษะการใช้งานภาษาไพธอน ซึ่งเป็นเครื่องมือที่ได้รับการยอมรับในระดับสากลว่าเป็นภาษาที่มีประสิทธิภาพสูงสำหรับงานวิศวกรรมและวิทยาศาสตร์ข้อมูล เนื้อหาในหลักสูตรครอบคลุมตั้งแต่พื้นฐานการเขียนโปรแกรม การติดตั้งเครื่องมือ การจัดการและทำความสะอาดข้อมูลทางอุทกวิทยา การวิเคราะห์ทางสถิติ ไปจนถึงการประยุกต์ใช้เทคโนโลยีขั้นสูงอย่างโครงข่ายประสาทเทียม (Artificial Neural Networks: ANN) เพื่อการพยากรณ์น้ำ

ผู้จัดทำหวังเป็นอย่างยิ่งว่า เอกสารฉบับนี้จะเป็นประโยชน์ต่อนักอุทกวิทยา วิศวกร และผู้สนใจ ให้สามารถนำความรู้ไปประยุกต์ใช้ในการวิเคราะห์ข้อมูล วางแผน และบริหารจัดการน้ำได้อย่างมีประสิทธิภาพทันสมัย และเกิดประโยชน์สูงสุดต่อองค์กรและประเทศชาติต่อไป

นายภควัต ลำจวน

มีนาคม 2569

สารบัญ

	หน้า
คำนำ	
สารบัญ	
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	1
1.3 คำถามในการวิจัย	2
1.4 สมมติฐานในการวิจัย	2
1.5 นิยามศัพท์ที่ใช้ในการวิจัย	2
1.6 ขอบเขตของการวิจัย	3
1.7 ประโยชน์ที่คาดว่าจะได้รับ	3
1.8 กรอบความคิดในการวิจัย	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	
2.1 ทฤษฎีที่เกี่ยวข้อง	8
2.2 ทฤษฎีโครงข่ายประสาทเทียม	8
2.3 สถาปัตยกรรมโครงข่ายประสาทเทียมเพอร์เซฟตรอนแบบหลายชั้น	14
2.4 ลักษณะโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ	16
2.5 วิธีการเรียนรู้แบบแพร่ย้อนกลับ	16
2.6 ขั้นตอนการเรียนรู้ของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ	18
2.7 ข้อดีและข้อเสียของการเรียนรู้แบบแพร่กลับ (DAYHOFT, 1990)	20
2.8 หลักการประยุกต์ใช้ ARTIFICIAL NEURAL NETWORKS (ANNS) MODEL	21
2.9 งานวิจัยที่เกี่ยวข้อง	23
บทที่ 3 วิธีการดำเนินงานวิจัย	
3.1 ขั้นตอนการดำเนินงาน	28
3.2 การวิเคราะห์ข้อมูลด้วยโปรแกรม PYTHON	29
บทที่ 4 ผลและการวิจารณ์	72
4.1 ผลการหาค่าสัมประสิทธิ์สหสัมพันธ์	72
4.2 ผลของการ VALIDATION ข้อมูล	76
4.3 ผลการทดสอบแบบจำลองโดยการ RUN เป็น TIMESERIES	77
บทที่ 5 สรุปและข้อเสนอแนะ	79
5.1 สรุปผลการศึกษา	79
5.2 ปัญหาและอุปสรรค	80
5.3 ข้อเสนอแนะ	80

สารบัญ (ต่อ)

	หน้า
อธิบายโค้ดโปรแกรม DATA1Y1COL (เรียงข้อมูลอัตราการไหลรายวันหลายปีในไฟล์เดียวให้เป็นไฟล์ .CSV รวมทุกปีแยกเป็นปีละ 1 คอลัมน์ในไฟล์เดียว	**87**
อธิบายโค้ดโปรแกรม MUSKINGUM ROUTING	**93**
การประยุกต์ใช้โปรแกรมคอมพิวเตอร์ด้วยภาษาไพธอน (การคำนวณทางด้านชลศาสตร์)	97

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ตั้งแต่อดีตจนถึงปัจจุบัน ประเทศไทยเกิดอุทกภัยที่รุนแรงหลายครั้ง โดยในปลายปี พ.ศ. 2554 ประเทศไทยได้ประสบปัญหาอุทกภัยครั้งใหญ่หรือที่นิยมเรียกกันว่ามหาอุทกภัย เป็นอุทกภัยรุนแรงที่ ส่งผลกระทบต่อบริเวณลุ่มแม่น้ำเจ้าพระยาและลุ่มแม่น้ำโขง เริ่มตั้งแต่ปลายเดือนกรกฎาคม พ.ศ. 2554 และสิ้นสุดเมื่อวันที่ 16 มกราคม พ.ศ. 2555 ซึ่งมีเหตุอันเนื่องมาจากฝนที่ตกหนักเป็นบริเวณ กว้างและฝนตกสะสมตลอดทั้งฤดู ปริมาณน้ำฝนจึงสะสมเป็นมวลน้ำจำนวนมหาศาล ทำให้ทางน้ำ ธรรมชาติไม่สามารถรองรับได้ จึงเอ่อล้นเข้าท่วมพื้นที่การเกษตร ที่อยู่อาศัย และพื้นที่อุตสาหกรรม อุทกภัยในครั้งนั้นได้สร้างความเสียหายรวมถึงความสูญเสียให้แก่ประชาชนและระบบเศรษฐกิจเป็น อย่างมาก มีราษฎรได้รับผลกระทบกว่า 12.8 ล้านคน ธนาคารโลกประเมินมูลค่าความเสียหายสูงถึง 1.44 ล้านล้านบาท (กรมอุตุนิยมวิทยา, 2554)

ทว่าในปัจจุบัน เมื่อเกิดฝนตกและมีปริมาณน้ำเพิ่มสูงขึ้น เราสามารถเตือนภัยการเกิดอุทกภัย ล่วงหน้าได้ หากมีการพยากรณ์ค่าปริมาณการไหลของน้ำที่แม่นยำและจะส่งผลให้มูลค่าการสูญเสีย จากการเกิดจากอุทกภัยนั้นลดลงได้ ซึ่งจากการศึกษาค้นคว้าและทดลอง ผู้จัดทำได้ใช้วิธีการวิเคราะห์ โครงข่ายประสาทเทียม (Artificial neural networks) และใช้โปรแกรมไพทอน (Python) เป็นตัวช่วยในการวิเคราะห์ข้อมูล โดยเลือกข้อมูลจากสถานีวัดน้ำท่า P.17 และสถานีใกล้เคียงคือ สถานีวัดน้ำท่า P.16,P.15,P.7A และ P.26A รวมไปถึงสถานีวัดน้ำฝน 260062 ของสำนักงานชลประทานที่ 3 มาเป็นกรณีศึกษา

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อพยากรณ์ปริมาณน้ำท่าที่จะเกิดขึ้นในอนาคต โดยนำข้อมูลของสถานีใกล้เคียงมา ช่วยเสริมในการพยากรณ์ปริมาณน้ำท่วม

1.2.2 ใช้วิธีการวิเคราะห์โครงข่ายประสาทเทียม (Artificial neural networks) เพื่อพยากรณ์ปริมาณน้ำท่าที่สถานีวัดน้ำ P.17

1.2.3 เพื่อตรวจสอบความแม่นยำของการพยากรณ์จากวัตถุประสงค์ข้อที่ 1.2.2

1.3 คำถามในการวิจัย

1.3.1 การใช้วิธีการวิเคราะห์โครงข่ายประสาทเทียม (Artificial neural networks) สามารถพยากรณ์ปริมาณน้ำท่ารายวันมีความแม่นยำได้มากสูงสุดกี่วัน

1.3.2 หากใช้ข้อมูลปริมาณน้ำท่ารายวัน อัตราการไหล และข้อมูลน้ำฝนรายวันโดยใช้ข้อมูลย้อนหลัง 10 ปี ประสิทธิภาพและหากปรับ Hyperparameters และ โครงสร้างชั้นซ่อน (Hidden layer) ของโครงข่ายประสาทเทียมความแม่นยำจะมากขึ้นเพียงใด

1.4 สมมติฐานในการวิจัย

1.4.1 สามารถใช้วิธีการวิเคราะห์โครงข่ายประสาทเทียม (Artificial neural networks) พยากรณ์ ปริมาณน้ำท่าล่วงหน้าได้ 1-3 วัน

1.4.2 ปริมาณน้ำท่าที่ได้จากการพยากรณ์โดยวิธีการวิเคราะห์โครงข่ายประสาทเทียม (Artificial neural networks) มีความแม่นยำล่วงหน้า 1-3 วันมากน้อยเพียงใด

1.5 นิยามศัพท์ที่ใช้ในการวิจัย

1.5.1 โครงข่ายประสาทเทียม (Artificial neural networks : ANNs)

1.5.2 โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Feed-forward Backpropagation Neural Networks)

1.5.3 การประมวลผลแบบดั้งเดิม (traditional approach) คือ เป็นการใช้ Server ทำหน้าที่ประมวลผล DATA ต่างๆ สามารถเพิ่ม CPU RAM ตามที่เราต้องการ เชื่อมต่อกับ Storage ทำหน้าที่เก็บข้อมูล ประกอบด้วย Storage controller และ Disk ซึ่ง Server กับ Storage แยกการทำงานกันอย่างชัดเจน

1.5.4 การพยากรณ์ (Prediction)

1.5.5 รากที่สองของค่าเฉลี่ยความผิดพลาดกำลังสอง (Root Mean Square Error: RMSE)

1.5.6 ค่าสัมประสิทธิ์สหสัมพันธ์ (Correlation Coefficient : R) เกณฑ์ในการเลือกใช้ต้องมี ค่ามากกว่าหรือเท่ากับ 0.8

1.5.7 การประมวลผลแบบโครงข่ายประสาทเทียม (neural networks approach)

1.6 ขอบเขตการวิจัย

1.6.1 ใช้โปรแกรม Python มาช่วยในการนำเสนอและวิเคราะห์ข้อมูล ที่ได้จากวิธีการวิเคราะห์โครงข่ายประสาทเทียม (Artificial neural networks)

1.6.2 ข้อมูลปริมาณน้ำฝน น้ำท่า รายวันตั้งแต่ปี ค.ศ. 2010-2021

1.6.3 พื้นที่ศึกษาสถานีวัดปริมาณน้ำ P.17 ที่ตั้งอยู่ในลุ่มแม่น้ำปิงและใช้ข้อมูลปริมาณน้ำสถานีข้างเคียงที่เกี่ยวข้อง สถานีน้ำฝน 120161

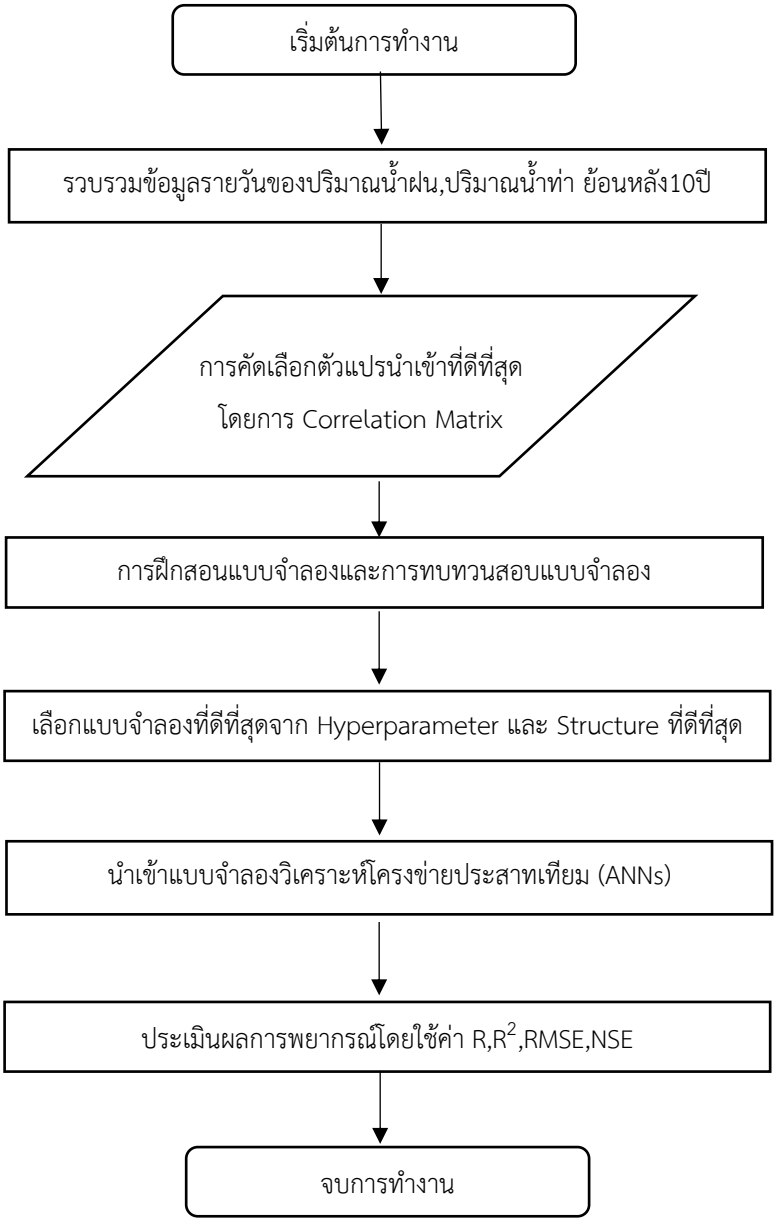
1.7 ประโยชน์ที่คาดว่าจะได้รับ

1.7.1 ได้แบบจำลองโครงข่ายประสาทเทียมพยากรณ์ปริมาณน้ำท่า ที่ให้มีความแม่นยำเพื่อนำไปใช้เตือนภัยน้ำท่วมให้กับสำนักบริหารจัดการน้ำและอุทกวิทยา กรมชลประทาน

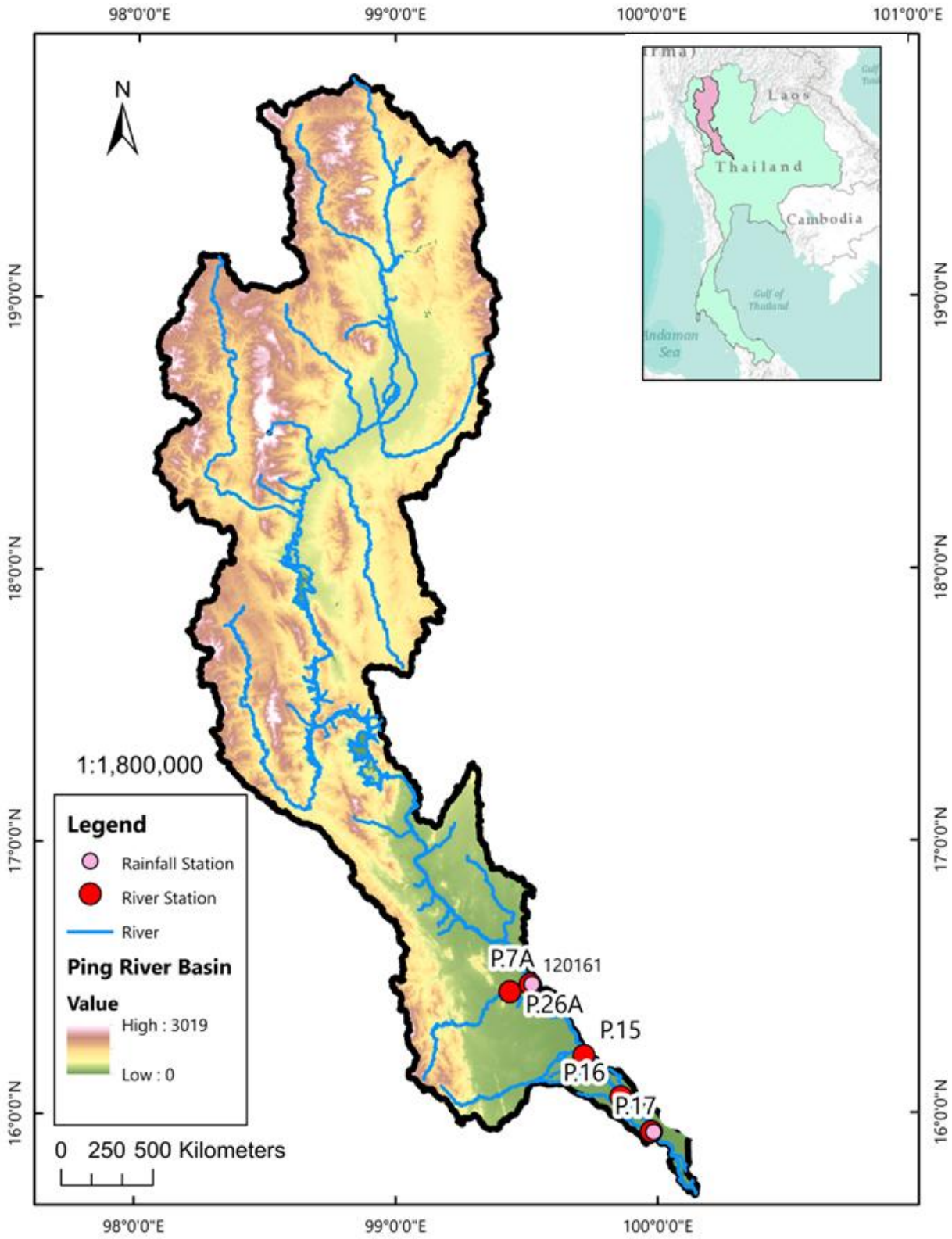
1.7.2 ได้องค์ความรู้ของการพยากรณ์ปริมาณน้ำท่ารายวันโดยการใช้เทคนิค Machine Learning

1.8 กรอบความคิดในการวิจัย

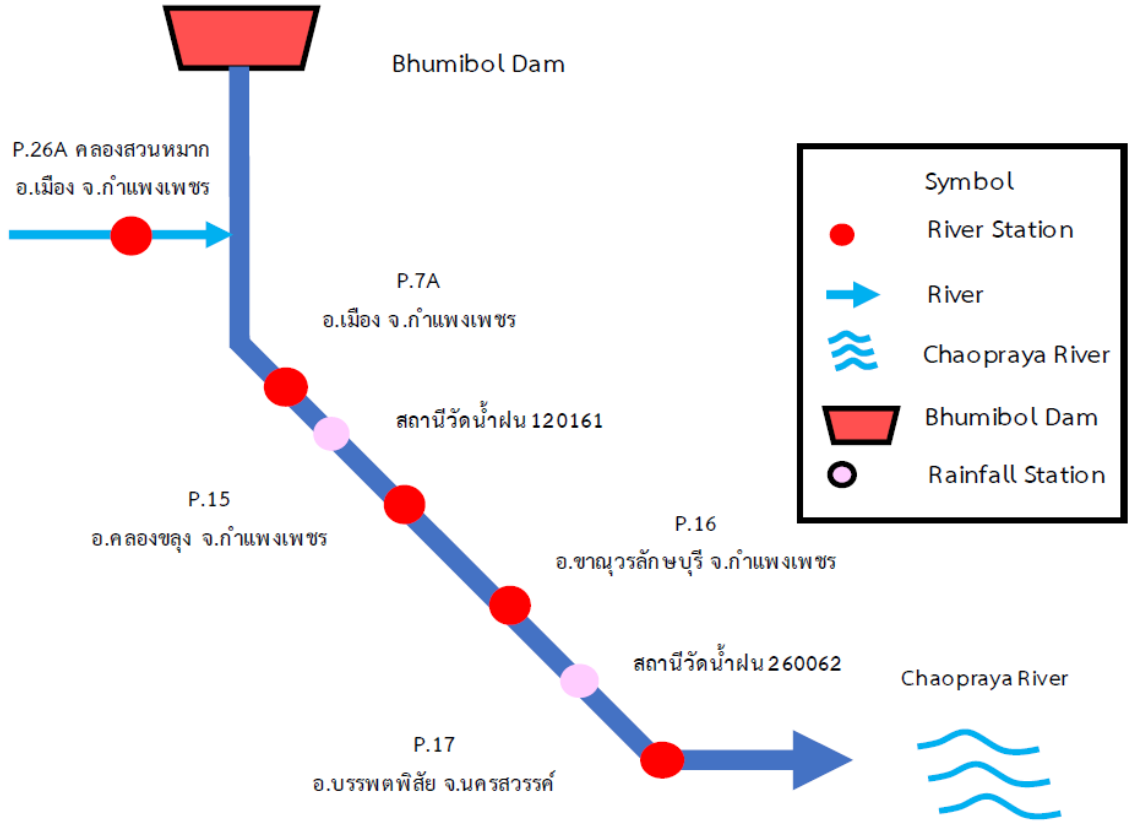
กรอบความคิดในการวิจัย แสดงในภาพที่ 1.1 ดังต่อไปนี้

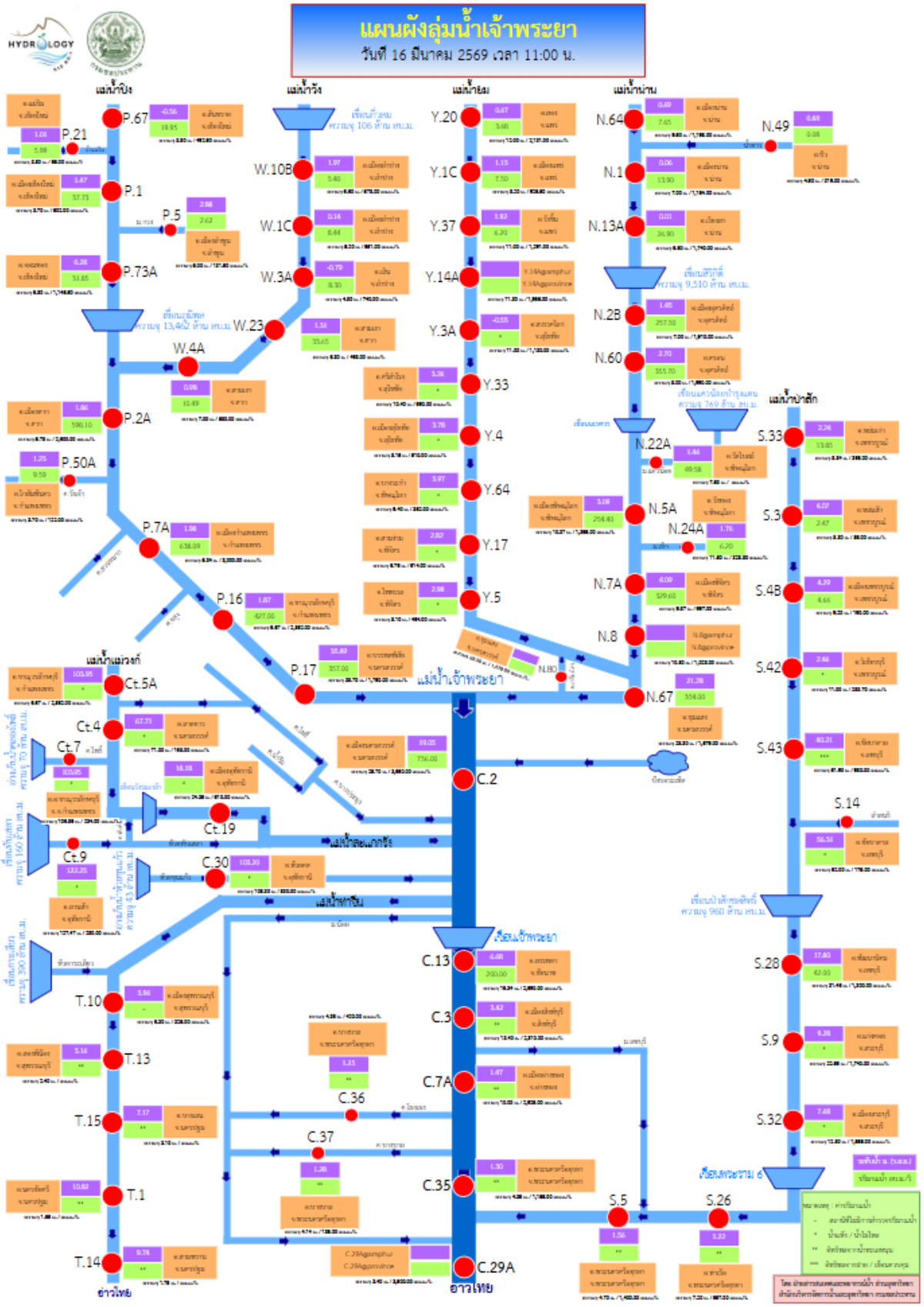


ภาพที่ 1.1 กรอบความคิดในการวิจัย


98°0'0"E
99°0'0"E
100°0'0"E
101°0'0"E
19°0'0"N
18°0'0"N
17°0'0"N
16°0'0"N
N
Laos
Thailand
Cambodia
Gulf of Thailand
Andaman Sea
1:1,800,000
Legend
Rainfall Station
River Station
River
Ping River Basin
Value
High : 3019
Low : 0
0 250 500 Kilometers
P.7A
120161
P.26A
P.15
P.16
P.17

ขอบเขตของสถานีศึกษาและแผนผังแสดงการไหลของลุ่มน้ำปิงตอนล่าง




แผนผังลุ่มน้ำเจ้าพระยา
วันที่ 16 มีนาคม 2569 เวลา 11:00 น.
HYDROLOGY
แม่น้ำปิง
แม่น้ำวัง
แม่น้ำยม
แม่น้ำน่าน
แม่น้ำป่าสัก
แม่น้ำแม่กลอง
แม่น้ำเจ้าพระยา
P.67
P.21
P.1
P.5
P.73A
W.10B
W.1C
W.3A
W.23
W.4A
P.2A
P.50A
P.7A
P.16
P.17
Y.20
Y.1C
Y.37
Y.14A
Y.3A
Y.33
Y.4
Y.64
Y.17
Y.5
N.64
N.49
N.1
N.13A
N.28
N.60
N.22A
N.5A
N.24A
N.7A
N.8
N.67
N.80
S.33
S.3
S.48
S.42
S.43
S.14
S.28
S.9
S.32
S.5
S.26
Ct.5A
Ct.4
Ct.7
Ct.19
Ct.9
C.30
T.10
T.13
T.15
T.1
T.14
C.2
C.13
C.3
C.7A
C.36
C.37
C.35
C.29A
ย่านาวัย
อาวไทย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

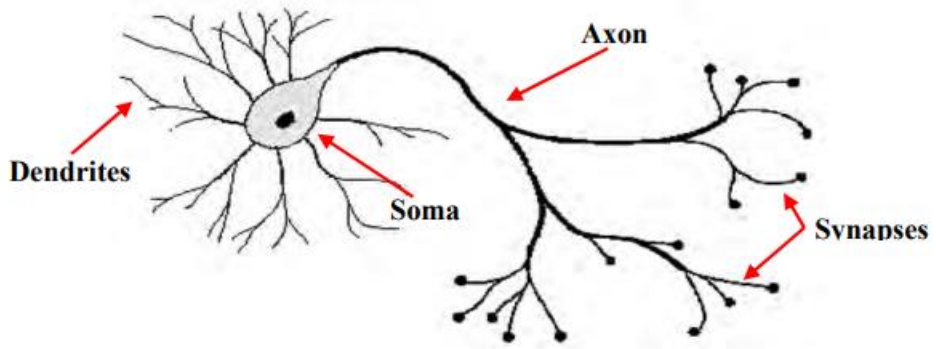
2.1 ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมถูกสร้างขึ้นเพื่อเลียนแบบการทำงานของสมองมนุษย์ มี การทำงานแบบขนานจำนวนมาก ในงานวิจัยนี้เลือกใช้โครงข่ายประสาทเทียมแบบเพอร์เซฟตรอนหลายชั้น (Multi-layer perceptron Neural Networks) หรือโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Feed-forward Backpropagation Neural Networks) ซึ่งเป็นโครงข่ายประสาทเทียมแบบต้องมีผู้สอน (Supervised Learning)

เรื่องราวของโครงข่ายประสาทเทียมนั้น มีพื้นฐานมากจากนิวรอนที่เป็นของจริง ในสมองมนุษย์ นิวรอนหรือเซลล์ประสาทในสมองนั้นจะมีหน้าที่หลักสองประการ คือ การคำนวณ และทำการส่งผลที่ได้จากการคำนวณไปยังอีกปลายหนึ่งของเซลล์ประสาทอย่างรวดเร็ว เพื่อให้สามารถส่งผลดังกล่าวไปยังเซลล์อื่น ๆ ได้อย่างทันท่วงที ซึ่งทั้งหมดนี้จะทำงาน โดยอาศัยหลักการทางไฟฟ้า โดยรูปร่างลักษณะของนิวรอนในสมองมนุษย์นั้นดังแสดงใน ภาพประกอบที่ 2.1 ซึ่งจะมีลักษณะคล้ายกับต้นไม้ที่ปราศจากใบ ที่แต่ละกิ่งและรากที่เชื่อมโยงกัน ด้วยลำต้น โดยส่วนประกอบการทำงานหลักของข่ายประสาทคือเซลล์ประสาท และกลุ่มเซลล์ ประสาทนั้นจะมีการรับรู้ โดยมีการทำงานคือจะรับอินพุต (Input) จากแหล่งต่างๆ นำมารวมเข้า ด้วยกันและมีการทำงานในแบบไม่เป็นเส้นตรง (Non-Linear) และส่งเอาท์พุตสุดท้ายออกมา

มนุษย์นั้นมีความซับซ้อนในกระบวนการคิดซ้ำ ทำให้เซลล์ประสาทมี ความหลากหลายอย่างมาก อย่างไรก็ดีเซลล์ประสาทตามธรรมชาติทั้งหมดมีส่วนประกอบพื้นฐาน 4 ส่วนเหมือนๆ กัน ส่วนประกอบเหล่านี้เรียกตามชื่อทางชีววิทยาว่า เดนไดร์ท (Dendrites), โซม่า (Soma), แอ็คซอน (Axon) และซินแนปส์ (Synapses) ดังแสดงในภาพ



ภาพที่ 2.1 แสดงความสัมพันธ์ของระบบประสาททั้ง 4 ส่วน (Maureen Candill, 1989)

เดนไดร์ทเป็นส่วนขยายหรือส่วนต่อของโซม่า ซึ่งมีลักษณะคล้ายขนซึ่งทำหน้าที่ เหมือนเป็นช่องทางนำเข้าของค่าอินพุต และเดนไดร์ทจะทำการรับอินพุตผ่านซินแนปส์ของเซลล์ประสาทอื่น โดยโซม่าจะคอยประมวลผลสัญญาณไฟฟ้าที่รับเข้ามาตลอดเวลา แล้วส่งผลการทำงานเป็นเอาท์พุตออกไปให้เซลล์ประสาทอื่น โดยผ่านทางแอ็คซอนและซินแนปส์

โครงข่ายประสาทเทียมนั้น มีคุณสมบัติที่เทียบเท่ากับสมองมนุษย์คนเรา ในด้าน ของการเรียนและจดจำ เช่น สมองของมนุษย์เรานั้น เมื่อได้รับการเรียนรู้บางสิ่งหลายๆครั้งนั้นก็จะเกิดการจดจำและเมื่อพบเห็นสิ่งที่ไม่เคยได้เรียนรู้มาก่อนก็สามารถที่จะอนุมานได้ว่าสิ่งนั้นคืออะไร จากความรู้ที่ได้รับจากการเรียนรู้ในก่อนหน้าที่ผ่านมา

โครงข่ายประสาทมีคุณสมบัติ 2 ประการ คือ (Wasserman, 1998)

1) การเรียนรู้ (learning) โครงข่ายประสาทสามารถเรียนรู้จากชุดฝึกสอนที่ได้ทำการป้อนให้ โครงข่ายประสาทเทียม ได้เรียนรู้

2) การระลึกหรือจดจำได้ (recall)

โครงข่ายประสาทเทียมในทัศนะของคอมพิวเตอร์นั้น จะประกอบไปด้วย หน่วยประมวลผล(processing elements (PE) ที่เชื่อมโยงกันหลายตัว ทำงานในลักษณะขนานกัน ไปคล้ายกับนิวรอนในสมองของมนุษย์ เพื่อแปลงข้อมูลจากรูปแบบหนึ่งไปสู่อีกรูปแบบหนึ่ง การใช้โครงข่ายประสาทเทียมนี้จะเป็นไปรูปแบบของการสอนให้คอมพิวเตอร์เรียนรู้แทนที่จะเป็น การป้อนโปรแกรมให้กับคอมพิวเตอร์โดยจุดมุ่งหมายของการสอนนิวรอนหรือการป้อนข้อมูลที่

ต้องการให้กับคอมพิวเตอร์เรียนรู้ คือ การทำให้ระบบคอมพิวเตอร์นี้สามารถแสดงคำตอบในรูปแบบที่ต้องการได้ ซึ่งจะเป็นการป้อนข้อมูลที่ถือว่ารู้อยู่แล้วเข้าไปให้โครงข่ายประสาทเทียมพร้อมด้วยค่าเอาท์พุตที่ต้องการให้โครงข่ายประสาทเทียมนั้นแสดงออกมาจากนั้นโครงข่ายประสาทเทียมจะทำการคำนวณและปรับค่าตัวเลขน้ำหนักเองโดยใช้กฎเกณฑ์ต่าง ๆ เข้าช่วยจนกระทั่งเอาท์พุตที่ได้ออกมานั้นถูกต้องแม่นยำอยู่ในเกณฑ์ที่น่าพอใจ ส่วนตัวเลขที่เป็นกลไกที่ทำให้โครงข่ายประสาทเทียมสามารถเรียนรู้และจดจำได้นั้นจะอยู่ในรูปที่เรียกว่า “ เมทริกซ์น้ำหนัก” โดยในงานวิจัยนี้ได้กำหนดในรูปแบบของ “ เมทริกซ์ของคำและค่าน้ำหนักของแต่ละคำหรือวลี ” ลักษณะการทำงานแบบนี้จะทำให้ผู้ใช้โครงข่ายประสาทเทียม สามารถที่จะป้อนข้อมูลใหม่ๆ เข้าไปแล้วปล่อยให้เป็นหน้าที่ของโครงข่ายประสาทเทียมชนิดนั้น ๆ ที่จะหาทางที่จะจัดการ ปรับตัวเพื่อที่จะหาคำตอบนั้นเอาเอง ความก้าวหน้าของโครงข่ายประสาทเทียมในปัจจุบันเป็นไป อย่างรวดเร็ว ทำให้สามารถแก้ปัญหาให้กับงานบางประเภทที่ไม่สามารถทำงานได้จนสัมฤทธิ์ผล และให้คำตอบในแบบที่ผู้ใช้งานพึงพอใจ

ข้อเปรียบเทียบระหว่างการประมวลผลแบบดั้งเดิมกับโครงข่ายประสาทเทียม

1. การประมวลผลแบบดั้งเดิม (traditional approach)เป็นการวิเคราะห์ข้อมูลโดยมนุษย์ ดังนั้นทรัพยากรมนุษย์จะถูกใช้ใน การพัฒนาอัลกอริทึมและโปรแกรม

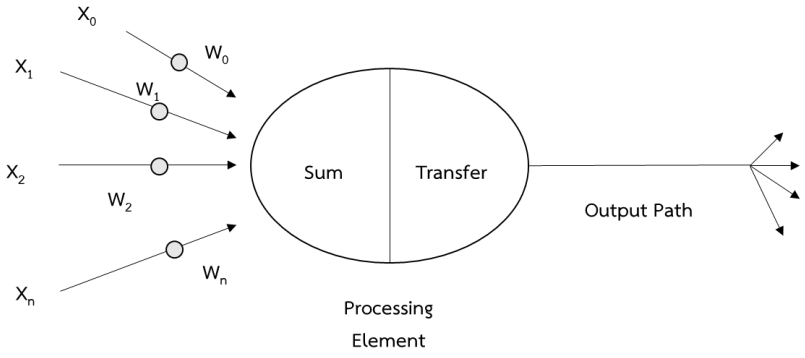
2. การประมวลผลแบบโครงข่ายประสาทเทียม (neural networks approach)

ในกระบวนการสอน จะมีการสอนวนซ้ำหลาย ๆ รอบ เพื่อให้โครงข่าย ประสาทเทียมเกิดการเรียนรู้ เมื่อสิ้นสุดการสอนแล้วโครงข่ายประสาทเทียมก็จะสามารถจำแนก ข้อมูลได้ และเมื่อมีข้อมูลใหม่ ๆ ที่ต้องการให้โครงข่ายรู้จักก็สามารถทำในทำนองเดียวกันและ ต้องมีการสอนใหม่ แต่ทำให้ประหยัดเวลาและแรงงานในการพัฒนาโปรแกรมขึ้นมาใหม่

การเก็บข้อมูลของโครงข่ายประสาทเทียมเป็นแบบกระจายและถูกใช้ร่วมกัน โดย หลาย ๆ เซลล์ประสาท ซึ่งต่างกับแบบดั้งเดิม คือ ข้อมูลจะเก็บไว้ในหน่วยความจำ การเก็บข้อมูล ของโครงข่ายประสาทเทียมแบบกระจายนั้นทำให้เกิดความซ้ำซ้อน ซึ่งเป็นการเพิ่มความทนทาน คือเป็นระบบสำรองทดแทน (faultierror tolerance system)

รูปแบบของโครงข่ายประสาทเทียม ส่วนใหญ่จะมีองค์ประกอบที่สำคัญ เช่น รูปแบบการเรียนรู้ภายในส่วนโพรเซสซิ่ง มีการบวก การคูณ การลบตัวเลขบรรจุอยู่ หน่วยโพรเซสซิ่งในโครงข่ายประสาทเทียมจะทำงานสัมพันธ์กัน โดยขึ้นกับการเชื่อมโยงกันหมด ทุกส่วนหรือเชื่อมโยงเพียงบางส่วนก็ได้ การสร้างโครงข่ายประสาทเทียมต้องเริ่มด้วยการพิจารณาจากเรื่องของการเชื่อมต่อกันภายใน และสถาปัตยกรรมที่สนับสนุนการสร้างโครงข่ายประสาท เทียมให้เป็นไปได้

หลักการทำงานของโครงข่ายประสาทเทียมจะมีอินพุตหลายค่าเข้ามาในโครงข่าย โดยจะถูกแทนด้วย สัญลักษณ์ทางคณิตศาสตร์ X(n) และแต่ละอินพุตนั้นจะถูกคูณด้วยค่าความรู้หรือเป็นค่าน้ำหนัก (weight) ซึ่งแทนด้วย W(n) โดยที่ปกติผลคูณของค่าน้ำหนักและอินพุตที่เข้าสู่โครงข่ายนั้น จะถูก นำมารวมกันและส่งผ่านเข้าไปในฟังก์ชัน (transfer function) เพื่อที่จะหาเอาท์พุตหรือผลลัพธ์ ออกมา โดยกระบวนการนี้นั้นทำให้ง่ายต่อการใช้งานและสามารถนำไปใช้กับโครงสร้างวงจร โครงข่ายอื่นที่ใช้ฟังก์ชันผลรวม (Summing functions) และฟังก์ชันการส่งผ่านที่ต่างกันได้ โดยที่ บางแอพพลิเคชั่นนั้น ต้องการคำตอบที่เป็น “ใช่หรือไม่ใช่” หรือค่าไบนารี (binary) โดย แอพพลิเคชั่นนั้นอาจจะรวมถึงการจดจำข้อความ การชี้เฉพาะคำพูด และการแปลความหมาย รูปภาพของเหตุการณ์



ภาพที่ 2.2 แสดงการทำงานของเซลล์ประเทียม

โดยทั่วไปการเรียนรู้ของโครงข่ายประสาทเทียม ก็คือ การสอนโครงข่ายให้ทำ การคำนวณข้อมูลเอาท์พุตพร้อมกับปรับปรุงค่าน้ำหนักโดยใช้ข้อมูลอินพุตที่ป้อนให้กับโครงข่าย โดยอาศัยกระบวนการทำซ้ำ (iterative) สามารถแบ่งเป็น 2 ประเภท คือ

1. การเรียนรู้แบบมีผู้สอน

การเรียนรู้แบบมีผู้สอนนั้นต้องการชุดข้อมูลอินพุตและเอาท์พุตเป้าหมายเป็นชุด ฝึกสอนควบคู่ (training pair) โดยปกติการสอนโครงข่ายนั้นจะใช้ชุดฝึกสอนควบคู่กันหลายชุด ในระหว่างการสอนโครงข่ายจะเกิดเอาท์พุตจริงซึ่งแตกต่างจากเอาท์พุตเป้าหมายทำให้ได้ค่าความคลาดเคลื่อนหรือค่าความผิดพลาด โดยโครงข่ายจะเรียนรู้ข้อมูลทั้งสองโดยการปรับค่าน้ำหนักเพื่อลดค่าความแตกต่างหรือค่าความผิดพลาดระหว่างค่าของตัวแปรเอาท์พุตของโครงข่ายกับค่าของ ข้อมูลเอาท์พุตที่ถูกต้องให้น้อยที่สุด การปรับค่าน้ำหนักจะปรับทีละน้อย ๆ โดยกระบวนการทำซ้ำ กับข้อมูลที่ละชุด จนกระทั่งค่าน้ำหนักในโครงข่ายลู่เข้า ซึ่งทั้งหมดนี้เรียกว่า การเรียนรู้ จากนั้น เมื่อเราป้อนค่าข้อมูลอินพุตล่าสุดซึ่งเป็นข้อมูลชุดใหม่ก็จะได้ค่าตัวแปร เอาท์พุตของโครงข่าย เมื่อ โครงข่ายทำการเรียนรู้แล้วก็จะป้อนข้อมูลอินพุตล่าสุดให้กับโครงข่าย เพื่อที่จะหาค่าของตัวแปร เอาท์พุตซึ่งคือ ค่าผลการทำนายหรือระบุคำหรือวลีสำคัญ เป็นต้น ซึ่งในวิทยานิพนธ์นี้จะใช้ การเรียนรู้แบบมีผู้สอนเท่านั้น

2. การเรียนรู้แบบไม่มีผู้สอน (unsupervised learning)

การเรียนรู้แบบไม่มีผู้สอนนั้นได้ถูกพัฒนาเพื่อให้ใกล้เคียงกับระบบการเรียนรู้ ของสมองมนุษย์มากยิ่งขึ้น โดยจะมีเพียงชุดข้อมูลอินพุตเท่านั้น จากนั้น กระบวนการเรียนรู้จะใช้ หลักทางสถิติ โดยหาค่าทางสถิติของชุดฝึกสอน และทำการจัดกลุ่มข้อมูลออกเป็นระดับต่าง ๆ โดยโครงข่ายประสาทเทียมจะหาค่าเอาท์พุตเองจากความสัมพันธ์ระหว่างข้อมูลอินพุตและเอาท์พุต

ประเภทของโครงข่ายประสาทเทียมจะมีกฎกำหนดวิธีการของโครงข่ายโดยแบ่งออกเป็น 5ประเภทดังนี้

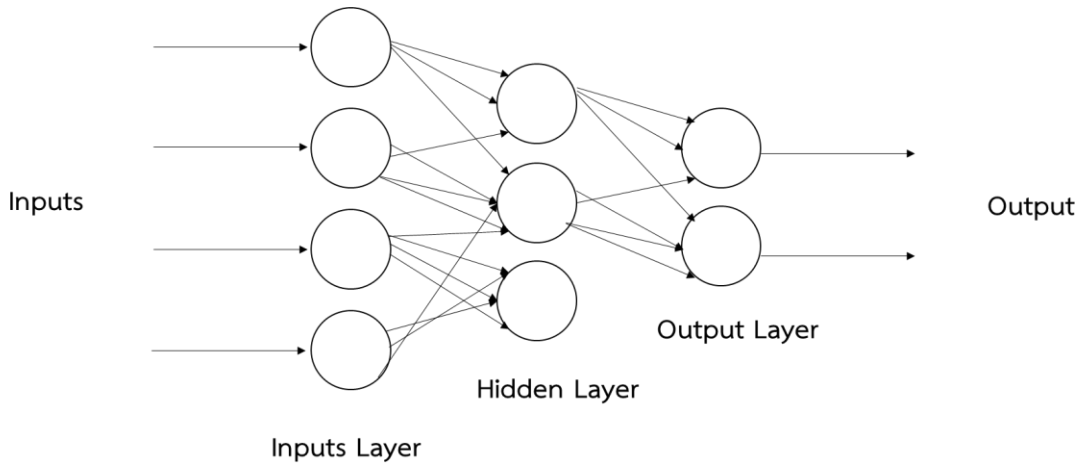
1. ประเภทการคาดเดา (Prediction)
2. ประเภทการจัดลำดับหมวดหมู่ (Classification)
3. ประเภทการเชื่อมโยงข้อมูล (Data association)
4. ประเภทกระบวนการสร้างความคิด (Data conceptualization)
5. ประเภทการกลั่นกรองข้อมูล (Data filtering)

ตาราง 2.1 แสดงความแตกต่างระหว่างประเภทของโครงข่ายประสาทเทียม

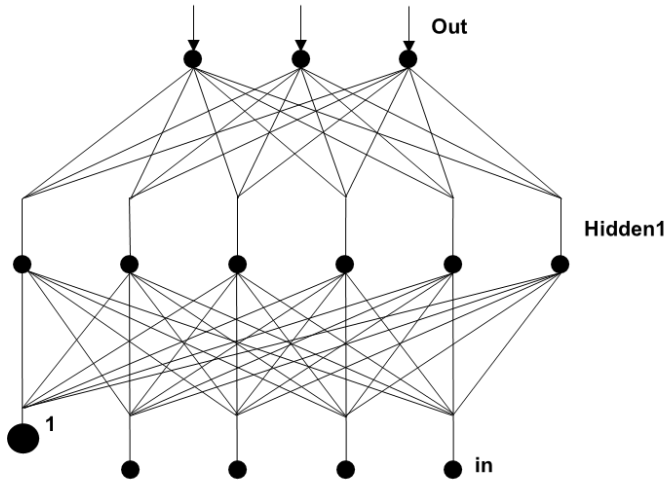
ชนิดโครงข่าย	โครงข่าย	การใช้
การคาดเดา Prediction	- Back-propagation - Delta Bar Delta - Extended Delta Bar Delta - Directed Random Search - Higher Order Neural Networkss - Self-organizing map into Back-propagation	ใช้ค่าอินพุตเพื่อคาดเดา เอาท์พุต
การจัดหมวดหมู่ Classification	- Learning Vector Quantization - Counter-propagation - Probabilistic Neural Networkss	ใช้ค่าอินพุตเพื่อกำหนดการจัดหมวดหมู่
การเชื่อมโยงข้อมูล Data Association	- Hopfield - Boltzmann Machine - Hamming Networks - Bidirectional associative Memory	เหมือนกับ Classification แต่มันจะจดจำข้อมูลที่มี error ด้วย
กระบวนการสร้างความคิด Data Conceptualization	- Adaptive Resonance - Self Organizing Map	วิเคราะห์อินพุตเพื่อการ จัดกลุ่ม
การกลั่นกรองข้อมูล Data Filtering	- Recirculation	ทำให้สัญญาณอินพุตเรียบสม่ำเสมอ

สถาปัตยกรรมโครงข่ายประสาทเทียมเพอร์เซฟตรอนแบบหลายชั้น

สถาปัตยกรรมโครงข่ายประสาทเทียมเพอร์เซฟตรอนแบบหลายชั้น หรือ โครงข่ายแบบป้อนไป ข้างหน้าโดยมีการเรียนรู้แบบย้อนกลับ นั้นถูกพัฒนาในต้นปี 1970 โดยจากหลายๆ แหล่งและเป็น การทำงานพัฒนาร่วมกันอย่างอิสระ โดยในปัจจุบันสถาปัตยกรรมแบ็ค พรอพาเกชั่น (ปัญญาประดิษฐ์) หรือแบบ แพร่ย้อนกลับนี้เป็นที่นิยมสูงสุดและยังมีประสิทธิภาพมาก รวมถึงยังมีความง่ายสำหรับการเป็น ต้นแบบสำหรับโครงข่ายประสาทเทียมที่มีความซับซ้อนมากขึ้นที่เป็นแบบหลายเลเยอร์ โดย เทคนิคการแพร่ย้อนของสถาปัตยกรรมแบบนี้นั้นได้ถูกใช้ในหลายแอพพลิเคชั่นด้วยกัน และยังมี ผลต่อชนิดของโครงข่ายขนาดใหญ่ในด้านของรูปร่างและวิธีการฝึกที่แตกต่างกันไป เพราะมีจุด แข็งที่สำคัญของเทคนิค คือ วิธีการทำงานแบบไม่เชิงเส้น (non-linear) ที่มีความเหมาะสมต่อ การแก้ปัญหาที่มีความไม่ชัดเจนโครงข่ายประสาทเทียมเพอร์เซฟตรอนแบบหลายชั้นนั้น มีลักษณะต้นแบบ คือ จะมีจำนวนหนึ่งชั้นอินพุต (Input Layer) หนึ่งชั้นเอาท์พุต (Output Layer) และอย่างน้อยหนึ่งชั้น ซ่อน (Hidden Layer) ซึ่งลักษณะของโครงข่ายประสาทเทียมนั้น ไม่มีข้อจำกัดทางทฤษฎีต่อจำนวน ของชั้นซ่อน แต่ตามแบบต้นฉบับจะมีเพียงหนึ่งชั้นหรือสองชั้นเท่านั้น โดยบางการทำงานที่ แก้ปัญหาที่ซับซ้อนจะต้องมีอย่างน้อยที่สุดสี่ชั้น (สามชั้นซ่อน กับหนึ่งชั้นเอาท์พุต) แต่ละชั้น เชื่อมต่อกับชั้นที่ตามมา ดังแสดงในภาพ



<u>ภาพที่ 2.3</u> สถาปัตยกรรมของโครงข่ายประสาทเทียม (Chester, M., 1993)



<u>ภาพที่ 2.4</u> ตัวอย่างโครงข่ายประสาทเทียมแบบ Feed forward Back-Propagation (Dave Anderson and George McNeill, 1992)

จาก (Dave Anderson and George McNeil, 1992) นั้น ชั้นอินพุตและชั้นเอาท์พุตของโครงข่ายประสาทเทียมจะเป็นการแสดงถึงการไหลของข้อมูล (Information) ในกระบวนการของการเรียกซ้ำหรือการแพร่กลับ โดยการเรียกซ้ำนั้น เป็นกระบวนการนำข้อมูลอินพุตสู่โครงข่าย ที่ได้รับการฝึกสอนแล้ว และรอรับคำตอบที่เอาท์พุตโดยการเปรียบเทียบค่าความแตกต่างหรือ ค่าความผิดพลาดจากการแพร่ย้อนกลับในช่วงของการเรียกซ้ำ โดยจะใช้ในขณะที่โครงข่ายกำลัง เรียนรู้จากข้อมูลชุดฝึกสอน จำนวนชั้นและจำนวนโหนดในแต่ละชั้นนั้นมีผลต่อการทำงานของโครงข่ายแบบป้อนไปข้างหน้าและมีการเรียนรู้แพร่ย้อนกลับ โดยสิ่งเหล่านี้เป็นสิ่งที่ ละเอียดอ่อนมาก และเป็นศิลปะของนักออกแบบโครงข่าย โดยไม่มีคำตอบที่ตายตัวหรือแน่นอน สำหรับการออกแบบโครงข่ายจะมีเพียงกฎทั่วไปที่นักวิจัยและวิศวกรส่วนใหญ่หยิบยกขึ้นมาและ ทำตามกฎนั้นดังต่อไปนี้

กฎที่ 1 เมื่อข้อมูลอินพุตและเอาท์พุตที่ต้องการมีความสัมพันธ์ที่ซับซ้อนมากขึ้น จำนวนของหน่วยประมวลผลในชั้นซ่อนก็ควรจะมากขึ้นด้วย

กฎที่ 2 ถ้ากระบวนการสามารถแยกเป็นหลายขั้นตอนได้ ก็ต้องเพิ่มชั้นซ่อนแต่ถ้ากระบวนการ ไม่สามารถแยกเป็นขั้นตอน ได้ชั้น ที่เพิ่มเข้าไปอาจทำให้สามารถจดจำอย่างง่ายได้ แต่จะไม่ใช่วิธีการ ธรรมดา

กฎที่ 3 ปริมาณข้อมูลการฝึกสร้างขอบเขตบนของหน่วยประมวลผลในชั้นซ่อน

ลักษณะโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ หรือแบบเพอร์เซฟตรอนหลายชั้นซึ่ง มีลักษณะหลักๆ ดังนี้คือ

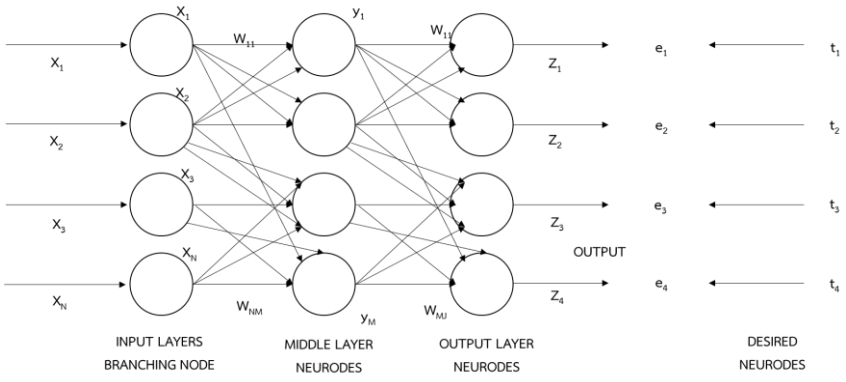
1. จำนวนชั้นต่างๆ โครงข่ายประสาทเทียมจะประกอบด้วยชั้นต่างๆ คือ ชั้นอินพุต ชั้นเอาท์พุต และชั้นซ่อนซึ่งจะอยู่ระหว่างชั้นอินพุตและชั้นเอาท์พุต

2. การเชื่อมต่อระหว่างชั้นต่างๆ นั้น จะมีลักษณะที่ทุกๆ โหนดในชั้นอินพุตนั้นจะ ส่งสัญญาณไปยังทุกๆ โหนดในชั้นซ่อน และทุกๆ โหนดในชั้นซ่อนจะทำการส่งสัญญาณต่อไปยัง โหนดในชั้นเอาท์พุต

3. การทำงานของชั้นต่างๆ นั้น ในชั้นอินพุตจะไม่มีการประมวลผลใด ๆ ทั้งสิ้น จะทำหน้าที่เพียงแค่รับสัญญาณหรือข้อมูลเข้าแล้วกระจายออกไปยังแต่ละโหนดในชั้นซ่อน ส่วน ชั้นซ่อนและชั้นเอาท์พุตนั้น จะเป็นชั้นที่มีการประมวลผล โดยภาพที่ 1.6 จะเป็นการแสดงถึง ลักษณะของโครงข่ายประสาทเทียมแบบเพอร์เซพตรอนแบบหลายชั้น ซึ่งจะประกอบไปด้วยชั้น อินพุต ชั้นซ่อนจำนวน 1 ชั้น และชั้นของเอาท์พุต โดยแต่ละโหนดจะถูกเชื่อมต่อกันเป็นโครงข่าย

วิธีการเรียนรู้แบบแพร่ย้อนกลับ

โครงข่ายแบบเพอร์เซพตรอนแบบหลายชั้นมีรายละเอียดดังนี้ ดังในภาพ



<u>ภาพที่ 2.5</u> สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

(Winston, Patrick Henry 1992)

ความหมายของตัวแปรต่างๆที่ใช้ แสดงตัวแปลต่างๆ พร้อมความหมาย ดังแสดงในภาพ

x_n	= อินพุตโหนดที่ n มีทั้งหมด N โหนด
s_m	= เอาท์พุตของชั้นซ่อน ก่อนทำการปรับค่า (activation) เป็น y_m
y_m	= เอาท์พุตของชั้นซ่อน หลังทำการปรับค่าของโหนดที่ m มีทั้งหมด M โหนด
v_j	= เอาท์พุตของชั้นเอาท์พุต ก่อนทำการปรับค่า (activation) เป็น z_j
z_j	= ค่าเอาท์พุตที่ได้ทำการปรับค่าแล้วของชั้นเอาท์พุตโหนดที่ j มีทั้งหมด J โหนด
t_j	= ค่าเอาท์พุตที่ต้องการที่ชั้นเอาท์พุตโหนดที่ j มีทั้งหมด J โหนด
w_{nm}	= น้ำหนักของเส้นเชื่อมระหว่างชั้นอินพุต กับชั้นซ่อน
w_{mj}	= น้ำหนักของเส้นเชื่อมระหว่างชั้นซ่อน กับชั้นเอาท์พุต η
η	= อัตราการเรียนรู้มีค่าอยู่ระหว่าง 0 ถึง 1
r	= จำนวนรอบที่จะทำการเรียนรู้ มี R เป็นจำนวนรอบที่กำหนด
q	= จำนวนชุดของข้อมูลตัวอย่าง มี Q เป็นตัวกำหนด
$e^{(q)}$	= ค่าผิดพลาดของข้อมูลตัวอย่าง
E	= ค่าผิดพลาดรวมเฉลี่ยของข้อมูลตัวอย่าง

ภาพที่ 2.6 ตัวแปลต่างๆ พร้อมความหมาย (Winston, Patrick Henry 1992 : 443-469)

ขั้นตอนการเรียนรู้ของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

1. กำหนดจำนวนโหนดในชั้นอินพุต (N), จำนวนโหนดในชั้นเอาท์พุต (I), จำนวนโหนดในชั้นซ่อน (M) จำนวน 1 ชั้นซ่อน และกำหนดจำนวนข้อมูลอินพุตและข้อมูล เอาท์พุต ต่อจากนั้นจะทำการกำหนดจำนวนรอบสูงสุดที่จะทำการเรียนรู้ (R) รวมถึงค่าผิดพลาดที่ ยอมรับได้

2. กำหนดค่าพารามิเตอร์ของอัตราการเรียนรู้ (η) ที่อยู่ในช่วง [0, 1]

3. การลุ่มน้ำหนักเริ่มต้นให้กับทุกๆเส้นเชื่อมโยงภายในโครงข่ายประสาท เทียมในทั้ง 2 ชั้น โดยให้มีค่าอยู่ระหว่าง [-1,1]

4. รับค่าอินพุตของข้อมูลชุดแรกหรือข้อมูลแถวแรก เพื่อใช้ในการคำนวณหา ค่าเอาท์พุตของโครงข่ายประสาทเทียม

5. คำนวณค่าผลรวมของโหนดในชั้นซ่อน (*S*) ก่อนทำการปรับค่ารวมด้วยฟังก์ชันกระตุ้น (Activation Function) ฟังก์ชันซิกมอยด์ ซึ่งจะได้ค่าของโหนดในชั้นซ่อนที่อยู่ ในช่วง [0, 1] โดยมีรายละเอียดดังสมการ 1.7.1-1 ถึง 1.7.1-3

ค่าเอาท์พุตของชั้นซ่อนก่อนทำการปรับค่า ดังสมการที่ 1.7.1-1

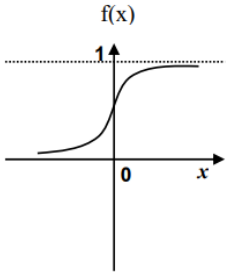
$$s_m = \sum_{n=1}^{N} x_n * w_{nm} \qquad \text{...(1.7.1-1)}$$

ค่าเอาท์พุตของชั้นซ่อนหลังทำการปรับค่า ดังสมการที่ 1.7.1-2

$$y_m = f(s_m) \qquad \text{...(1.7.1-2)}$$

ฟังก์ชันที่ใช้ปรับค่า ดังสมการที่ 1.7.1-3

$$f(x) = \frac{1}{1+e^{-x}} \qquad \text{...(1.7.1-3)}$$



c) SIGMOID

<u>ภาพที่ 2.7</u> แสดงค่าที่ได้จากฟังก์ชันซิกมอยด์

6. คำนวณค่าเอาท์พุตของโหนดในชั้นเอาท์พุตด้วยสมการที่ 1.1.7-4 จากนั้นทำ การปรับค่าผลรวมด้วยฟังก์ชันกระตุ้นฟังชันซิกมอยด์ ดังสมการที่ 1.7.1-3 ซึ่งจะได้ค่าของโหนดในชั้นเอาท์พุตที่อยู่ในช่วง [0, 1] สำหรับค่าของผลลัพธ์ในโหนดของชั้นเอาท์พุตนั้น หลังทำการปรับค่า แสดงดังสมการที่ 1.1.7-5

$$v_j = \sum_{m=1}^{M} y_m * w_{m_j} \qquad \text{...(1.1.7-4)}$$

ค่าเอาท์พุตของชั้นเอาท์พุตหลังทำการปรับค่า ดังสมการที่ 1.7.1-5

$$z_j = f(v_j) \qquad \text{...(1.7.1-5)}$$

7. การหาค่าความผิดพลาดของโหนดในชั้นเอาท์พุตและทำการปรับน้ำหนัก นั้น โดยการนำเอาท์พุตที่คำนวณได้จริงเปรียบเทียบกับเอาท์พุตที่ได้กำหนดไว้ เพื่อหาค่าความ ผิดพลาดของข้อมูล โดยถ้าค่าผิดพลาดของข้อมูลน้อยกว่าค่าผิดพลาดที่ยอมรับได้แล้วนั้น โครงข่ายประสาทเทียมก็จะทำการรับข้อมูลชุดต่อไปเข้าสู่โครงข่าย โดยถ้าไม่ได้ทำการปรับ น้ำหนักแล้วนั้นโครงข่ายประสาทเทียมจะทำการรับข้อมูลแถวถัดไป และจะกลับไปทำ ข้อ 5 แต่ถ้าเป็นข้อมูลชุดสุดท้ายของข้อมูลแล้วนั้น จะไปทำข้อ 8

ค่าความผิดพลาดในแต่ละแถวของข้อมูล ดังสมการที่ 1.7.1-6

$$e^q = \frac{1}{2} \sum_{j=1}^{j} \left(t_j - z_j\right)^2 \qquad \text{...(1.7.1-6)}$$

การปรับน้ำหนักระหว่างโหนดในชั้นซ่อนและชั้นเอาท์พุต ดังสมการที่ 1.7.1-7

$$w_{nm}^{\,r+1} = w_{mj}^{\,r} + \eta \{ t_j^{\,q} - z_j^{\,q} * [z_j^{\,q} (1 - z_j^{\,q}) * y_m^{\,q} \} \qquad \text{...(1.7.1-7)}$$

การปรับน้ำหนักระหว่างโหนดในชั้นอินพุตและชั้นซ่อน มีสมการที่ 1.7.1-8

$$w_{nm}^{\,r+1} = w_{nm}^{\,r} + \eta \{ \sum {}^{j}{}_{j=1} (t_j^{\,q} - z_j^{\,q}) [z_j^{\,q} (1 - z_j^{\,q}) * w_{mj}^{\,r} \} * [y_m^{\,q} (1 - y_m^{\,q}) [x_n^{\,q}] \qquad \text{..(1.7.1-8)}$$

8. คำนวณค่าผิดพลาดรวมเฉลี่ย (Mean Squared Eror : MSE) ในทุก ๆ แถว ข้อมูล โดยนำค่าผิดพลาดของแต่ละแถวของชุดข้อมูลมารวมกัน แล้วทำการหาค่าเฉลี่ย เพื่อใช้ใน การตรวจสอบว่าผลลัพธ์ของทุกๆ ข้อมูลในแต่ละรอบนั้นมีค่าน้อยกว่าค่าผิดพลาดที่ยอมรับได้ใน ทุกๆ แถวข้อมูลหรือไม่ โดยถ้าค่าความผิดพลาดยอมรับได้ให้จบการเรียนรู้ แต่ถ้าค่าความผิดพลาด เฉลี่ยมากกว่าค่าที่ยอมรับได้ให้ตรวจสอบว่าได้ทำการเรียนรู้ครบตามจำนวนรอบที่กำหนดไว้ หรือไม่ ถ้าครบ

แล้วให้จบการเรียนรู้ แต่ถ้ายังไม่ครบให้กลับไปทำข้อ 6.3.2.4 ใหม่ ซึ่งก็คือเริ่มต้น การเรียนรู้รอบใหม่ หาค่าผิดพลาดรวมเฉลี่ย ดังสมการที่ 1.7.1-9

$$E=\frac{1}{Q}\sum_{q=1}^{Q} e^{(q)} \qquad ...(1.7.1\text{-}9)$$

ข้อดีและข้อเสียของการเรียนรู้แบบแพร่กลับ (Dayhoft, 1990)

ข้อดี คือ การเรียนรู้แบบแพร่กลับนั้นจะมีความสามารถในการจดจำรูปแบบ ซึ่ง การเรียนรู้แบบแพร่กลับสามารถที่จะเรียนรู้ความสัมพันธ์ของรูปแบบได้มากมาย โดยการเรียนรู้ แบบแพร่กลับต้องการตัวอย่างรูปแบบที่จะเรียนรู้ ความยืดหยุ่นของการเรียนรู้อยู่ที่ความ หลากหลายในการออกแบบทางเลือกต่าง ๆ เช่น จำนวนชั้น เส้นเชื่อมโยง จำนวนโหนดในแต่ละชั้น ที่ได้ทำการกำหนดขึ้น โดยความยืดหยุ่นนี้เอง ทำให้การเรียนรู้แบบแพร่กลับสามารถแก้ปัญหางาน ประยุกต์ได้อย่างมากมาย

ข้อจำกัด คือ การใช้เวลามากในการฝึกสอนโครงข่ายประสาทเทียมให้เกิด การเรียนรู้สำหรับการเรียนรู้แก้ปัญหา

หลักการประยุกต์ใช้ Artificial Neural Networks (ANNs) Model ในการพยากรณ์ปริมาณน้ำท่ารายวัน

Artificial Neural Networks (ANNs) Model เป็น Black Box ชนิด Backpropagation Algorithm ซึ่งจะสนใจความสัมพันธ์ของ Input ไปสู่ Output เป็นสำคัญ ซึ่งมี ข้อดีในการประยุกต์ใช้ได้ง่าย สะดวก และรวดเร็ว

ในการใช้งาน โดยไม่ได้เน้นข้อมูลทางด้านกายภาพ มากนัก ANNs Model สามารถนำไปประยุกต์ใช้ได้หลายแขนง แต่ในที่นี้จะกล่าวถึงการนำมา ประยุกต์ใช้ในด้านอุทกศาสตร์ โดยเป็นการทำโมเดลในการพยากรณ์ปริมาณน้ำท่ารายวัน โดยมี หลักเกณฑ์ในการนำ ANNs Model ไปประยุกต์ใช้ดังนี้ คือ

1. เลือกสถานีที่จะทำการพยากรณ์ ควรเป็นสถานีที่อยู่ในลำน้ำสายหลัก และ เป็นจุดสำคัญที่จะเตือนภัยต่อพื้นที่ด้านท้ายสถานีได้อย่างทันสถานการณ์ได้เป็นอย่างดี และมีข้อมูล เพียงพอที่จะนำมาทำการ Training และ Testing สำหรับการคัดเลือกทำโมเดลด้วย

2. แบ่งช่วงข้อมูลที่จะนำมาทำการ Training และ Testing โดยการแบ่งช่วง ข้อมูลที่จะนำมาทำการ Training และ Testing ต้องแบ่งให้ชัดเจนลงไปว่า โมเดลจะทำการพยากรณ์ ปริมาณน้ำท่าในช่วงฤดูฝนหรือฤดูแล้ง ซึ่งควรจะแยกเป็นคนละโมเดลให้ชัดเจน

3. ระยะเวลาที่จะนำมาทำโมเดล ต้องไม่ควรเกิน 10 ปีย้อนหลัง ซึ่งต้องคำนึงถึงการเปลี่ยนแปลงสภาพของพื้นที่ลุ่มน้ำ เพราะถ้าสภาพลุ่มน้ำมีการเปลี่ยนแปลงใหม่ เช่น มี การผันน้ำออกจากลำน้ำเดิม หรือมีการสร้างอาคารใหม่ขวางทางลำน้ำ ก็จะต้องใช้ข้อมูลใหม่นั้น มาทำการ Training และ Testing เพื่อคัดเลือกโมเดล โดยให้สภาพลุ่มน้ำใกล้เคียงกับสภาพปัจจุบันมาก ที่สุดนั่นเอง

4. การครอบคลุมขนาดของปริมาณน้ำท่าที่จะนำมาทำการ Training และ Testing ช่วงของข้อมูลที่จะนำมาทำการ Training ขนาดของข้อมูลปริมาณน้ำท่าต้องครอบคลุมขนาด ของปริมาณน้ำท่าช่วงของข้อมูล Testing ทั้งด้าน Upper และ Lower Boundary และจำนวนข้อมูล ที่จะนำมาทำการ Training ต้องมากกว่า Testing (จำนวนข้อมูล Testing ประมาณ 2/3 ของจำนวน ในข้อมูล Training) และข้อมูลที่จะทำการ Testing ควรจะเป็นปีที่อยู่ใกล้ปีปัจจุบันเพื่อเสริมความ มั่นใจได้ว่าโมเดลจะเหมาะสมต่อการนำไปใช้งานจริงได้อย่างแม่นยำนั่นเอง

5. การคัดเลือก Input ในการนำมาประยุกต์ใช้ในการทำโมเดลในการ พยากรณ์ปริมาณน้ำท่ารายวัน คือ ตัวปริมาณน้ำท่ารายวัน และปริมาณน้ำฝนรายวัน เป็นสำคัญ ตัวอย่างความสัมพันธ์ทั่วไปของสมการที่จะใช้ในการพยากรณ์ปริมาณน้ำท่า

$$Q1_{t+1} = f\,(Q1_t, Q2_t \ldots, R1_t, R2_{t-1}, \ldots) \qquad \ldots (1.7.1\text{-}9)$$

ภาพที่ 2.8 สมการตัวแปรสมมติในการคำนวณ

Q1, 02, คือ ปริมาณน้ำท่ารายวันที่สถานีต่าง ๆ ที่มีอิทธิพลต่อ Output (ปริมาณ น้ำท่าที่จะทำการพยากรณ์ล่วงหน้า) ที่ระยะเวลา Lag Time ต่าง ๆ

R1, R2, ... คือ ปริมาณน้ำฝนรายวันที่สถานีต่าง ๆ ที่มีอิทธิพลต่อ Output (ปริมาณ น้ำท่าที่ จะทำการ พยากรณ์ล่วงหน้า) ที่ระยะเวลา Lag Time ต่าง ๆ

I คือ ที่เวลาวันปัจจุบัน

Lag Time ของ Input ที่จะนำมาเป็นตัว Input เข้าสู่โมเดล ดูได้จากระยะทางที่ สถานีน้ำฝน และสถานีน้ำท่าเดินทางมาถึงสถานีน้ำท่าที่จะทำการพยากรณ์นั่นเอง ขึ้นอยู่กับ สภาพ ของลุ่มน้ำนั้น ๆ และดูได้จากค่า Correlation Coefficient ประกอบกันด้วย ซึ่งตัว Input ที่จะ นำมาใช้ในการป้อนเข้าสู่โมเดลควรเป็นตัวที่มีความสัมพันธ์กับตัว Output เป็นอย่างมาก ทำได้ด้วย การหาค่า Correlation Coefficient ระหว่าง Input กับ Output แล้วนำค่าที่มี Correlation Coefficient ที่สูงมาเป็นตัว Input นั่นเอง โดยควรเลือกสถานีที่มีผลต่อ Output ซึ่งดูได้จากสภาพลุ่มน้ำว่าข้อมูลที่มีการบันทึกไว้นั้นเอื้ออำนวยมากน้อยเพียงใด และควรให้ Input ที่ใช้ในโมเดลมีความ กระชับ สะดวกรวดเร็ว ในการนำไปใช้งาน ซึ่งไม่ควรเกิน 5 ตัวแปร

6. ทำการ Training และ Testing โดยทำการคัดเลือก Input แบบต่าง ๆ เพื่อ คัดเลือกชุดข้อมูลที่ดีที่สุดเป็นโมเดลในการใช้งาน ซึ่งในการเลือกโมเดลที่ดีที่สุดนั้น ดูได้จากตัว Testing ได้ดังนี้คือ ดูจากค่า RMSE และ ดูจากไฮโดกราฟ (Visual Inspection) ประกอบกันเป็นหลัก นั่นเอง

7. การนำไปใช้งานจริง เมื่อได้โมเดลเป็นที่เรียบร้อยแล้ว ตอนนำโมเดลไปใช้ งานจริงในการพยากรณ์ในแต่ละครั้งจะให้ค่า Error เกิดขึ้น ดังนั้นควรนำ Error ที่เกิดขึ้นในแต่ละวัน ไปปรับแก้ผลจากการพยากรณ์ในครั้งต่อไป ด้วยเทคนิคต่าง ๆ ขึ้นอยู่กับผู้นำไปใช้งานนั่นเอง

8. ทำการ Update โมเดล เมื่อใช้งานไป 3-5 ปี เพื่อรักษาสภาพลุ่มน้ำให้ ใกล้เคียงกับที่เป็นอยู่ปัจจุบันให้มากที่สุดนั่นเอง

2.1งานวิจัยที่เกี่ยวข้อง

ทวีศักดิ์ วงไพศาล (2558) ได้ศึกษาปัจจัยที่มีผลต่อความแม่นยำในการพยากรณ์ระดับน้ำหลากที่สถานีวัดระดับน้ำ การศึกษาครั้งนี้พิจารณาถึงการกำหนดจำนวนหน่วยในชั้นซ่อนและการกำหนดข้อมูลนำเข้าที่ให้คู่การพยากรณ์ที่มีความแม่นยำ โดยใช้กระบวนการเรียนรู้ด้วยวิธี Levenberg-Marquardt (LM) และวิธี BayesianRegularization (BR) เป้าหมายการศึกษาคือระดับน้ำ2 หลากที่สถานีวัดระดับน้ำ M.7 ในแม่น้ำมูล จังหวัดอุบลราชธานี ทำการพยากรณ์ระดับน้ำล่วงหน้า 3 วัน ผลการศึกษาพบว่า ผลโดยรวมที่ทำให้แบบจำลองมีความแม่นยำได้แก่ (1) จำนวนหน่วยในชั้นซ่อนที่เหมาะสมมีค่าเท่ากับ 0.5 เท่าของจำนวนหน่วยในชั้นนำข้อมูลเข้า (2) กระบวนการเรียนรู้ด้วยวิธี LM ให้ผลที่ดีกว่าวิธี BR และ (3) การพยากรณ์ระดับน้ำของสถานีวัดระดับน้ำที่ทำให้ได้ผลการจำลองที่แม่นยำ ควรมีข้อมูลนำเข้าที่เป็นระดับน้ำของสถานีทางด้านเหนือน้ำของสถานีวัดระดับน้ำที่ต้องการพยากรณ์ที่อยู่ในแม่น้ำสายหลักที่ใกล้ที่สุด และข้อมูลระดับน้ำของสถานีวัดระดับน้ำที่ต้องการพยากรณ์

ยุพิน และ คณะ (2560) ได้ใช้โครงข่ายประสาทเทียมเพื่อคาดการณ์น้ำท่วมในเขตเทศบาลนครเชียงใหม่ ช่วงเดือนกรกฎาคม ถึง กันยายน พ.ศ. 2537-2549 ณ สถานีวัดระดับน้ำ P.1 ณ สะพานนวรัฐ ที่ได้รับผลกระทบจากการเปลี่ยนแปลงสภาพภูมิอากาศ โดยใช้ข้อมูลน้ำฝน รายวันจากแบบจำลอง WRF-ECHAM5 ที่มีขนาดกริดน้ำฝน 20*20 กิโลเมตร เป็นข้อมูลนำเข้าโดยครอบคลุมพื้นที่ศึกษา ทั้งหมด 6 กริด ซึ่งแบบจำลองโครงข่ายประสาทเทียมจะใช้กระบวนการเรียนรู้แบบ LM (Levenberg Marquardt) มีจำนวนโหนดในชั้นซ่อนเร้น 15 โหนด (จำนวนร้อยละ50จากจำนวนข้อมูลนำเข้า) โดย เลือกใช้เหตุการณ์น้ำท่วมระหว่างปี พ.ศ. 2548-2549 เป็นเหตุการณ์ในการเรียนรู้ผลการศึกษาพบว่า การคาดการณ์พบว่าจะมีเหตุการณ์น้ำท่วมเกือบทุกปีรวมทั้งสิ้น 67 เหตุการณ์โดยมีเหตุการณ์น้ำท่วม ที่มีระดับน้ำสูงกว่า 5 เมตร จำนวน 13 เหตุการณ์โดยในปี พ.ศ. 2549 เป็นปีที่มีระดับน้ำท่วมสูงสุด คือ 5.57 เมตร และปริมาณน้ำฝนที่ส่งผลต่อเหตุการณ์น้ำท่วมคือมีปริมาณน้ำฝนตกมากกว่า 100 มิลลิเมตร ในพื้นที่ศึกษาโดยเฉพาะกริดที่ 1 ที่ครอบคลุมพื้นที่ในเขตเทศบาลนครเชียงใหม่

รติพร จันทร์กลั่น (2560) ได้สร้างแบบจำลองด้วยเทคนิคการเรียนรู้ของเครื่องเพื่อคาดการณ์ปริมาณน้ำท่า โดยงานวิจัยนี้ได้เสนอการพัฒนาวิธีการคาดการณ์ปริมาณน้ำท่าด้วยวิธีการ ANN-GS ซึ่งเป็นเทคนิคการผสมผสานของวิธีการเรียนรู้ 2 อัลกอริทึม ได้แก่ โมเดลที่ถูกพัฒนาจาก

วิธีการ วิเคราะห์การถดถอย คือโมเดลเชิงเส้นโดยนัยทั่วไป (Generalized Linear Model: GLM) และ ซัพพอร์ตเวกเตอร์รีเกรสชัน จากนั้นนำผลการคาดการณ์ของทั้งสองโมเดลมาใช้ในการเรียนรู้ของโครงข่ายประสาทเทียมเพื่อใช้สร้างโมเดลในการคาดการณ์ปริมาณน้ำท่าที่มีประสิทธิภาพ การทดสอบประสิทธิภาพของ ANN-GS จะเปรียบเทียบกับโครงข่ายประสาทเทียม การ ถดถอยเชิงเส้น อาริม่า โมเดลเชิงเส้นโดยนัยทั่วไป และซัพพอร์ตเวกเตอร์รเกรสชัน โดยข้อมูลที่ใช้ ประสิทธิภาพในการคาดการณ์น้ำท่าโดยใช้มาตรวัดค่ารากที่สองของความคลาดเคลื่อนกำลังสองเฉลี่ยแสดงให้เห็นว่า ANN-GS มีความผิดพลาดต่ำสุดเมื่อเทียบกับอัลกอริทึมอื่น ๆ

ธิษณ์ปัณฑา คนโทฉิมพลี และคณะ (2562) ได้ศึกษาการประยุกต์ใช้โครงข่ายใยประสาทเทียมในการพยากรณ์ปริมาณน้ำไหลเข้าอ่างเก็บน้ำที่สำคัญในจังหวัดระยอง โดยทำการศึกษาความสัมพันธ์ของข้อมูลปริมาณน้ำฝนที่ใช้ในการพยากรณ์จำนวนทั้งหมด 4 สถานี และโครงสร้างแบบจำลอง ANN ที่เหมาะสมในแต่ละอ่างเก็บน้ำ โดยทดสอบจำนวนเซลล์ของชั้นที่ซ่อนตั้งแต่ 2-10, 0.5n, n, 2n และ 5n เซลล์เมื่อ n คือจำนวนของตัวแปรนำเข้า สำหรับการการพยากรณ์ปริมาณน้ำไหลเข้าอ่างเก็บน้ำรายวันได้เพิ่มค่าเฉลี่ยเคลื่อนที่ (Moving Average) 3 วัน เพื่อศึกษาหาจำนวนวันที่สามารถพยากรณ์ปริมาณน้ำไหลเข้าอ่างเก็บน้ำล่วงหน้าในแต่ละอ่างเก็บน้ำ โดยจะใช้ช่วงเวลาของข้อมูลนำเข้าที่แตกต่างกันทั้ง 3 ช่วงเวลาพยากรณ์ ผลการพัฒนาแบบจำลองโครงข่ายใยประสาทเทียมแบบแพร่กระจายย้อนกลับ (Back Propagation) พบว่า ทั้ง 4 อ่างเก็บน้ำสามารถพยากรณ์ปริมาณน้ำไหลเข้าอ่างเก็บน้ำและให้ผลเป็นที่ยอมรับได้ล่วงหน้า 3 วัน 1 สัปดาห์ และ 1 เดือน ซึ่งจากผลการศึกษาดังกล่าว แสดงให้เห็นว่าแบบจำลองโครงข่ายใยประสาทเทียมซึ่งเป็นแบบจำลองที่อาศัยความสัมพันธ์ของสถิติข้อมูลในอดีต สามารถให้ผลการพยากรณ์ที่มีความน่าเชื่อถือและนำไปใช้ในการพยากรณ์เพื่อช่วยในการปฏิบัติงานจริงได้อย่างมีประสิทธิภาพ

วิรัช กองสิน (2562) ได้ศึกษาระบบควบคุมปริมาณน้ำให้กับพืชระยะสั้นด้วยโครงข่ายประสาทเทียมเพื่อเพิ่มประสิทธิภาพในการใช้น้ำในภาคการเกษตรในบางพื้นที่การใช้น้ำมีความจำเป็นมากแต่ปริมาณน้ำมีอย่างจำกัด งานวิจัยนี้จะทดลองกับผักคะน้า ซึ่งเป็นพืชระยะสั้นที่จังหวัดสุพรรณบุรี นิยมเพาะปลูกและบริโภคกันอย่างกว้างขวางในหลายอำเภอ จึงมีแนวคิดในการลดอัตราการใช้น้ำในการปลูกผักคะน้าเพื่อเพิ่มผลผลิต ในการทดลองแปลงแรกใช้ระบบให้น้ำแบบตั้งเวลาและแปลงที่สองใช้ระบบให้น้ำแบบโครงข่ายประสาทเทียม จากการเปรียบเทียบปริมาณน้ำรวมจากแบบ

โครงข่ายประสาทเทียมจะใช้น้ำน้อยกว่าแบบระบบตั้งเวลาจากการทดลองทำให้ทราบว่าแบบโครงข่ายประสาทเทียมสามารถลดปริมาณการใช้น้ำในการปลูกผักได้

พรรณปพร บุญแปง และคณะ (2563) ได้ศึกษาความเป็นไปได้สำหรับการคาดการณ์แผ่นดินไหวในประเทศไทยด้วยแบบจำลองโครงข่ายประสาทเทียม โดยมีขอบเขตของการศึกษา 3 ประเด็น คือ รูปแบบการเกิดแผ่นดินไหวของประเทศไทยศึกษา หาตัวแปรนำเข้าที่เหมาะสมสำหรับการคาดการณ์แผ่นดินไหว และศึกษาการออกแบบโครงสร้างของแบบจำลองโครงข่ายฯที่เหมาะสม ซึ่งพบว่า

1. รูปแบบการเกิดแผ่นดินไหวของประเทศไทยมีสาเหตุหลักจากการเคลื่อนตัวของแนวรอยเลื่อนมีพลัง

2. ตัวแปรที่ได้จากสมการความสัมพันธ์ Gutenberg-Richter ถือเป็นตัวแปรนำเข้าแบบจำลองที่มีการศึกษามากที่สุด โดยตัวแปรการดาดการณ์ที่นิยมศึกษามากที่สุด คือ การดาดการณ์ขนาดแผ่นดินไหว

3. การออกแบบโครงสร้างของแบบจำลองนิยมใช้โครงข่ายแบบ Feed Forward Neural Networks ที่มีการเรียนรู้แบบ Back Propagation โดยจำนวนชั้นซ่อนเร้นที่ทำให้ผลลัพธ์การดาดการณ์แผ่นดินไหวมีความแม่นยำมากที่สุด คือ จำนวน 2 ชั้น สำหรับจำนวนโหนดของชั้นซ่อนเร้นพบว่าขึ้นอยู่กับจำนวนตัวแปรข้อมูลนำเข้าของแบบจำลองโครงข่ายฯ

ทศพล ภูผิวฟ้า (2565) ได้สร้างแบบจำลองพารามิเตอร์บ่งรูปร่างของการแจกแจงค่าสุดขีดนัยทั่วไปจากข้อมูลน้ำฝน-น้ำท่า และข้อมูลภาพถ่ายดาวเทียมโดยใช้วิธีโครงข่ายประสาทเทียม โดยมี วัตถุประสงค์หลักของงานวิจัยนี้คือการหาตัวแปรที่ส่งผลต่อการเปลี่ยนแปลงพารามิเตอร์บ่งรูปร่าง (Shape parameter) โดยพิจารณาจากค่าสัมประสิทธิ์สหสัมพันธ์ (Correlation Coefficient) พร้อมทั้งสร้างแบบจำลองพารามิเตอร์บ่งรูปร่างภายใต้การแจกแจงค่าสุดขีดนัยทั่วไป (Generalized Extreme Value: GEV) เพื่อนำผลการศึกษามาคาดการณ์พื้นที่เสี่ยงต่อการเกิดอุทกภัยน้ำท่วมด้วยแผนที่ระดับการเกิดซ้ำ (Return level) ทั้งนี้ผู้วิจัยได้รวบรวมข้อมูลดาวเทียม ข้อมูลอุตุนิยมวิทยาและข้อมูลอุทกวิทยา ของลุ่มน้ำชี ทั้งหมด 11 ปีย้อนหลัง ในการศึกษานี้เราทำการประมาณค่าพารามิเตอร์บ่ง รูปร่างด้วยแบบจำลองกระบวนการคงที่ (Stationary process) ด้วยวิธีภาวะน่าจะเป็นสูงสุด (Maximum tikelihood estimation: MLE) และแบบจำลองกระบวนการไม่คงที่ (Non-stationary process) ด้วยวิธีโครงข่ายประสาทเทียม (Artificial neural networks: ANN) ภายใต้

การแจกแจง GEV และทำการเปรียบเทียบประสิทธิภาพของแบบจำลอง ทั้ง 2 รูปแบบด้วยค่าสัมประสิทธิ์ NashSutclife (Nash-Sutcliffe Efficiency: NSE) จากการศึกษาพบว่า แบบจำลองพบว่ามีค่า NSE สูงกว่า ทำให้สรุปได้ว่า การพัฒนาแบบจำลองแบบไม่คงที่ (Non-stationary Process) จึงเป็นแบบจำลองที่เหมาะสมกับสภาพแวดล้อมของโลกที่มีการเปลี่ยนแปลงตลอดเวลา

Imen Aichouri และคณะ (2015) ศึกษาการคาดการณ์ของน้ำท่าสำหรับลุ่มน้ำในพื้นที่กึ่ง แห้งแล้งในแถบชายฝั่งทะเลเมดิเตอร์เรเนียนโดยใช้ ANN แบบเพอร์เซ็ปตรอนหลายชั้น (Multilayer Perceptron Networks: MLPN) ซึ่งเป็นอัลกอริทึมที่ใช้ได้กับเหตุการณ์ที่เป็นเชิงเส้นและไม่เป็นเชิง เส้น โดยไม่จำเป็นต้องใช้สมมติฐานใดๆ เปรียบเทียบผลกับการถดถอยเชิงเส้น (Multiple Linear Regression) โดยใช้ข้อมูลอินพุตเป็นปริมาณน้ำท่าและน้ำฝนรายวันย้อนหลัง 1- 7 วัน ทำการวัดประสิทธิภาพด้วยค่าเฉลี่ยความผิดพลาดกำลังสอง (Average Squared of Error: ASE), ค่า สัมประสิทธิ์การตัดสินใจ (Coefficient of Determination: R), ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย (Mean Absolute Relative Error: MARE) ผลการศึกษาแสดงให้เห็นว่า ANN มีประสิทธิภาพในการ คาดการณ์น้ำท่าดีกว่าวิธีการถดถอยเชิงเส้น

Gokcen Uysal และคณะ (2016) ได้ศึกษาการคาดการณ์ข้อมูลน้ำท่าในพื้นที่ที่มีหิมะปกคลุม โดยใช้ข้อมูลในช่วงเดือนที่เป็นต้นฤดูใบไม้ผลิต่อเนื่องไปถึงปลายฤดูร้อน (1 มีนาคม - 30 มิถุนายน) เพื่อใช้ในการพิจารณาบริหารจัดการทรัพยากรน้ำโดยเฉพาะอย่างยิ่งการเฝ้าระวังน้ำท่วม การคาดการณ์น้ำท่าใช้ข้อมูลน้ำฝน อุณหภูมิเฉลี่ยรายวัน และข้อมูลจากดาวเทียม MODIS บริเวณพื้นที่หิมะปกคลุม (Snow Cover Area: SCA) ซึ่งเป็นข้อมูลจากกราฟ (Snow Depletion Curves, SCD) ที่เวลาย้อนหลัง 1 วัน เปรียบเทียบโมเดล ANN แบบเพอร์เซ็ปตรอนหลายชั้น (Multilayer Perceptron Networks: MLPN) และแบบเรเดียลเบสิสฟังก์ชัน (Radial Basis Function: RBF) วัดประสิทธิภาพ โดยใช้ค่าสัมประสิทธิ์การตัดสินใจ (Coefficient of determination: R) ค่าความผิดพลาดเฉลี่ย (Mean Error: ME) ค่ารากที่สองของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (Root Mean Squared error: RMSE) และค่าคลาดเคลื่อนสัมบูรณ์ (Mean Absolute Error: MAE) ผลการทดลอง สรุปว่า ANNs แบบเพอร์เซ็ปตรอนหลายชั้นนั้นมีประสิทธิภาพดีกว่าแบบเรเดียลเบสิสฟังก์ชัน

Alireza Sharifi และคณะ (2017) ได้ศึกษาการทำนายน้ำท่ารายวันโดยใช้อัลกอริทึมเชิงเส้นและ ไม่เป็นเชิงเส้น ใช้ข้อมูลน้ำฝนและน้ำท่าย้อนหลัง ทำการเลือกข้อมูลย้อนหลังตั้งแต่ 1 ถึง 4 วัน การ สร้างโมเดล ใช้ 3 วิธีการในการคัดเลือกข้อมูลนำเข้าได้แก่ Gamma test, Forward

selection และ Factor analysis ทีมวิจัยนี้เลือกใช้อัลกอริทึม ANN SVM LR และ Adaptive Neuro-Fuzzy Inference System (ANFIS) จากนั้น ทำการเปรียบเทียบประสิทธิภาพอัลกอริทึมด้วย R^2NS และ RMSE ผล การทดลองแสดงให้เห็นว่าการใช้โมเดล SVM กับข้อมูลที่คัดเลือกด้วย Gamma Test ให้ ประสิทธิภาพดีที่สุดในการคาดการณ์น้ำท่ารายวัน

บทที่ 3

วิธีการดำเนินงานวิจัย

3.1 ขั้นตอนการดำเนินงาน

3.1.1 ขั้นตอนเตรียมข้อมูลสำหรับที่จะใช้ในการพยากรณ์

(1) เลือกสถานีที่จะทำการพยากรณ์ปริมาณน้ำท่ารายวัน โดยมีเกณฑ์ในการเลือกสถานีดังนี้

- สถานีที่จะพยากรณ์ปริมาณน้ำท่ารายวันควรจะเป็นสถานีที่สำคัญและเป็นตัวแทนของลำน้ำในลุ่มน้ำย่อยและลุ่มน้ำหลัก
- สถานีที่เลือกมาทำการพยากรณ์ควรเป็นสถานีที่มีความมั่นคงในเรื่องตำแหน่งที่ตั้ง ไม่มีการเคลื่อนย้ายสถานี มีการจัดเก็บข้อมูล ที่มีความต่อเนื่อง มีช่วงระยะเวลาการเก็บข้อมูลที่ยาวนานต่อเนื่อง และไม่ขาดหาย

(2) การเลือกสถานีวัดปริมาณน้ำฝนรายวัน และสถานีวัดปริมาณน้ำท่ารายวันในพื้นที่ที่เลือกเฉพาะสถานีสำรวจข้อมูลที่มีอิทธิพล กับสถานีวัดปริมาณน้ำท่าที่จะทำการพยากรณ์ได้ โดยทั่วไปมีหลักเกณฑ์ดังนี้

- ใช้ลักษณะทางกายภาพของลุ่มน้ำ
- ใช้หลักความสัมพันธ์ของข้อมูลทางคณิตศาสตร์ โดยใช้ค่าสัมประสิทธิ์สหสัมพันธ์ (Correlation Coefficient) ซึ่งการวิเคราะห์ค่าสัมประสิทธิ์เทียบกับระยะเวลาย้อนหลัง ของสถานีแต่ละสถานีมีอิทธิพลตามค่าสัมประสัมประสิทธิ์สหพันธ์กับสถานีที่จะทำการพยากรณ์ เพื่อหาระยะเวลาย้อนหลังเป็นจำนวนวันถึงเท่าไหร่ ที่มีอิทธิพลต่อการเกิดปริมาณตะกอนที่วันปัจจุบันที่สถานีที่ทำการพยากรณ์

(3) การกำหนดหน่วยนำเข้าแบบจำลองระบบโครงข่ายประสาทเทียม มีขั้นตอนดังนี้

- จำนวนหน่วยในชั้นนำเข้า (Input) ประกอบด้วยข้อมูลปริมาณน้ำท่ารายวันและข้อมูลปริมาณน้ำฝนรายวันของสถานีต่างๆที่มีอิทธิพลเทียบกับ

ระยะเวลาย้อนหลัง (Lag time)กี่วันรวมกับข้อมูลปริมาณน้ำท่ารายวันของสถานีที่ทำการพยากรณ์ของวันปัจจุบันตามแต่ละกรณีของลุ่มน้ำที่ศึกษา

- จำนวนหน่วยในชั้นแสดงผล (Output) โดยใช้ข้อมูลปริมาณน้ำท่ารายวันจริงของสถานีที่ทำการพยากรณ์ ตามแต่กรณีที่ต้องการศึกษา

(4) จัดเตรียมข้อมูล โดยใช้โปรแกรม Microsoft Excel จัดเรียงให้อยู่ในรูปแบบของหน่วยนำเข้า แบบจำลองโครงข่ายประสาทเทียม ตามแต่ละรูปแบบของลุ่มน้ำที่ศึกษา ซึ่งแต่ละลุ่มน้ำจะมีไฟล์ Excel

(5) นำข้อมูลที่เตรียมใน Microsoft Excel เข้าสู่โปรแกรม Python เพื่อเตรียมสู่ในกระบวนการเรียนรู้และกระบวนการทดสอบ

(6) ทดสอบประสิทธิภาพความแม่นยำและความน่าเชื่อถือของการพยากรณ์

3.2 การวิเคราะห์ข้อมูลด้วยโปรแกรม Python

3.2.1 โปรแกรมที่ใช้ศึกษา

โปรแกรมภาษา Python (ไพทอน) เป็นภาษาโปรแกรมคอมพิวเตอร์ระดับสูง ที่ถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย โดยการตัดความซับซ้อนของโครงสร้างและไวยากรณ์ของภาษาออกไป เพื่อเอาไว้แปลงเป็นชุดคำสั่ง สั่งการใช้งานต่ออุปกรณ์เทคโนโลยีหลาย ๆ อย่าง ซึ่งการสั่งงานด้วยภาษา Python มักจะเขียนให้มีการทำงานแบบ Interpreter คือเป็นการแปลชุดคำสั่งทีละบรรทัดเพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่ต้องการโดยใช้โปรแกรม Spyder ในการเขียนภาษาไพธอน ซึ่งโปรแกรม Spyder เป็น โปรแกรมที่ถูกเขียนอย่างสมบูรณ์ใน Python ซึ่งออกแบบโดยนักวิทยาศาสตร์และมีไว้สำหรับนักวิทยาศาสตร์นักวิเคราะห์ข้อมูลและวิศวกรโดยเฉพาะ เป็นที่รู้จักกันในชื่อ Scientific Python Development IDE Pandas มาจากคำว่า PANEL DATA ซึ่งเป็น Library สำหรับวิเคราะห์ข้อมูลด้วย Pythonโดย Pandas เป็นโมดูลสำหรับรายงานข้อมูล (Data Analysis) ในภาษาไพทอน โดยสามารถสร้างหรือรายงานข้อมูลจากไฟล์ csv ได้ โดยวิธีการใช้งาน Pandas คือ โหลดไฟล์ข้อมูล เช่น CSV เข้าไปแล้วเราจะได้ข้อมูลในรูปแบบตาราง (Data Frame) ที่แบ่งข้อมูลตามแถวและคอลัมน์

3.2.2 ขั้นตอนการทำโปรแกรม

3.2.2.1 โค้ดภาษา Python สำหรับการหาค่าสัมประสิทธิ์สหพันธ์

```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
P17_all=pd.read_csv('D:/Project75
Data/Discharge/csv/P.17.csv',header=None)
P7A_all=pd.read_csv('D:/Project75
Data/Discharge/csv/P.7A.csv',header=None)

P17_all.columns = ['Datetime', 'P17']
P7A_all.columns = ['Datetime', 'P7A']

P17_all['Datetime']=pd.to_datetime(P17_all['Datetime'])
P17_all=P17_all.set_index('Datetime')

P7A_all['Datetime']=pd.to_datetime(P7A_all['Datetime'])
P7A_all=P7A_all.set_index('Datetime')

P17_P7Amerge =
pd.merge(P17_all,P7A_all,how='inner',left_index=True
,right_index=True)

i=1
while i!= 14:
    N='P7A'+'t-'+str(i)
    P17_P7Amerge[N]=P17_P7Amerge['P7A'].shift(i)
    i=i+1
P17_P7Amergecorr=P17_P7Amerge.corr()

import seaborn as sns
fig, ax = plt.subplots(figsize=(11,5))
ax.set_title('Correlation P17 and P7A 2010-2021')
ax=sns.heatmap(P17_P7Amergecorr,annot=True, cmap="twilight")
plt.xticks(rotation=0)
```

ภาพที่ 3.1 แสดงโค้ดโปรแกรมภาษา Python สำหรับการหาค่าสัมประสิทธิ์สหสัมพันธ์

จากรูปที่ 3.1 แสดงโค้ดโปรแกรมภาษา Python สำหรับการหาค่าสัมประสิทธิ์สหสัมพันธ์ (r) โดยมีการนำเข้าแพคเกจเสริม 4 แพคเกจด้วยกัน คือ

- Pandas เป็นแพคเกจสำหรับการจัดการข้อมูลที่ทำให้สามารถจัดการข้อมูลได้ง่ายขึ้น
- Numpy เป็นแพคเกจที่ใช้ในการคำนวณทางคณิตศาสตร์
- Matplotlib.pyplot เป็นแพคเกจสำหรับการสร้างกราฟ
- Seaborn เป็นแพคเกจสำหรับสร้างกราฟทางสถิติในภาษา Python

โดยโค้ดตัวอย่างในรูปที่ เป็นการหาค่าสัมประสิทธิ์สหสัมพันธ์ (r) ของตัวแปรที่นำเข้าข้อมูลที่ใน. csv โดยประกอบด้วยข้อมูลน้ำฝนจากสถานี 120161 และข้อมูลน้ำท่าจากสถานี P.17 ,P.15 ,P16 ,P7A ,P.26A ย้อนหลัง 10 ปี

คำอธิบายภาพที่ 3.1 สำหรับการหาค่าสัมประสิทธิ์สหสัมพันธ์

บรรทัดที่ 1-2	หมายถึง	การนำเข้าแพคเกจที่จำเป็นสำหรับการใช้งานได้แก่ pandas เป็นแพคเกจสำหรับการจัดการข้อมูลและ matplotlib.pyplot เป็น module สำหรับสร้างกราฟ โดยให้ชื่อย่อว่า plt และ numpy เป็นแพคเกจสำหรับการคำนวณทางคณิตศาสตร์ในภาษา Python
บรรทัดที่ 4-7	หมายถึง	การอ่านไฟล์.csv และสร้างคอลัมน์ Datetime P.17 , P.15 , P.16, P.7A , P.26A , RF ได้เป็น DataFrame

Data - DataFrame

Index	P17t	P17	P15	P16	P7A	P26A	RF
2010-04-01T00:00:00.000000000	211.6	211.6	375.9	249	229	0	0
2010-04-02T00:00:00.000000000	211.6	211.6	364.8	243	245	0	0
2010-04-03T00:00:00.000000000	218	218	375.9	246	197.4	0	0
2010-04-04T00:00:00.000000000	186	186	368.5	243	221	0	0
2010-04-05T00:00:00.000000000	205.2	205.2	361.1	231	181.8	0	0
2010-04-06T00:00:00.000000000	174.8	174.8	368.5	243	205.2	0	0
2010-04-07T00:00:00.000000000	186	186	333	178.2	125	0	0
2010-04-08T00:00:00.000000000	132.8	132.8	309.2	136.5	177.9	0	0
2010-04-09T00:00:00.000000000	172	172	299	126.5	163.5	0	0
2010-04-10T00:00:00.000000000	158	158	305.8	131.5	181.8	0	0
2010-04-11T00:00:00.000000000	169.2	169.2	285.4	114	174	0	0
2010-04-12T00:00:00.000000000	160.8	160.8	299	124	160	0	0
2010-04-13T00:00:00.000000000	152.4	152.4	292.2	124	177.9	0	0
2010-04-14T00:00:00.000000000	146.8	146.8	253.2	105.2	174	0	0
2010-04-15T00:00:00.000000000	138.4	138.4	282	111.8	185.7	0	0
2010-04-16T00:00:00.000000000	149.6	149.6	259.6	96.4	185.7	0	3.8
2010-04-17T00:00:00.000000000	152.4	152.4	224.4	72.2	185.7	0	0.2
2010-04-18T00:00:00.000000000	152.4	152.4	240.4	85.4	177.9	0	0
2010-04-19T00:00:00.000000000	146.8	146.8	253.2	89.8	163.5	0	0
2010-04-20T00:00:00.000000000	138.4	138.4	292.2	111.8	209.1	0	0
2010-04-21T00:00:00.000000000	166.4	166.4	329.6	114	156.5	0	0
2010-04-22T00:00:00.000000000	135.6	135.6	278.8	105.2	167	0	0
2010-04-23T00:00:00.000000000	132.8	132.8	299	147.4	185.7	0	0
2010-04-24T00:00:00.000000000	163.6	163.6	361.1	207	167	0	0
2010-04-25T00:00:00.000000000	189.2	189.2	387	228	167	0	0

ภาพที่ 3.2 Dataframes ประกอบคำอธิบาย Code บรรทัดที่ 4-9

บรรทัดที่ 9-10	หมายถึง	ให้คอลัมน์ P.17 และ P.7A แสดงเป็น Datetime
บรรทัดที่ 12-16	หมายถึง	จัดการข้อมูลทั้งหมดที่ใส่ไปในตัวโค้ด
บรรทัดที่ 18-20	หมายถึง	นำเข้าข้อมูล P.17 และ P.7A มาผสานกันโดยข้อมูลที่ผสานต้องเท่ากัน
บรรทัดที่ 22-27	หมายถึง	การหาค่าสัมประสิทธิ์สหพันธ์ (r) โดยใช้คำสั่ง while วนลูปและใช้ฟังก์ชัน.corr() ทำการคำนวณหาสัมประสิทธิ์สหสัมพันธ์

P17_P7Amergecorr - DataFrame

Index	P17	P7A	P7At-1	P7At-2	P7At-3	P7At-4	P7At-5	P7At-6	P7At-7	P7At-8	P7At-9	P7At-10	P7At-11	P7At-12	P7At-13
P17	1	0.939875	0.957995	0.939962	0.906299	0.869267	0.833311	0.799059	0.767088	0.738549	0.712658	0.69069	0.672966	0.658285	0.644405
P7A	0.939875	1	0.972092	0.928551	0.883871	0.842476	0.801164	0.761719	0.726157	0.694658	0.669591	0.650046	0.636236	0.624265	0.613074
P7At-1	0.957995	0.972092	1	0.972089	0.928543	0.883861	0.842462	0.801145	0.761697	0.726132	0.694629	0.669561	0.650014	0.636203	0.62423
P7At-2	0.939962	0.928551	0.972089	1	0.972086	0.928537	0.883851	0.842447	0.801127	0.761675	0.726107	0.694601	0.66953	0.649982	0.63617
P7At-3	0.906299	0.883871	0.928543	0.972086	1	0.972084	0.928531	0.88384	0.842433	0.801109	0.761653	0.726082	0.694573	0.669501	0.64995
P7At-4	0.869267	0.842476	0.883861	0.928537	0.972084	1	0.972082	0.928525	0.883831	0.84242	0.801093	0.761633	0.726059	0.694549	0.669473
P7At-5	0.833311	0.801164	0.842462	0.883851	0.928531	0.972082	1	0.972079	0.928519	0.883821	0.842407	0.801076	0.761614	0.726037	0.694523
P7At-6	0.799059	0.761719	0.801145	0.842447	0.88384	0.928525	0.972079	1	0.972077	0.928513	0.883811	0.842393	0.801059	0.761594	0.726014
P7At-7	0.767088	0.726157	0.761697	0.801127	0.842433	0.883831	0.928519	0.972077	1	0.972075	0.928507	0.883802	0.84238	0.801043	0.761574
P7At-8	0.738549	0.694658	0.726132	0.761675	0.801109	0.84242	0.883821	0.928513	0.972075	1	0.972072	0.928501	0.883792	0.842368	0.801027
P7At-9	0.712658	0.669591	0.694629	0.726107	0.761653	0.801093	0.842407	0.883811	0.928507	0.972072	1	0.97207	0.928495	0.883782	0.842354
P7At-10	0.69069	0.650046	0.669561	0.694601	0.726082	0.761633	0.801076	0.842393	0.883802	0.928501	0.97207	1	0.972068	0.928489	0.883773
P7At-11	0.672966	0.636236	0.650014	0.66953	0.694573	0.726059	0.761614	0.801059	0.84238	0.883792	0.928495	0.972068	1	0.972065	0.928484
P7At-12	0.658285	0.624265	0.636203	0.649982	0.669501	0.694549	0.726037	0.761594	0.801043	0.842368	0.883782	0.928489	0.972065	1	0.972063
P7At-13	0.644405	0.613074	0.62423	0.63617	0.64995	0.669473	0.694523	0.726014	0.761574	0.801027	0.842354	0.883773	0.928484	0.972063	1

<u>ภาพที่ 3.3</u> Dataframe ประกอบคำอธิบาย code บรรทัด 22-27

บรรทัดที่ 29-33 หมายถึง นำเข้าคำสั่ง seaborn ซึ่งเป็นแพคเกจสำหรับสร้างกราฟฟิกทางสถิติใน Python ที่ทำงานร่วมกับคำสั่ง matplotlib.pyplot กำหนด สี ตำแหน่ง ชื่อ และขนาด

3.2.2.2 โค้ดภาษา Python สำหรับการทำ MLP Regressor Validations

```
import numpy as np
import pandas as pd
#Load the data
data='D:/Project75 Data/Regression/TotalX.csv'
Data=pd.read_csv(data,header=None)

#Change Type of Column [0] from object to Datetime64
Data[0]=pd.to_datetime(Data[0])
Data=Data.set_index(0)

Data.insert(1, "P17", Data[1], True)
Data.columns = ['P17t', 'P17', 'P15', 'P16',
'P7A','P26A','RF']

#Add X's time with delta_hr
dh_x1=1
dh_x2=1
dh_x3=1
dh_x4=1
dh_x5=1
dh_x6=1
dh_x7=2

X1=Data['P17t'].shift(dh_x1, freq='D')
X2=Data['P17'].shift(dh_x2, freq='D')
X3=Data['P15'].shift(dh_x3, freq='D')
X4=Data['P16'].shift(dh_x4, freq='D')
X5=Data['P7A'].shift(dh_x5, freq='D')
X6=Data['P26A'].shift(dh_x6, freq='D')
X7=Data['RF'].shift(dh_x7, freq='D')
Y=Data['P17']
```

ภาพที่ 3.4 แสดงโค้ดโปรแกรมภาษา Python สำหรับการทำ MLP Regressor Validations

จากรูปที่ 3.4 แสดงโค้ดโปรแกรมภาษา Python สำหรับสำหรับการทำ MLP Regressor Validations โดยมีการนำเข้าแพคเกจเสริม 2 แพคเกจด้วยกัน คือ

- Pandas เป็นแพคเกจสำหรับการจัดการข้อมูลที่ทำให้สามารถจัดการข้อมูลได้ง่ายขึ้น
- Numpy เป็นแพคเกจที่ใช้ในการคำนวณทางคณิตศาสตร์

โดยโค้ดตัวอย่างในรูปที่ เป็นการหาค่า Validation ของตัวแปรที่นำเข้าข้อมูลที่อยู่ในไฟล์.csv โดยประกอบด้วยข้อมูลน้ำฝนจากสถานี 120161 และข้อมูลน้ำท่าจาสถานี P.17 ,P.15, P16, P.7A, P.26A ในช่วงเวลาตั้งแต่ ปี ค.ศ. 2010-2021

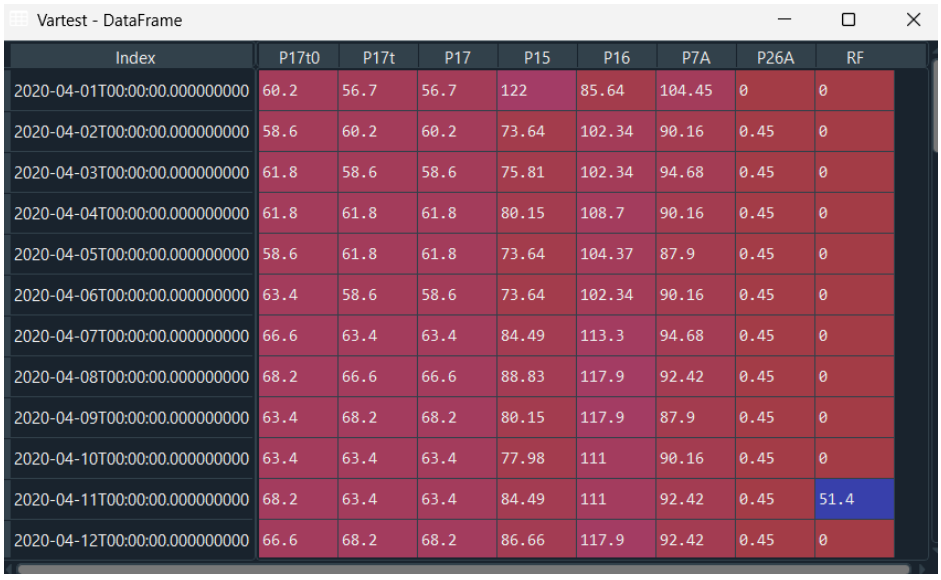
บรรทัดที่ 1-2	หมายถึง	การนำเข้าแพคเกจที่จำเป็นสำหรับการใช้งานได้แก่ pandas เป็นสำหรับจัดการข้อมูลและ numpy เป็นการคำนวณทางคณิตศาสตร์
บรรทัดที่ 4-5	หมายถึง	นำเข้าข้อมูลที่อยู่ในไฟล์.csv โดยประกอบด้วยข้อมูลน้ำฝนและข้อมูลน้ำท่า
บรรทัดที่ 7-9	หมายถึง	นำเข้าข้อมูลเป็น Datetime
บรรทัดที่ 11-12	หมายถึง	สร้างคอลัมน์ P.17t , P.17, P.15 ,P.16 , P.7A , P26A, RF
บรรทัดที่ 16-22	หมายถึง	ทำการเลือกวันของตัวแปรต่างๆ
บรรทัดที่ 24-31	หมายถึง	ทำการสร้างฟังก์ชันให้เป็น Data Series ของแต่ละตัวแปรเพื่อที่จำทำการ Merge (การผสานข้อมูล) ในขั้นตอนต่อไป

```
33
34 merge=pd.merge(Y,X1,how='inner',left_index=True,right_index=True)
35 merge=pd.merge(merge,X2,how='inner',left_index=True,right_index=True)
36 merge=pd.merge(merge,X3,how='inner',left_index=True,right_index=True)
37 merge=pd.merge(merge,X4,how='inner',left_index=True,right_index=True)
38 merge=pd.merge(merge,X5,how='inner',left_index=True,right_index=True)
39 merge=pd.merge(merge,X6,how='inner',left_index=True,right_index=True)
40 merge=pd.merge(merge,X7,how='inner',left_index=True,right_index=True)
41 merge.columns =  ['P17t0','P17t','P17','P15', 'P16', 'P7A','P26A','RF']
42 merge=merge.dropna()
43
44 # Select data for testing Apr2020 and Oct2020
45 Vartest_L=merge.loc['2020-04-01':'2020-04-30']
46 Vartest_H=merge.loc['2020-10-01':'2020-10-31']
47 Vartest = pd.concat([Vartest_L , Vartest_H])
48
49 merge=merge.loc[(merge.index < '2020-04-01') |
50 (merge.index >= '2020-05-01')]
51 merge=merge.loc[(merge.index < '2020-10-01') |
52 (merge.index >= '2020-11-01')]
53
54 xtest = Vartest.iloc[:,lambda Vartest:[1,2,3,4,5,6,7]].astype(float)
55 ytest = Vartest.iloc[:,lambda Vartest:[0]].astype(float)
56 ytest.columns = ['Yobs_test']
57
58 X = merge.iloc[:,lambda merge:[1,2,3,4,5,6,7]].astype(float)
59 Y = merge.iloc[:,0].astype(float) # output variable
60 (what we are trying to predict)
61 Y=Y.to_frame()
62 Y.columns = ['P17']
```

ภาพที่ 3.5 แสดงโค้ดโปรแกรมภาษา Python สำหรับการทำ MLP Regressor Validations

บรรทัดที่ 33-41	หมายถึง	ทำการ merge dataframe ทุก dataframe โดยใช้ Dataframe Index ที่เหมือนกัน และทำการ Drop แถวที่มีค่า NaN
บรรทัดที่ 44	หมายถึง	ใช้คำสั่ง Vartest_L (ช่วงปริมาณน้ำน้อย) เลือกข้อมูลสำหรับทดสอบโดยใช้ข้อมูลตั้งแต่ 1 เมษายน 2020 ถึง 1 พฤษภาคม 2020
บรรทัดที่ 45	หมายถึง	ใช้คำสั่ง Vartest_H (ช่วงปริมาณน้ำมาก) เลือกข้อมูลสำหรับทดสอบโดยใช้ข้อมูลตั้งแต่ 1 ตุลาคม 2020 ถึง 1 พฤศจิกายน 2020

บรรทัดที่ 53-55 1,2,3,4,5,6	หมายถึง	ทำการสร้าง Dataframe, Xtest จากคอลัมน์ที่
		และYtest จากคอลัมน์ที่ 0 โดย Dataframe ใหม่ จากคอลัมน์ Ytest ชื่อว่า Yobs_test
บรรทัดที่ 57-58	หมายถึง	ทำการเลือกตัวแปร X1,X2,X3,X4,X5,X6,X7 จาก Dataframe เลือกคอลัมน์ที่ 1,2,3,4,5,6,7 และ ตัวแปร Y เลือกคอลัมน์ 0
บรรทัดที่ 61-62	หมายถึง	หลังจากที่ merge เสร็จสิ้น ให้ Y เป็นคอลัมน์ P17



Vartest - DataFrame

Index	P17t0	P17t	P17	P15	P16	P7A	P26A	RF
2020-04-01T00:00:00.000000000	60.2	56.7	56.7	122	85.64	104.45	0	0
2020-04-02T00:00:00.000000000	58.6	60.2	60.2	73.64	102.34	90.16	0.45	0
2020-04-03T00:00:00.000000000	61.8	58.6	58.6	75.81	102.34	94.68	0.45	0
2020-04-04T00:00:00.000000000	61.8	61.8	61.8	80.15	108.7	90.16	0.45	0
2020-04-05T00:00:00.000000000	58.6	61.8	61.8	73.64	104.37	87.9	0.45	0
2020-04-06T00:00:00.000000000	63.4	58.6	58.6	73.64	102.34	90.16	0.45	0
2020-04-07T00:00:00.000000000	66.6	63.4	63.4	84.49	113.3	94.68	0.45	0
2020-04-08T00:00:00.000000000	68.2	66.6	66.6	88.83	117.9	92.42	0.45	0
2020-04-09T00:00:00.000000000	63.4	68.2	68.2	80.15	117.9	87.9	0.45	0
2020-04-10T00:00:00.000000000	63.4	63.4	63.4	77.98	111	90.16	0.45	0
2020-04-11T00:00:00.000000000	68.2	63.4	63.4	84.49	111	92.42	0.45	51.4
2020-04-12T00:00:00.000000000	66.6	68.2	68.2	86.66	117.9	92.42	0.45	0

ภาพที่ 3.6 Dataframe ประกอบคำอธิบาย Code บรรทัดที่ 33-41

```
64 ### Normalization of the data ###
65 from sklearn.preprocessing import MinMaxScaler
66 scaler = MinMaxScaler(feature_range=(0, 1))
67 scalerX = scaler.fit_transform(X.values)
68 scalerX = pd.DataFrame(scalerX, index=X.index, columns=X.columns)
69 scalerY = scaler.fit_transform(Y.values)
70 scalerY = pd.DataFrame(scalerY, index=X.index, columns=Y.columns)
71
72 from sklearn.metrics import mean_squared_error
73 # Function to calculate R2, RMSE, EI, and MAPE
74 def Cal_R2(Yobs_Ycal): # Yobs_Ycal must be dataframe 2 columns
75     R = Yobs_Ycal.corr(method="pearson")
76     R = R.iloc[0,1]
77     R2 = round(R**2,6)
78     print("R2 =" , R2)
79     return R2
80
81 def Cal_rmse(Yobs,Ycal):
82     mse = mean_squared_error(Yobs, Ycal)
83     Rmse = mse**0.5
84     print("RMSE =" , round(Rmse,6))
85     return Rmse
86 def nse(Yobs,Ycal):
87     global NSE
88     NSE = round(1-(np.sum((Yobs-Ycal)**2)/np.sum((Yobs-
89 np.mean(Ycal))**2)),4)
90     print("NSE =" , NSE)
91     return NSE
92 def Cal_mape(Yobs,Ycal):
93     Mape = round(np.mean(np.abs((Yobs - Ycal) / Yobs))*100,6)
94     print("MAPE =" , Mape)
95     return Mape
```

ภาพที่ 3.7 แสดงโค้ดโปรแกรมภาษา Python สำหรับการทำ MLP Regressor Validations

บรรทัดที่ 65- 70	หมายถึง	การนำเข้าแพคเกจที่มาใช้ในการทำนายพยากรณ์ข้อมูลต่างๆโดยจะให้ข้อมูลที่เป็น scalar มากจาก Dataframe
บรรทัดที่ 72	หมายถึง	การนำเข้าแพคเกจจที่จำเป็นสำหรับการใช้งาน ได้แก่ Skelearn.metric ของ R^2,RMSE,EI,MAPE
บรรทัดที่ 74-79	หมายถึง	คำนวณค่า R^2 (Coefficient Of Determination) โดยคำนวณจากค่า Ycal และ Yobs

บรรทัดที่ 81-85	หมายถึง	คำนวณค่า RMSE (Root Mean Square Error) โดยคำนวณจากค่า Ycal และ Yobs
บรรทัดที่ 86-91	หมายถึง	คำนวณค่า NSE (Nash Sutcliffe efficiency) โดยคำนวณจากค่า Ycal และ Yobs
บรรทัดที่ 92-95	หมายถึง	คำนวณค่า MAPE (Mean Absolute Percentage Error) โดยคำนวณจากค่า Ycal และ Yobs

```
  # Import ANNs
  from sklearn.neural network import MLPRegressor
  from sklearn.model selection import train_test_split

# Create lists for collecting the R2, RMSE, MAPE (Validation period)
A,B,C,D =[],[],[],[]
# Loop for random number 1-3
  seed = np.arange(1,4,1)
  for i in range(0,len(seed)):
    test_data_size = 0.20
Xtrain, Xvalidate, Ytrain, Yvalidate = train_test_split(scalerX,
scalerY, test_size
    = test_data_size, random_state  = seed[i])
Xtrainx, Xvalidatex, Ytrainx, Yvalidatex = train_test_split(X,Y,
test_size
    = test_data_size, random_state = seed[i])

    Hdn=np.arange(2,21,1)
    for i in range(0,len(Hdn)):
        Ann =
MLPRegressor(hidden_layer_sizes=(Hdn[i])).fit(Xtrain,Ytrain)

        # make a prediction
        Ycal = Ann.predict(Xvalidate)
        Ycal = pd.DataFrame(Ycal)
        Ycal = scaler.inverse_transform(Ycal)

        # keep Yvalidate original (Datetime Index)
        Yvalidate0 = scaler.inverse_transform(Yvalidate)
        Yvalidate1=Yvalidate0.copy()
        Yvalidate1=np.append(Yvalidate0, Ycal, axis=1)
        Yvalidate1=pd.DataFrame(Yvalidate1)
        Yvalidate1.columns=['Yobs','Ycal']
        Yvalidate1.index=Yvalidate.index

 Print("==========================")
    print("Model performance of the validation period (20%)")
    R2 = Cal_R2(Yvalidate1) # Calculate R2
   nse = Cal_nse(Yvalidate1['Yobs'],Yvalidate1['Ycal'])#Calculate nse
    Rmse = Cal_rmse(Yvalidate1['Yobs'],Yvalidate1['Ycal'])
    # Calculate Root mean square  error (RMSE)
    Mape = Cal_mape(Yvalidate1['Yobs'],Yvalidate1['Ycal'])
    # Calculate Mean absolute percent error (MAPE)
```

ภาพที่ 3.8 แสดงโค้ดโปรแกรมภาษา Python สำหรับการทำ MLP Regressor Validations

บรรทัดที่ 105-106	หมายถึง	การนำเข้าแพคเกจ ANNs มาใช้ในการพยากรณ์และ Train Test
บรรทัดที่ 109-110	หมายถึง	สร้าง List A,B,C,D โดยการทดสอบจะทำทั้งหมด3 Random
บรรทัดที่ 111	หมายถึง	Seed การสุ่มข้อมูล
บรรทัดที่ 114-118	หมายถึง	แบ่งข้อมูลเป็นช่วงเรียนรู้ (Training=80%)
บรรทัดที่ 121-124	หมายถึง	สร้างรูปแบบชั้นซ่อนเร่นในการทำนาย 20 ตัว โดยเริ่มที่ 0
บรรทัดที่ 127-137	หมายถึง	สร้างการทำนายข้อมูล Ycal ให้แสดงมาจาก Dataframeและจัดเก็บข้อมูลที่ Validate โดยเก็บเป็น Yobs,Ycal
บรรทัดที่ 139-145	หมายถึง	แบ่งข้อมูลเป็นช่วงทดสอบ (Validation=20%) โดยคำนวณค่า R^2,RMSE,NSE,MAPE โดยคำนวณจาก Ycal,Yobs

```
147      # Collect the results
148         A.append(R2)
149         df1 = pd.DataFrame(A)
150         df1 = df1.T
151 
152         B.append(Rmse)
153         df2 = pd.DataFrame(B)
154         df2 = df2.T
155 
156         C.append(Mape)
157         df3 = pd.DataFrame(C)
158         df3 = df3.T
159 
160         D.append(NSE)
161         df4 = pd.DataFrame(D)
162         df4 = df4.T
163        dfA, dfB, dfC, dfD = pd.DataFrame(),
164 pd.DataFrame() ,pd.DataFrame(),
165        pd.DataFrame()
166      row=19 # number of row that you want
167      i=0
168      col_df = 0
169      while i < len(df1.columns):
170     dfA[col_df]=df1.iloc[0,i:i+row].reset_index(
171 drop=True)
172     i=i+row
173     col_df=col_df+1
174     dfA=dfA.T
175     row=19 # number of row that you want
176     i=0
177    col_df = 0
178     while i < len(df2.columns):
179     dfB[col_df]=df2.iloc[0,i:i+row].reset_index
180 (drop=True)
181     i=i+row
182     col_df=col_df+1
183     dfB=dfB.T
```

ภาพที่ 3.9 แสดงโค้ดโปรแกรมภาษา Python สำหรับการทำ MLP Regressor Validations

```
184     row=19 # number of row that you want
185     i=0
186     col_df = 0
187     while i < len(df3.columns):
188 
189 dfC[col_df]=df3.iloc[0,i:i+row].reset_index(drop=True)
190     i=i+row
191     col_df=col_df+1
192     dfC=dfC.T
193     row=19 # number of row that you want
194     i=0
195    col_df = 0
196    while i < len(df4.columns):
197 
198 dfD[col_df]=df4.iloc[0,i:i+row].reset_index(drop=True)
199     i=i+row
200     col_df=col_df+1
201     dfD=dfD.T
202  dfA.to_csv('D:/Project75Data/Regression/Validation
203 /Case1/'+'Cast1_R2_Day1'+'.csv')
204 dfB.to_csv('D:/Project75Data/Regression/Validation
205 /Case1/'+'Cast1_RMSE_Day1'+'.csv')
206 dfC.to_csv('D:/Project75Data/Regression/Validation
207 /Case1/'+'Cast1_MAPE_Day1'+'.csv')
208 dfD.to_csv('D:/Project75Data/Regression/Validation
209 /Case1/'+'Cast1_EI_Day1'+'.csv')
```

ภาพที่ 3.10 แสดงโค้ดโปรแกรมภาษา Python สำหรับการทำ MLP Regressor Validations

บรรทัดที่ 147-162	หมายถึง	การเก็บข้อมูลที่คำนวณได้ (R^2,RMSE,NSE,MAPE) ไว้ในตัวแปร A,B,C,D
บรรทัดที่ 164	หมายถึง	ให้ dfA, dfB, dfC, dfD เป็นค่าที่มาจาก Dataframe
บรรทัดที่ 166-174	หมายถึง	ให้แถวที่ 19 เป็นผลลัพธ์ที่ต้องการ โดยเริ่มที่แถว 0 โดย dfA คือค่า df1 ที่มาจาก Dataframe เช่นกัน
บรรทัดที่ 176-183	หมายถึง	ให้แถวที่ 19 เป็นผลลัพธ์ที่ต้องการ โดยเริ่มที่แถว 0 โดย dfB คือค่า df2 ที่มาจาก Dataframe เช่นกัน
บรรทัดที่ 184-191	หมายถึง	ให้แถวที่ 19 เป็นผลลัพธ์ที่ต้องการ โดยเริ่มที่แถว 0 โดย dfC คือค่า df3 ที่มาจาก Dataframe เช่นกัน

บรรทัดที่ 192-199	หมายถึง	ให้แถวที่ 19 เป็นผลลัพธ์ที่ต้องการ โดยเริ่มที่แถว 0 โดย dfD คือค่า df4 ที่มาจาก Dataframe เช่นกัน
บรรทัดที่ 200-207	หมายถึง	คือการ Save ข้อมูล dfA, dfB, dfC, dfD เป็น.csv ไปยังแหล่งที่อยู่ของไฟล์ที่เลือก

Random 1

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	1	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.929	0.685	0.700	0.723	0.970	0.887	0.924	0.932	0.843	0.967	0.948	0.973	0.984	0.962	0.980	0.957	0.986	0.971	0.973
			RMSE	87.790	248.979	150.457	143.035	47.387	91.259	75.385	72.000	114.680	49.710	66.654	47.772	34.186	53.462	38.610	57.300	32.535	46.171	46.060
			MAPE	35.273	135.895	79.096	50.952	23.758	39.207	54.244	30.831	101.957	22.918	74.778	35.384	16.046	17.826	16.469	43.402	17.343	33.942	22.290
B	1	P17 = f(P17,P15,P16,RF)	R2	0.885	0.688	0.969	0.901	0.952	0.961	0.924	0.950	0.966	0.975	0.952	0.969	0.961	0.962	0.951	0.959	0.972	0.960	0.973
			RMSE	109.773	161.430	50.627	88.818	104.382	56.511	76.722	61.150	50.343	43.209	59.778	48.959	55.454	56.360	74.695	55.204	45.826	54.542	45.058
			MAPE	81.736	150.486	19.977	28.275	68.548	21.508	48.703	54.032	37.537	36.164	25.412	12.869	30.058	23.874	22.420	19.773	17.429	28.825	21.454
C	1	P17= f(P17,P15,P16)	R2	0.966	0.964	0.899	0.951	0.972	0.959	0.952	0.950	0.863	0.966	0.967	0.952	0.967	0.969	0.967	0.969	0.949	0.963	0.939
			RMSE	62.763	59.070	108.747	62.975	47.013	61.399	75.741	67.781	102.857	81.031	50.727	59.869	50.488	48.796	49.618	48.828	63.283	52.495	84.204
			MAPE	39.107	27.146	54.606	55.722	22.405	31.625	63.941	24.712	53.975	35.474	17.769	30.823	33.849	32.047	21.875	20.052	45.486	18.855	41.784
D	1	P17 = f(P17,P16,P7A)	R2	0.976	0.972	0.907	0.975	0.953	0.972	0.974	0.882	0.944	0.939	0.960	0.969	0.952	0.979	0.952	0.970	0.968	0.963	0.963
			RMSE	48.680	98.358	320.566	100.060	69.606	53.425	44.653	94.412	69.214	67.535	54.864	49.034	59.895	45.937	60.211	48.400	56.288	52.801	52.563
			MAPE	20.301	66.219	252.755	71.154	18.301	30.680	21.279	44.083	55.531	41.960	23.636	22.818	31.059	25.162	27.075	15.919	50.571	24.355	39.533

ภาพที่ 3.11 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการ Random ครั้งที่ 1 คาดการณ์ล่วงหน้า 1 วันและแสดง Hidden Layer 20 node ของค่า R^2 RMSE และ MAPE

Random 2

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	1	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.680	0.932	0.011	0.950	0.908	0.897	0.962	0.937	0.958	0.919	0.974	0.985	0.981	0.984	0.973	0.979	0.967	0.980	0.973
			RMSE	157.496	70.182	263.527	62.183	115.840	85.816	52.309	67.004	54.781	75.889	43.498	32.771	37.967	36.271	43.335	39.433	48.137	38.440	47.125
			MAPE	124.543	74.418	183.695	77.496	49.322	69.184	29.950	62.779	19.819	29.968	19.869	24.442	16.570	18.829	26.572	19.832	25.050	25.181	34.536
B	1	P17 = f(P17,P15,P16,RF)	R2	0.000	0.971	0.972	0.970	0.961	0.941	0.893	0.973	0.960	0.948	0.964	0.967	0.974	0.970	0.956	0.973	0.948	0.967	0.962
			RMSE	271.504	104.396	54.183	48.910	52.462	70.745	98.731	43.831	53.018	60.657	50.171	48.916	44.363	47.467	55.522	44.429	60.343	48.574	53.146
			MAPE	117.761	85.032	42.551	19.666	24.160	27.980	28.127	29.766	43.196	75.073	25.226	18.154	17.202	20.402	23.491	29.329	57.708	34.595	15.523
C	1	P17= f(P17,P15,P16)	R2	0.948	0.935	0.938	0.967	0.894	0.949	0.963	0.937	0.952	0.955	0.962	0.959	0.964	0.957	0.945	0.946	0.958	0.962	0.962
			RMSE	86.116	65.630	63.360	213.459	122.448	57.596	61.015	66.115	73.231	57.643	51.118	52.738	48.898	52.936	59.235	59.129	52.488	49.542	50.688
			MAPE	76.792	32.514	72.973	162.103	55.459	33.003	21.602	76.177	53.377	62.933	42.372	22.096	28.687	18.237	25.902	47.795	21.055	28.126	30.864
D	1	P17 = f(P17,P16,P7A)	R2	0.000	0.954	0.780	0.964	0.932	0.971	0.966	0.966	0.870	0.959	0.964	0.954	0.962	0.969	0.953	0.938	0.953	0.949	0.954
			RMSE	253.032	60.547	173.648	50.101	69.833	47.943	64.426	46.690	91.310	58.687	48.788	55.637	50.829	45.797	55.201	66.183	57.824	60.324	54.371
			MAPE	157.833	36.509	97.479	26.244	69.160	27.760	45.476	26.691	32.954	21.422	20.460	48.454	21.915	36.753	26.166	17.857	32.156	25.136	53.733

ภาพที่ 3.12 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการ Random ครั้งที่ 2 คาดการณ์ล่วงหน้า 1 วันและแสดง Hidden Layer 20 node ของค่า R^2 RMSE และ MAPE

Random 3

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	1	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.883	0.743	0.856	0.945	0.873	0.979	0.921	0.948	0.930	0.886	0.971	0.971	0.972	0.973	0.961	0.933	0.962	0.970	0.965
			RMSE	88.297	142.218	95.464	58.824	96.937	36.201	69.725	62.300	68.553	87.020	42.772	42.935	42.569	40.927	49.139	64.625	50.444	43.219	46.932
			MAPE	124.543	74.418	183.695	77.496	49.322	69.184	29.950	62.779	19.819	29.968	19.869	24.442	16.570	18.829	26.572	19.832	25.050	25.181	34.536
B	1	P17 = f(P17,P15,P16,RF)	R2	0.003	0.896	0.967	0.953	0.933	0.952	0.893	0.912	0.968	0.957	0.933	0.965	0.953	0.931	0.956	0.957	0.966	0.942	0.967
			RMSE	248.699	207.125	48.274	54.309	64.409	56.074	81.846	74.720	45.806	54.044	68.418	47.847	54.554	72.001	52.261	53.425	47.059	60.602	47.143
			MAPE	171.926	142.095	19.664	26.315	50.186	21.666	34.511	54.374	16.219	61.453	67.571	35.752	51.006	42.232	45.652	24.543	21.483	50.519	16.593
C	1	P17= f(P17,P15,P16)	R2	0.950	0.948	0.942	0.959	0.934	0.923	0.951	0.949	0.894	0.955	0.921	0.939	0.959	0.953	0.951	0.958	0.922	0.947	0.957
			RMSE	80.901	66.854	59.017	90.032	61.650	70.347	52.830	54.182	77.719	53.614	66.994	60.997	49.207	52.795	53.875	49.683	70.376	55.743	50.377
			MAPE	52.718	22.485	22.920	89.547	13.392	58.245	13.324	27.850	87.082	13.425	56.648	53.383	13.624	24.164	24.158	17.181	71.013	25.581	25.477
D	1	P17 = f(P17,P16,P7A)	R2	0.861	0.948	0.956	0.078	0.895	0.963	0.941	0.919	0.964	0.965	0.963	0.920	0.968	0.953	0.937	0.964	0.906	0.959	0.962
			RMSE	89.956	133.204	51.452	242.089	82.861	46.347	74.460	70.465	45.318	61.028	49.993	68.202	43.647	53.036	60.225	47.122	76.429	50.069	46.811
			MAPE	75.829	116.876	19.174	210.049	73.469	25.481	28.244	91.423	20.662	31.768	17.312	18.668	23.879	12.933	12.953	19.492	76.203	16.748	25.807

ภาพที่ 3.13 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการ Random ครั้งที่ 3 คาดการณ์ล่วงหน้า 1 วันและแสดง Hidden Layer 20 node ของค่า R^2 RMSE และ MAPE

Average

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	1	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.831	0.787	0.522	0.873	0.917	0.921	0.936	0.939	0.910	0.924	0.964	0.976	0.979	0.973	0.971	0.956	0.971	0.974	0.970
			RMSE	111.195	153.793	169.816	88.014	86.721	71.092	65.806	67.101	79.338	70.873	50.975	41.159	38.240	43.554	43.694	53.786	43.705	42.610	46.706
			MAPE	94.786	94.910	148.829	68.648	40.801	59.192	38.048	52.129	47.198	27.618	38.172	28.089	16.396	18.494	23.204	27.688	22.481	28.101	30.454
B	1	P17 = f(P17,P15,P16,RF)	R2	0.296	0.852	0.969	0.941	0.948	0.951	0.903	0.945	0.965	0.960	0.950	0.967	0.963	0.954	0.954	0.963	0.962	0.957	0.967
			RMSE	209.992	157.650	51.028	64.012	73.751	61.110	85.766	59.900	49.723	52.637	59.456	48.574	51.457	58.609	60.826	51.019	51.076	54.572	48.449
			MAPE	123.808	125.871	27.397	24.752	47.631	23.718	37.114	46.057	32.317	57.564	39.403	22.258	32.755	28.836	30.521	24.548	32.207	37.980	17.857
C	1	P17= f(P17,P15,P16)	R2	0.955	0.949	0.926	0.959	0.933	0.944	0.955	0.945	0.903	0.959	0.950	0.950	0.963	0.960	0.954	0.958	0.943	0.957	0.953
			RMSE	76.593	63.851	77.042	122.155	77.037	63.114	63.195	62.693	84.602	64.096	56.280	57.868	49.531	51.509	54.243	52.547	62.049	52.593	61.757
			MAPE	56.206	27.382	50.166	102.457	30.419	40.958	32.956	42.913	64.811	37.277	38.930	35.434	25.387	24.816	23.978	28.343	45.851	24.187	32.708
D	1	P17 = f(P17,P16,P7A)	R2	0.612	0.958	0.881	0.672	0.927	0.969	0.960	0.922	0.926	0.954	0.962	0.948	0.961	0.967	0.947	0.957	0.942	0.957	0.960
			RMSE	130.556	97.370	181.889	130.750	74.100	49.238	61.179	70.522	68.614	62.417	51.215	57.625	51.457	48.257	58.546	53.902	63.514	54.398	51.248
			MAPE	84.654	73.201	123.136	102.482	53.643	27.974	31.666	54.066	36.382	31.717	20.469	29.980	25.618	24.949	22.065	17.756	52.977	22.080	39.691

ภาพที่ 3.14 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการ Random ทั้งหมด คาดการณ์ล่วงหน้า 1 วันและแสดง Hidden Layer 20 node ของค่า R^2 RMSE และ MAPE

Random 1

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	2	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.000	0.827	0.923	0.942	0.942	0.888	0.923	0.902	0.914	0.952	0.950	0.930	0.869	0.932	0.945	0.952	0.948	0.931	0.961
			RMSE	273.458	121.710	79.743	67.464	72.620	101.484	79.488	100.744	85.486	60.898	61.367	70.265	98.532	69.337	62.025	60.414	60.382	70.915	53.380
			MAPE	255.377	92.288	70.990	51.443	31.269	43.450	27.406	44.541	57.729	33.738	35.627	34.012	48.351	51.213	30.012	51.815	19.027	35.529	27.875
B	2	P17 = f(P17,P15,P16,RF)	R2	0.000	0.871	0.916	0.904	0.914	0.870	0.928	0.917	0.920	0.924	0.940	0.933	0.896	0.925	0.928	0.943	0.913	0.943	0.938
			RMSE	270.523	101.188	91.044	100.888	79.049	99.779	74.185	76.711	79.212	73.526	65.928	69.003	86.677	73.393	82.882	63.597	77.909	64.173	66.197
			MAPE	241.251	31.566	73.388	82.049	32.618	35.485	77.494	37.312	38.512	33.551	31.277	50.302	48.179	53.007	58.144	20.262	29.917	19.777	24.228
C	2	P17= f(P17,P15,P16)	R2	0.565	0.515	0.925	0.919	0.922	0.826	0.914	0.913	0.926	0.925	0.909	0.927	0.914	0.918	0.921	0.896	0.925	0.927	0.929
			RMSE	286.005	224.325	74.473	98.494	76.095	125.422	81.367	81.623	87.257	77.503	82.039	73.414	81.479	79.547	76.667	88.487	74.696	73.212	72.403
			MAPE	101.311	39.676	19.068	47.863	30.217	104.416	47.782	66.928	33.369	28.077	36.243	20.665	33.755	22.753	20.403	54.517	21.741	23.465	22.124
D	2	P17 = f(P17,P16,P7A)	R2	0.891	0.911	0.875	0.884	0.937	0.932	0.910	0.901	0.895	0.895	0.865	0.915	0.900	0.846	0.936	0.926	0.931	0.925	0.907
			RMSE	133.521	118.385	97.882	104.774	97.363	79.524	94.000	88.650	88.007	87.952	121.498	79.511	87.369	130.588	68.728	74.166	73.285	74.306	86.300
			MAPE	70.766	44.884	59.367	114.551	55.209	48.701	52.676	71.588	42.005	29.954	32.442	49.694	20.498	41.011	24.794	49.167	17.976	18.258	50.692

ภาพที่ 3.15 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการ Random ครั้งที่ 1 คาดการณ์ล่วงหน้า 2 วันและแสดง Hidden Layer 20 node ของค่า R^2 RMSE และ MAPE

Random 2

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	2	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.930	0.899	0.919	0.902	0.830	0.897	0.905	0.923	0.851	0.920	0.901	0.917	0.918	0.928	0.907	0.928	0.905	0.923	0.933
			RMSE	79.787	88.880	98.994	86.434	113.346	87.642	101.578	76.038	106.860	81.668	109.703	79.085	78.195	73.699	83.560	73.387	85.692	76.630	72.115
			MAPE	40.857	37.199	56.683	32.426	31.680	51.210	68.056	29.140	89.685	27.126	37.987	37.692	46.664	55.031	75.579	52.777	35.545	52.471	27.717
B	2	P17 = f(P17,P15,P16,RF)	R2	0.887	0.908	0.902	0.059	0.914	0.919	0.889	0.901	0.909	0.899	0.915	0.880	0.907	0.905	0.890	0.902	0.911	0.908	0.914
			RMSE	100.666	109.658	92.660	273.733	86.143	79.047	113.170	87.453	85.429	86.643	80.484	96.618	84.186	85.467	98.495	86.712	85.135	84.254	81.645
			MAPE	24.939	67.719	24.480	207.384	60.452	30.271	63.083	31.555	22.848	52.516	50.912	31.882	44.863	37.797	32.322	33.489	22.823	20.395	27.285
C	2	P17= f(P17,P15,P16)	R2	0.000	0.669	0.827	0.904	0.869	0.909	0.898	0.907	0.907	0.896	0.907	0.898	0.892	0.905	0.903	0.916	0.909	0.905	0.888
			RMSE	266.574	155.726	109.883	86.657	96.325	82.595	86.773	81.516	102.592	89.318	85.151	112.932	89.255	88.592	85.128	79.919	80.558	84.924	89.492
			MAPE	238.489	182.652	93.348	68.125	75.467	23.226	42.199	23.696	71.060	23.013	43.493	55.441	71.274	71.787	19.581	25.855	24.529	55.797	45.085
D	2	P17 = f(P17,P16,P7A)	R2	0.905	0.861	0.915	0.913	0.859	0.890	0.921	0.905	0.884	0.930	0.897	0.890	0.919	0.924	0.917	0.906	0.916	0.927	0.892
			RMSE	94.517	124.814	113.798	121.895	103.789	92.490	86.469	82.254	90.044	71.753	89.130	117.874	82.596	74.659	76.391	83.884	79.106	74.484	87.187
			MAPE	33.675	186.742	75.278	87.053	85.202	24.907	42.523	30.081	37.848	25.357	23.450	80.700	23.898	32.939	28.540	21.824	46.933	21.896	53.168

ภาพที่ 3.16 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการ Random ครั้งที่ 2 คาดการณ์ล่วงหน้า 2 วันและแสดง Hidden Layer 20 node ของค่า R^2 RMSE และ MAPE

Random 3

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	2	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.009	0.836	0.757	0.893	0.904	0.902	0.911	0.864	0.923	0.899	0.901	0.908	0.906	0.925	0.909	0.915	0.914	0.914	0.912
			RMSE	261.378	106.307	125.846	85.088	80.305	80.927	76.670	94.627	70.878	81.400	80.586	78.144	80.890	70.028	77.315	76.767	75.839	74.863	76.825
			MAPE	244.886	56.371	71.735	33.332	50.957	87.439	62.645	34.540	29.905	45.067	35.601	76.290	46.236	24.512	49.984	30.083	27.325	45.624	21.634
B	2	P17 = f(P17,P15,P16,RF)	R2	0.902	0.836	0.431	0.767	0.862	0.910	0.866	0.901	0.914	0.909	0.907	0.916	0.909	0.906	0.894	0.923	0.911	0.895	0.908
			RMSE	87.068	127.609	238.157	125.235	98.638	79.317	101.595	82.390	105.293	77.936	97.239	75.499	78.893	79.245	84.998	72.263	77.330	84.589	78.539
			MAPE	92.050	82.023	117.934	85.228	27.572	61.586	23.941	22.568	68.431	30.486	62.509	24.329	26.885	23.413	32.501	30.860	24.130	48.771	29.633
C	2	P17= f(P17,P15,P16)	R2	0.594	0.047	0.772	0.795	0.749	0.889	0.852	0.813	0.857	0.826	0.830	0.876	0.848	0.879	0.858	0.895	0.885	0.877	0.887
			RMSE	159.673	258.891	157.791	112.605	140.090	85.712	108.492	114.997	96.229	105.874	102.866	88.798	97.300	87.903	94.314	83.435	85.918	88.701	85.100
			MAPE	66.095	89.000	69.144	78.732	191.973	20.705	97.935	70.668	35.783	50.288	35.033	52.478	31.158	30.098	46.489	30.677	19.624	51.730	25.215
D	2	P17 = f(P17,P16,P7A)	R2	0.700	0.858	0.004	0.720	0.891	0.887	0.911	0.884	0.850	0.905	0.882	0.861	0.900	0.813	0.863	0.862	0.887	0.883	0.859
			RMSE	242.541	180.415	251.067	229.265	86.579	84.038	76.147	87.261	96.742	77.347	96.962	93.932	81.997	107.944	93.471	94.521	86.248	87.960	94.654
			MAPE	141.405	119.148	204.873	89.552	39.205	45.167	21.865	44.318	27.298	26.431	34.538	29.370	30.359	59.800	42.574	67.763	36.055	56.310	49.402

ภาพที่ 3.17 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการ Random ครั้งที่ 3 คาดการณ์ล่วงหน้า 2 วันและแสดง Hidden Layer 20 node ของค่า R^2 RMSE และ MAPE

Average

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	2	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.313	0.854	0.867	0.912	0.892	0.895	0.913	0.896	0.896	0.923	0.917	0.918	0.898	0.929	0.920	0.932	0.922	0.923	0.935
			RMSE	204.874	105.632	101.528	79.662	88.757	90.018	85.912	90.470	87.742	74.655	83.885	75.831	85.873	71.021	74.300	70.189	73.971	74.136	67.440
			MAPE	180.373	61.953	66.469	39.067	37.969	60.700	52.703	36.074	59.106	35.310	36.405	49.331	47.083	43.585	51.858	44.892	27.299	44.541	25.742
B	2	P17 = f(P17,P15,P16,RF)	R2	0.596	0.872	0.749	0.577	0.897	0.900	0.895	0.906	0.914	0.911	0.921	0.909	0.904	0.912	0.904	0.922	0.912	0.915	0.920
			RMSE	152.752	112.818	140.620	166.619	87.943	86.048	96.317	82.185	89.978	79.368	81.217	80.373	83.252	79.369	88.792	74.191	80.125	77.672	75.460
			MAPE	119.413	60.436	71.934	124.887	40.214	42.448	54.839	30.478	43.263	38.851	48.233	35.506	39.976	38.072	40.989	28.203	25.623	29.648	27.048
C	2	P17= f(P17,P15,P16)	R2	0.386	0.410	0.841	0.873	0.847	0.875	0.888	0.878	0.897	0.882	0.882	0.900	0.885	0.901	0.894	0.902	0.906	0.903	0.901
			RMSE	237.417	212.981	114.049	99.252	104.170	97.909	92.211	92.712	95.360	90.898	90.018	91.715	89.344	85.347	85.370	83.947	80.391	82.279	82.332
			MAPE	135.298	103.776	60.520	64.907	99.219	49.449	62.639	53.764	46.737	33.793	38.256	42.861	45.396	41.546	28.824	37.016	21.965	43.664	30.808
D	2	P17 = f(P17,P16,P7A)	R2	0.832	0.877	0.598	0.839	0.896	0.903	0.914	0.897	0.876	0.910	0.881	0.889	0.906	0.861	0.905	0.898	0.911	0.912	0.886
			RMSE	156.860	141.204	154.249	151.978	95.910	85.351	85.539	86.055	91.598	79.017	102.530	97.106	83.987	104.397	79.530	84.190	79.546	78.917	89.380
			MAPE	81.949	116.925	113.173	97.052	59.872	39.592	39.021	48.662	35.717	27.247	30.143	53.255	24.918	44.583	31.969	46.251	33.655	32.155	51.087

ภาพที่ 3.18 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการเฉลี่ยจาก Random ทั้งหมด คาดการณ์ล่วงหน้า 2 วันและแสดง Hidden Layer 20 node ของค่า R^2 RMSE และ MAPE

Random 1

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	3	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.865	0.000	0.855	0.873	0.895	0.824	0.897	0.859	0.850	0.877	0.878	0.883	0.864	0.888	0.896	0.884	0.886	0.872	0.880
			RMSE	103.855	264.588	108.463	92.256	84.646	112.073	85.410	97.283	103.425	90.850	92.571	91.694	100.656	89.328	83.601	88.761	89.104	94.249	91.731
			MAPE	41.222	230.953	119.222	29.416	37.088	38.374	31.884	62.875	30.431	34.713	42.233	49.297	64.771	42.272	34.887	32.949	50.397	47.930	41.043
B	3	P17 = f(P17,P15,P16,RF)	R2		0.229	0.882	0.856	0.825	0.748	0.848	0.864	0.888	0.885	0.860	0.876	0.852	0.883	0.885	0.873	0.874	0.884	0.884
			RMSE	258.123	271.010	88.829	108.599	108.365	141.787	101.670	95.198	87.665	89.180	96.773	91.340	101.571	89.337	88.364	93.772	93.743	88.271	88.181
			MAPE	156.928	159.561	31.281	65.131	46.164	88.209	45.092	39.667	35.048	31.367	47.560	29.553	41.671	40.279	30.872	48.876	35.756	28.958	30.392
C	3	P17= f(P17,P15,P16)	R2	0.886	0.715	0.794	0.866	0.837	0.878	0.874	0.882	0.881	0.880	0.876	0.859	0.886	0.878	0.858	0.876	0.881	0.871	0.879
			RMSE	190.905	173.635	136.870	99.392	114.345	154.611	94.105	92.041	91.901	92.764	93.031	101.875	93.326	92.612	101.011	93.763	91.616	94.726	93.015
			MAPE	133.504	119.777	46.518	68.840	82.138	90.716	50.953	55.193	34.451	36.650	29.069	71.678	33.198	60.248	33.992	59.725	42.747	26.253	36.015
D	3	P17 = f(P17,P16,P7A)	R2	0.891	0.781	0.889	0.890	0.878	0.869	0.789	0.899	0.874	0.845	0.888	0.867	0.893	0.821	0.900	0.893	0.889	0.897	0.879
			RMSE	90.062	132.329	117.561	102.990	94.452	98.957	195.841	91.062	98.123	105.467	94.188	96.849	87.415	112.113	84.586	91.956	89.007	87.190	91.746
			MAPE	41.712	62.619	49.957	51.337	74.555	49.680	107.803	28.412	62.341	48.680	37.593	51.610	26.141	53.999	31.257	30.739	44.804	26.775	31.676

ภาพที่ 3.19 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการ Random ครั้งที่ 1 คาดการณ์ล่วงหน้า 3 วันและแสดง Hidden Layer 20 node ของค่า R^2 RMSE และ MAPE

Random 2

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	3	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.796	0.848	0.836	0.809	0.813	0.838	0.839	0.670	0.831	0.774	0.818	0.831	0.817	0.813	0.802	0.848	0.814	0.827	0.818
			RMSE	133.589	130.306	134.010	123.441	120.443	226.958	113.315	159.723	114.973	132.404	118.688	115.043	120.162	123.121	125.553	110.408	119.965	116.976	119.503
			MAPE	141.743	78.298	79.617	56.958	55.911	172.526	44.369	110.932	45.247	69.393	38.036	63.134	53.812	73.380	42.235	34.881	80.043	33.204	27.895
B	3	P17 = f(P17,P15,P16,RF)	R2	0.501	0.150	0.000	0.837	0.839	0.607	0.825	0.818	0.845	0.810	0.809	0.793	0.818	0.837	0.838	0.828	0.838	0.829	0.831
			RMSE	267.328	290.971	279.177	182.538	111.889	208.915	133.799	121.055	112.388	121.637	122.973	128.894	119.058	114.720	114.275	118.099	112.960	117.201	116.871
			MAPE	169.928	296.916	228.163	139.036	35.563	152.151	64.090	45.741	27.923	56.017	90.977	91.606	28.519	35.645	34.262	31.732	40.765	32.987	35.326
C	3	P17= f(P17,P15,P16)	R2	0.834	0.713	0.000	0.830	0.422	0.839	0.808	0.834	0.823	0.836	0.844	0.836	0.844	0.834	0.838	0.835	0.848	0.835	0.836
			RMSE	120.363	146.464	275.162	114.229	227.896	112.737	120.671	116.154	118.589	136.387	112.511	117.295	118.230	115.734	112.338	114.347	111.100	113.772	113.895
			MAPE	82.784	101.755	142.707	42.736	136.211	32.212	36.458	51.625	45.230	74.456	26.624	23.481	49.404	40.781	30.398	32.429	40.423	35.353	35.615
D	3	P17 = f(P17,P16,P7A)	R2	0.477	0.812	0.843	0.696	0.845	0.835	0.692	0.815	0.814	0.824	0.841	0.833	0.831	0.835	0.849	0.842	0.854	0.815	0.848
			RMSE	231.324	130.077	110.020	161.358	143.313	115.308	152.294	120.284	123.499	126.161	113.495	114.068	115.483	114.945	109.114	110.065	108.344	121.865	110.429
			MAPE	33.041	75.122	27.601	33.088	75.736	57.274	130.556	77.975	72.536	96.250	43.041	80.708	42.914	50.964	29.392	28.015	33.442	47.765	41.422

ภาพที่ 3.20 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการ Random ครั้งที่ 2 คาดการณ์ล่วงหน้า 3 วันและแสดง Hidden Layer 20 node ของค่า R^2 RMSE และ MAPE

Random 3

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	3	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.863	0.063	0.835	0.673	0.804	0.852	0.812	0.842	0.760	0.855	0.875	0.852	0.785	0.793	0.852	0.871	0.868	0.861	0.856
			RMSE	139.207	267.217	106.467	166.122	116.588	101.854	112.818	100.672	123.435	104.520	88.704	97.006	127.457	114.719	96.800	89.975	92.672	93.845	95.294
			MAPE	111.521	269.317	94.944	68.814	75.267	40.675	73.163	47.607	46.331	62.211	30.080	33.129	49.041	52.472	31.456	36.503	28.165	32.865	34.729
B	3	P17 = f(P17,P15,P16,RF)	R2	0.796	0.802	0.850	0.835	0.791	0.854	0.822	0.856	0.842	0.809	0.833	0.851	0.866	0.856	0.865	0.844	0.858	0.855	0.864
			RMSE	157.452	112.201	99.188	156.540	121.996	106.152	106.347	96.893	102.685	111.964	102.605	97.670	91.983	95.788	92.554	99.185	94.627	95.877	92.658
			MAPE	103.411	51.947	72.838	140.003	78.557	56.960	31.023	36.665	85.280	38.260	41.732	52.763	28.970	52.149	28.832	31.650	37.397	29.343	31.001
C	3	P17= f(P17,P15,P16)	R2	0.749	0.699	0.788	0.828	0.825	0.747	0.816	0.820	0.803	0.778	0.826	0.830	0.825	0.823	0.819	0.798	0.802	0.820	0.827
			RMSE	152.631	141.389	134.173	108.472	109.819	128.526	112.169	110.134	115.568	121.420	108.009	108.434	119.806	108.296	109.052	116.003	114.102	110.102	109.251
			MAPE	194.026	144.190	81.692	35.890	43.740	92.220	77.157	29.075	70.980	39.091	27.291	34.443	51.299	46.486	46.882	44.718	48.116	28.421	50.073
D	3	P17 = f(P17,P16,P7A)	R2	0.643	0.736	0.774	0.773	0.738	0.812	0.804	0.808	0.832	0.828	0.814	0.844	0.801	0.820	0.838	0.847	0.759	0.824	0.839
			RMSE	428.490	145.726	134.058	124.545	130.635	113.137	114.136	123.602	106.281	119.225	112.580	103.307	117.439	110.163	103.180	102.454	125.555	109.352	105.296
			MAPE	368.103	101.560	146.952	41.653	45.604	68.614	34.676	72.430	30.508	71.297	25.793	40.410	90.575	45.603	26.747	37.860	29.875	61.426	41.885

ภาพที่ 3.21 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการ Random ครั้งที่ 3 คาดการณ์ล่วงหน้า 3 วันและแสดง Hidden Layer 20 node ของค่า R^2 RMSE และ

Average

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	3	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.841	0.304	0.842	0.785	0.837	0.838	0.850	0.790	0.813	0.835	0.857	0.855	0.822	0.832	0.850	0.868	0.856	0.853	0.852
			RMSE	125.550	220.704	116.313	127.273	107.225	146.962	103.848	119.226	113.944	109.258	99.988	101.248	116.092	109.056	101.985	96.381	100.580	101.690	102.176
			MAPE	98.162	192.856	97.928	51.729	56.089	83.858	49.805	73.805	40.670	55.439	36.783	48.520	55.875	56.042	36.193	34.778	52.869	38.000	34.556
B	3	P17 = f(P17,P15,P16,RF)	R2	0.432	0.394	0.577	0.843	0.819	0.737	0.832	0.846	0.858	0.835	0.834	0.840	0.845	0.859	0.863	0.848	0.857	0.856	0.860
			RMSE	227.634	224.727	155.731	149.226	114.083	152.285	113.939	104.382	100.913	107.593	107.450	105.968	104.204	99.948	98.398	103.685	100.444	100.449	99.237
			MAPE	143.422	169.474	110.761	114.723	53.428	99.107	46.735	40.691	49.417	41.881	60.089	57.974	33.053	42.691	31.322	37.419	37.972	30.429	32.240
C	3	P17= f(P17,P15,P16)	R2	0.823	0.709	0.527	0.841	0.695	0.821	0.833	0.845	0.836	0.831	0.849	0.842	0.852	0.845	0.838	0.836	0.844	0.842	0.847
			RMSE	154.633	153.830	182.068	107.365	150.687	131.958	108.982	106.109	108.686	116.857	104.517	109.201	110.454	105.547	107.467	108.038	105.606	106.200	105.387
			MAPE	136.771	121.907	90.306	49.155	87.363	71.716	54.856	45.298	50.220	50.066	27.661	43.201	44.634	49.172	37.091	45.624	43.762	30.009	40.568
D	3	P17 = f(P17,P16,P7A)	R2	0.670	0.776	0.835	0.786	0.820	0.839	0.762	0.841	0.840	0.832	0.848	0.848	0.842	0.825	0.862	0.861	0.834	0.845	0.855
			RMSE	249.959	136.044	120.546	129.631	122.800	109.134	154.090	111.649	109.301	116.951	106.754	104.741	106.779	112.407	98.960	101.492	107.635	106.136	102.490
			MAPE	147.619	79.767	74.837	42.026	65.298	58.523	91.012	59.606	55.128	72.076	35.476	57.576	53.210	50.189	29.132	32.205	36.040	45.322	38.328

ภาพที่ 3.22 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการเฉลี่ยจาก Random ทั้งหมด คาดการณ์ล่วงหน้า 3 วันและแสดง Hidden Layer 20 node ของค่า R^2 RMSE และ MAPE

Random 1

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	4	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.784	0.026	0.797	0.723	0.817	0.823	0.825	0.809	0.832	0.817	0.834	0.834	0.808	0.819	0.812	0.827	0.848	0.836	0.797
			RMSE	124.327	246.835	135.085	134.899	107.639	108.647	105.739	112.469	102.527	108.280	103.361	104.659	113.841	109.727	109.592	104.285	99.457	103.696	116.446
			MAPE	72.274	159.361	74.520	99.023	52.591	34.156	40.380	79.728	37.035	49.098	39.532	45.176	74.067	74.725	43.652	39.354	33.196	43.410	70.436
B	4	P17 = f(P17,P15,P16,RF)	R2	0.812	0.726	0.796	0.773	0.822	0.815	0.814	0.821	0.816	0.801	0.829	0.814	0.813	0.820	0.817	0.805	0.818	0.808	0.817
			RMSE	108.479	137.639	139.825	119.831	111.444	108.704	108.673	105.774	118.281	111.555	104.289	108.729	109.689	108.988	108.483	114.788	108.184	112.595	107.298
			MAPE	50.851	70.097	99.225	65.393	38.827	61.291	48.393	52.239	71.016	33.162	36.128	37.848	48.348	38.266	42.423	69.675	56.224	49.968	53.301
C	4	P17= f(P17,P15,P16)	R2	0.771	0.812	0.822	0.811	0.817	0.831	0.806	0.822	0.829	0.819	0.816	0.810	0.824	0.798	0.827	0.818	0.831	0.808	0.797
			RMSE	123.914	114.413	108.969	115.616	114.864	106.567	114.908	136.888	107.430	112.338	120.368	113.422	109.968	123.117	108.393	110.811	106.240	117.376	119.758
			MAPE	76.869	49.623	35.553	36.441	37.161	40.234	80.283	79.485	35.289	43.735	42.789	51.960	47.981	42.077	45.840	51.530	39.460	48.358	70.378
D	4	P17 = f(P17,P16,P7A)	R2	0.692	0.818	0.390	0.791	0.831	0.750	0.820	0.832	0.831	0.819	0.829	0.837	0.835	0.830	0.799	0.787	0.827	0.834	0.840
			RMSE	144.104	140.984	243.043	124.100	106.545	134.371	111.098	109.455	107.051	111.283	110.772	113.460	105.902	106.609	120.831	119.217	107.371	105.694	103.335
			MAPE	95.779	96.362	96.457	73.207	41.731	82.446	69.675	58.220	39.224	46.915	35.638	39.082	38.745	38.495	56.126	44.527	30.846	39.531	32.324

ภาพที่ 3.23 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการ Random ครั้งที่ 1 คาดการณ์ล่วงหน้า 4 วันและแสดง Hidden Layer 20 ครั้ง ของค่า R^2 RMSE และ MAPE

Random 2

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	4	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.741	0.731	0.659	0.761	0.736	0.730	0.761	0.702	0.751	0.743	0.749	0.756	0.754	0.689	0.769	0.721	0.751	0.785	0.751
			RMSE	159.288	150.816	165.611	141.659	146.515	146.458	138.905	153.964	145.526	144.642	141.057	138.963	141.387	157.269	136.064	148.675	140.243	133.097	141.219
			MAPE	58.297	47.662	124.483	43.519	51.280	57.240	60.112	96.958	59.310	66.401	58.475	62.086	53.609	74.644	39.345	88.764	41.716	48.620	31.332
B	4	P17 = f(P17,P15,P16,RF)	R2	0.733	0.736	0.713	0.720	0.739	0.751	0.746	0.749	0.745	0.750	0.735	0.759	0.755	0.729	0.699	0.749	0.772	0.773	0.773
			RMSE	145.729	147.576	167.790	164.730	143.654	144.827	143.192	141.098	144.497	141.223	145.072	139.925	141.356	147.483	154.222	142.392	136.565	135.686	135.568
			MAPE	38.442	53.878	86.618	182.234	63.937	45.804	55.495	40.047	49.159	41.819	64.157	34.963	55.300	70.345	127.249	75.330	59.370	34.785	55.997
C	4	P17= f(P17,P15,P16)	R2	0.733	0.760	0.753	0.726	0.004	0.684	0.729	0.757	0.761	0.754	0.747	0.770	0.768	0.749	0.750	0.743	0.751	0.768	0.771
			RMSE	144.316	144.591	146.033	151.745	285.284	164.671	145.952	140.208	136.917	139.684	142.258	137.455	138.053	141.635	141.885	144.121	141.200	137.948	139.499
			MAPE	57.577	63.396	50.991	128.217	122.750	139.352	84.457	71.985	33.319	56.267	69.451	46.642	44.583	66.702	82.029	38.478	53.970	51.581	48.563
D	4	P17 = f(P17,P16,P7A)	R2	0.697	0.758	0.731	0.759	0.634	0.762	0.646	0.771	0.774	0.760	0.730	0.760	0.740	0.718	0.758	0.766	0.776	0.751	0.781
			RMSE	155.022	167.938	145.252	138.812	206.960	140.239	165.280	138.232	136.534	137.258	147.469	139.517	142.855	150.161	138.557	138.491	134.209	140.101	135.016
			MAPE	47.562	86.460	43.324	44.680	80.474	34.326	120.578	68.780	56.358	37.252	109.619	56.402	61.673	89.788	41.007	58.216	49.828	45.690	45.798

ภาพที่ 3.24 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการ Random ครั้งที่ 2 คาดการณ์ล่วงหน้า 4 วันและแสดง Hidden Layer 20 ครั้ง ของค่า R^2 RMSE และ MAPE

Random 3

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	4	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.740	0.000	0.797	0.744	0.714	0.791	0.728	0.755	0.820	0.740	0.785	0.764	0.819	0.825	0.790	0.793	0.822	0.815	0.802
			RMSE	128.945	254.243	123.536	126.030	132.225	113.086	129.495	122.738	104.554	127.951	114.667	120.254	104.711	103.278	114.783	111.991	104.114	106.268	109.831
			MAPE	68.016	109.717	81.207	103.495	120.725	37.288	52.937	79.649	31.566	50.163	38.855	34.951	31.589	50.683	38.087	33.337	41.331	50.900	56.589
B	4	P17 = f(P17,P15,P16,RF)	R2	0.739	0.805	0.786	0.777	0.753	0.791	0.750	0.816	0.802	0.819	0.784	0.800	0.789	0.811	0.764	0.790	0.802	0.818	0.803
			RMSE	296.432	109.452	119.184	117.255	122.471	112.582	131.274	106.208	113.634	111.407	114.807	111.180	113.237	107.356	119.687	113.180	110.253	105.497	109.359
			MAPE	222.048	38.277	47.062	73.719	96.877	42.750	95.833	34.675	54.009	53.470	74.877	60.117	61.326	60.458	76.836	43.705	44.881	35.552	34.467
C	4	P17= f(P17,P15,P16)	R2	0.729	0.765	0.710	0.649	0.781	0.001	0.764	0.763	0.752	0.750	0.780	0.778	0.781	0.784	0.766	0.780	0.756	0.738	0.774
			RMSE	145.664	151.223	155.356	178.196	129.984	253.785	126.646	123.620	126.035	125.944	120.015	119.542	119.231	118.099	122.108	119.691	124.853	129.175	120.546
			MAPE	90.886	103.574	117.587	89.832	73.100	132.286	101.447	57.048	95.108	35.775	43.101	40.602	39.317	37.444	32.678	34.199	52.035	54.560	46.880
D	4	P17 = f(P17,P16,P7A)	R2	0.799	0.051	0.109	0.763	0.770	0.788	0.780	0.782	0.773	0.781	0.746	0.743	0.781	0.807	0.789	0.782	0.731	0.791	0.768
			RMSE	222.279	253.841	254.238	124.452	121.384	115.656	151.281	134.368	120.875	119.469	128.077	127.081	119.358	110.447	117.266	135.350	130.383	116.086	122.489
			MAPE	167.513	133.639	155.530	58.477	44.952	40.627	95.327	81.111	65.527	60.470	48.378	32.706	58.272	31.075	48.014	69.333	53.879	43.050	53.460

ภาพที่ 3.25 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการ Random ครั้งที่ 3 คาดการณ์ล่วงหน้า 4 วันและแสดง Hidden Layer 20 ครั้ง ของค่า R^2 RMSE และ MAPE

Average

Case	TOF	Model		Number Of Node																		
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	4	P17 = f(P17,P15,P16,P7A,P26A,RF)	R2	0.755	0.252	0.751	0.743	0.756	0.781	0.771	0.755	0.801	0.767	0.789	0.785	0.794	0.778	0.790	0.780	0.807	0.812	0.783
			RMSE	137.520	217.298	141.411	134.196	128.793	122.730	124.713	129.724	117.536	126.958	119.695	121.292	119.980	123.425	120.146	121.650	114.605	114.353	122.498
			MAPE	66.196	105.580	93.403	82.012	74.865	42.895	51.143	85.445	42.637	55.221	45.621	47.404	53.088	66.684	40.361	53.818	38.768	47.643	52.786
B	4	P17 = f(P17,P15,P16,RF)	R2	0.761	0.756	0.765	0.756	0.771	0.786	0.770	0.796	0.788	0.790	0.783	0.791	0.786	0.786	0.760	0.781	0.797	0.800	0.798
			RMSE	183.547	131.556	142.266	133.939	125.856	122.038	127.713	117.693	125.470	121.395	121.389	119.945	121.427	121.276	127.464	123.454	118.334	117.926	117.408
			MAPE	103.780	54.084	77.635	107.115	66.547	49.948	66.574	42.320	58.061	42.817	58.387	44.309	54.991	56.356	82.169	62.903	53.492	40.102	47.922
C	4	P17= f(P17,P15,P16)	R2	0.744	0.779	0.762	0.729	0.534	0.505	0.766	0.781	0.781	0.774	0.781	0.786	0.791	0.777	0.781	0.780	0.779	0.771	0.781
			RMSE	137.964	136.742	136.786	148.519	176.711	175.007	129.169	133.572	123.461	125.989	127.547	123.473	122.417	127.617	124.129	124.874	124.098	128.166	126.601
			MAPE	75.111	72.198	68.044	84.830	77.670	103.957	88.729	69.506	54.572	45.259	51.780	46.401	43.960	48.741	53.516	41.402	48.488	51.500	55.274
D	4	P17 = f(P17,P16,P7A)	R2	0.729	0.542	0.410	0.771	0.745	0.767	0.749	0.795	0.793	0.787	0.768	0.780	0.785	0.785	0.782	0.778	0.778	0.792	0.796
			RMSE	173.801	187.588	214.178	129.121	144.963	130.089	142.553	127.352	121.487	122.670	128.773	126.686	122.705	122.406	125.551	131.019	123.988	120.627	120.280
			MAPE	103.618	105.487	98.437	58.788	55.719	52.466	95.193	69.370	53.703	48.212	64.545	42.730	52.897	53.119	48.382	57.359	44.851	42.757	43.861

ภาพที่ 3.26 แสดงผลการ Run Code MLR Regressor Validation ซึ่งในรูปจะเป็นการเฉลี่ยจาก Random ทั้งหมด ในการคาดการณ์ล่วงหน้า 4 วันและแสดง Hidden Layer 20 ครั้ง ของค่า R^2 RMSE และ MAPE

ตารางที่ 3.1 สรุปผลค่าเฉลี่ยของ Randomในแต่ละ Case พร้อมทั้งแสดงค่า R^2 , RMSE , MAPE

CaseA	Avg			
TOF	R2	RMSE	MAPE	**Structure**
1	0.979	38.240	38.240	*6-14-1*
2	0.935	67.440	25.742	*6-20-1*
3	0.868	96.381	34.556	*6-17-1*
4	0.812	114.353	37.748	*6-19-1*
Case B	Avg			
TOF	R2	RMSE	MAPE	**Structure**
1	0.969	48.449	17.857	*4-20-1*
2	0.922	74.191	25.623	*4-17-1*
3	0.863	98.398	30.429	*4-16-1*
4	0.800	117.408	40.102	*4-20-1*
Case C	Avg			
TOF	R2	RMSE	MAPE	**Structure**
1	0.9633	49.531	23.9780	*3-14-1*
2	0.9063	80.391	21.9647	*3-18-1*
3	0.8520	104.517	27.6613	*3-12-1*
4	0.7860	122.417	41.4020	*3-13-1*
Case D	Avg			
TOF	R2	RMSE	MAPE	**Structure**
1	0.969	48.257	17.756	*3-15-1*
2	0.914	78.917	24.918	*3-19-1*
3	0.862	98.960	29.132	*3-16-1*
4	0.796	120.280	42.730	*3-20-1*

3.2.2.3 โค้ดภาษา Python สำหรับพล็อตกราฟเลือกโมเดลที่ดีที่สุดในแบบจำลอง โดยการนำจากค่าเฉลี่ยที่ทั้งหมดที่ทำการพยากรณ์ 1-3 วัน

```
import matplotlib.pyplot as plt
plt.figure(figsize=(14,3))
plt.rcParams.update({'font.family':'Palatino Linotype'})
plt.rcParams['font.size'] = '13'
# plt.title('Statistical values of the MLR, Model A
(Validation period 20%)' , fontsize=12)
ax1 = plt.subplot2grid(shape=(1,2), loc=(0,0), colspan=1)
ax2 = plt.subplot2grid((1,2), (0,1), colspan=1)
# ax3 = plt.subplot2grid((1,3), (0,2), colspan=1)

plt.show()

color1 = 'red'
color2 ='dodgerblue'
color3 = 'mediumseagreen'
color4 = 'Darkorange'
ax1.plot([1,2,3,4], [0.979,0.935,0.868,0.812],color=color1,
label="Case A")
ax1.plot([1,2,3,4], [0.969,0.922,0.863,0.800],color=color2,
label="Case B")
ax1.plot([1,2,3,4], [0.963,0.906,0.852,0.786],color=color3,
label="Case C")
ax1.plot([1,2,3,4], [0.969,0.914,0.862,0.796],color=color4,
label="Case D")
ax1.grid(True)

ax1.set_xlabel("Forecast period (hrs)", fontsize=14)
ax1.set_ylabel("R2 ", fontsize=14)
ax1.set_xticks([1,2,3,4])
ax1.set_xlim(1, 4)
ax1.set_ylim(0.7, 1)
ax1.legend(loc="lower right", fontsize=12,ncol=2)
# ax1.set_title('MLR, All water level,AWL')
ax1.tick_params(axis='both', which='major', labelsize=14)
# ax1.text(1.5, 120, '(A) ANN', fontsize=14)
ax1.patch.set_facecolor('lightcyan')
ax1.patch.set_alpha(0.5)
```

ภาพที่ 3.27 คำสั่งแสดงการพล็อตกราฟเลือกโมเดลของแบบจำลอง

บรรทัดที่ 1-2	หมายถึง	การนำเข้าแพคเกจ matplotlib.pyplot สำหรับการสร้างกราฟและกำหนดขนาดของกราฟ
บรรทัดที่ 3-4	หมายถึง	เลือกชนิดและขนาด font ที่จะแสดงในกราฟ
บรรทัดที่ 7-8	หมายถึง	ระบุตำแหน่งกราแสดงผลของกราฟแต่ละอันที่ต้องการ
บรรทัดที่ 13-16	หมายถึง	กำหนดสีที่จะใช้ในการแสดงของกราฟ
บรรทัดที่ 17-24	หมายถึง	นำเข้าข้อมูลค่าเฉลี่ยที่ได้จากการคำนวณโดยเลือกโมเดลที่ดีที่สุดมาใส่และกำหนดให้เป็นแต่ Case
บรรทัดที่ 27-29	หมายถึง	กำหนดชื่อแกนของกราฟและขนาดตัวอักษร
บรรทัดที่ 30-31	หมายถึง	กำหนดขอบเขตของกราฟที่จะแสดง
บรรทัดที่ 32	หมายถึง	ระบุตำแหน่งชื่อของกราฟ
บรรทัดที่ 34-37	หมายถึง	กำหนดขนาดของชื่อกราฟและสีพื้นหลังของกราฟ

```
38 ax2.plot([1,2,3,4],
39 [38.240,67.440,96.381,114.353],color=color1,label="CaseA")
40 ax2.plot([1,2,3,4],
41 [48.449,74.191,98.398,117.408],color=color2,label="CaseB")
42 ax2.plot([1,2,3,4],
43 [49.531,80.391,104.517,122.417],color=color3label="CanseC")
44 ax2.plot([1,2,3,4],
45 [48.257,78.917,98.960,120.280],color=color4,label="CaseD")
46 ax2.grid(True)
47 ax2.grid(True)
48 ax2.set_xlabel("Forecast period (hrs)", fontsize=14)
49 ax2.set_ylabel("RMSE (CMS)", fontsize=14)
50 ax2.set_xticks([1,2,3,4])
51 ax2.set_xlim(1, 4)
52 ax2.set_ylim(40, 120)
53 ax2.legend(loc="lower right", fontsize=12,ncol=2)
54 # ax2.set_title('MLR, Low water level,LWL')
55 ax2.tick_params(axis='both', which='major',labelsize=14)
56 # ax2.text(26, 0.11, '(B) MLR, Low water level,LWL',
57 fontsize=14)
58 ax2.patch.set_facecolor('azure')
59 ax2.patch.set_alpha(0.15)
60 plt.tight_layout()
```

ภาพที่ 3.28 คำสั่งแสดงการพล็อตกราฟเลือกโมเดลของแบบจำลอง

บรรทัดที่ 38-45	หมายถึง	นำเข้าข้อมูลจากค่าเฉลี่ยที่ได้จากการคำนวณทั้งหมดมาใส่และกำหนดให้เป็นแต่ Case
บรรทัดที่ 47-48	หมายถึง	กำหนดชื่อแกนของกราฟและขนาดตัวอักษร
บรรทัดที่ 50-52	หมายถึง	กำหนดขอบเขตของกราฟที่จะแสดง
บรรทัดที่ 53	หมายถึง	ระบุตำแหน่งชื่อของกราฟ
บรรทัดที่ 55-58	หมายถึง	กำหนดขนาดของชื่อกราฟและสีพื้นหลังของกราฟ
บรรทัดที่ 60	หมายถึง	แสดงรูปกราฟออกมาให้อยู่ให้ในขอบเขตพอดี

3.2.2.4 โค้ดภาษา Python สำหรับการทดสอบข้อมูลที่จะพยากรณ์โดยใช้ ANNs

```
1  import numpy as np
2  import pandas as pd
3  #Load the data
4  data='D:/Project75 Data/Regression/TotalX.csv'
5  Data=pd.read_csv(data,header=None)
6
7  #Change Type of Column [0] from object to Datetime64
8  Data[0]=pd.to_datetime(Data[0])
9  Data=Data.set_index(0)
10
11 Data.insert(1, "P17", Data[1], True)
12 Data.columns = ['P17t', 'P17', 'P15', 'P16',
13 'P7A','P26A','RF']
14
15 #Add X's time with delta_hr
16 dh_x1=1
17 dh_x2=1
18 dh_x3=1
19 dh_x4=1
20 dh_x5=1
21 dh_x6=1
22 dh_x7=2
23
24 X1=Data['P17t'].shift(dh_x1, freq='D')
25 X2=Data['P17'].shift(dh_x2, freq='D')
26 X3=Data['P15'].shift(dh_x3, freq='D')
27 X4=Data['P16'].shift(dh_x4, freq='D')
28 X5=Data['P7A'].shift(dh_x5, freq='D')
29 X6=Data['P26A'].shift(dh_x6, freq='D')
30 X7=Data['RF'].shift(dh_x7, freq='D')
31 Y=Data['P17']
```

ภาพที่ 3.29 แสดงโค้ดที่จะใช้ในการทดสอบชุดข้อมูลที่ดีที่สุด

บรรทัดที่ 1-2	หมายถึง	นำแพคเกจ numpy เป็นแพคเกจที่ใช้ในการคำนวณทางคณิตศาสตร์ Pandas เป็นแพคเกจสำหรับการจัดการข้อมูลที่ทำให้จัดข้อมูลได้ง่าย
บรรทัดที่ 4-5	หมายถึง	นำเข้าข้อมูลไฟล์.csv จากแหล่งที่อยู่ในเครื่อง
บรรทัดที่ 11	หมายถึง	เพิ่มคอลัมน์ P.17 ไว้หน้าคอลัมน์ที่มีอยู่

บรรทัดที่ 12	หมายถึง	สร้างคอลัมน์ P.17t , P.17 , P.15 , P.16 , P.7A , P.26A , RF
บรรทัดที่ 16-22	หมายถึง	เป็นการกำหนดว่าข้อมูลแต่ละตัวนั้นพยากรณ์กี่วัน
บรรทัดที่ 24-31	หมายถึง	แทนค่าชุดข้อมูลด้วยตัวแปรโดยให้หน่วยข้อมูลเป็นวัน

```
merge=pd.merge(Y,X1,how='inner',left_index=True,right_index=True)
merge=pd.merge(merge,X2,how='inner',left_index=True,right_index=True)
merge=pd.merge(merge,X3,how='inner',left_index=True,right_index=True)
merge=pd.merge(merge,X4,how='inner',left_index=True,right_index=True)
merge=pd.merge(merge,X5,how='inner',left_index=True,right_index=True)
merge=pd.merge(merge,X6,how='inner',left_index=True,right_index=True)
merge=pd.merge(merge,X7,how='inner',left_index=True,right_index=True)
merge.columns =['P17t0','P17t','P17','P15', 'P16', 'P7A','P26A','RF']
merge=merge.dropna()

# Select data for testing Apr2020 and Oct2020
Vartest_L=merge.loc['2020-04-01':'2020-04-30']
Vartest_H=merge.loc['2020-10-01':'2020-10-31']
Vartest = pd.concat([Vartest_L , Vartest_H])

merge=merge.loc[(merge.index < '2020-04-01') |
(merge.index >= '2020-05-01')]
merge=merge.loc[(merge.index < '2020-10-01') |
(merge.index >= '2020-11-01')]

xtest = Vartest.iloc[:,lambda Vartest:[1,2,3,4,5,6,7]].astype(float)
ytest = Vartest.iloc[:,lambda Vartest:[0]].astype(float)
ytest.columns = ['Yobs_test']

X = merge.iloc[:,lambda merge:[1,2,3,4,5,6,7]].astype(float)
Y = merge.iloc[:,0].astype(float) # output variable
(what we are trying to predict)

Y=Y.to_frame()
Y.columns = ['P17']
```

<u>ภาพที่ 3.30</u> แสดงโค้ดที่จะใช้ในการทดสอบชุดข้อมูลที่ดีที่สุด

บรรทัดที่ 33-34	หมายถึง	ทำการ merge dataframe ทุก dataframe โดยใช้Dataframe Index ที่เหมือนกัน
บรรทัดที่ 40-41	หมายถึง	จะทำการ merge คอลัมน์ดังในโค้ดและทำการ drop แถวที่มีค่า Nan

บรรทัดที่ 44-46	หมายถึง	เลือกค่าที่จะทำการทดสอบในช่วงเดือน เมษายนและช่วงเดือนตุลาคมโดยให้ Vartest L เป็นช่วงเดือนเมษายนแล้วจะทำการmerge ข้อมูลในเดือนนั้น และให้ Vartest H เป็นช่วงเดือนคุลาคมแล้วทำการ merge ข้อมูลในเดือนนั้น
บรรทัดที่ 48-49	หมายถึง	ใช้คำสั่ง Vartest_L (ช่วงปริมาณน้ำน้อย) เลือกข้อมูลสำหรับทดสอบ โดยใช้ข้อมูลตั้งแต่ 1 เมษายน 2020 ถึง 1 พฤษภาคม 2020
บรรทัดที่ 50-51	หมายถึง	ใช้คำสั่ง Vartest_H (ช่วงปริมาณน้ำมาก) เลือกข้อมูลสำหรับทดสอบโดยใช้ข้อมูลตั้งแต่ 1 ตุลาคม 2020 ถึง 1 พฤศจิกายน 2020
บรรทัดที่ 53-55	หมายถึง	ทำการสร้าง Dataframe, Xtest จากคอลัมน์ที่ 1,2,3,4,5,6 และ Ytest จากคอลัมน์ที่ 0 โดย Dataframe ใหม่จากคอลัมน์ Ytest ชื่อว่า Yobs_test
บรรทัดที่ 57-58	หมายถึง	ทำการเลือกตัวแปร X1,X2,X3,X4,X5,X6,X7 จาก Dataframe เลือกคอลัมน์ที่ 1,2,3,4,5,6,7 และตัวแปร Y เลือกคอลัมน์ 0
บรรทัดที่ 61-62	หมายถึง	หลังจากที่ merge เสร็จสิ้น ให้ Y เป็นคอลัมน์ P17

```
64 ### Normalization of the data ###
65 from sklearn.preprocessing import MinMaxScaler
66 scaler = MinMaxScaler(feature_range=(0, 1))
67 scalerX = scaler.fit_transform(X.values)
68 scalerX = pd.DataFrame(scalerX, index=X.index,
69 columns=X.columns)
70 scalerY = scaler.fit_transform(Y.values)
71 scalerY = pd.DataFrame(scalerY, index=X.index,
72 columns=Y.columns)
73 from sklearn.metrics import mean_squared_error
74 # Function to calculate R2, RMSE, EI, and MAPE
75 def Cal_R2(Yobs_Ycal): # Yobs_Ycal must be dataframe 2
76 columns
77     R = Yobs_Ycal.corr(method="pearson")
78     R = R.iloc[0,1]
79     R2 = round(R**2,6)
80     print("R2 =" , R2)
81     return R2
82 def Cal_rmse(Yobs,Ycal):
83     mse = mean_squared_error(Yobs, Ycal)
84     Rmse = mse**0.5
85     print("RMSE =" , round(Rmse,6))
86     return Rmse
87 def Cal_ei(Yobs,Ycal):
88     avg_Yobs = sum(Yobs) / len(Yobs)
89     st= sum((Yobs-(avg_Yobs))**2)
90     se= sum((Yobs-Ycal)**2)
91     sr= st-se
92     Ei= round(sr/st,6)
93     # print("EI =" , Ei)
94     return Ei
95 def Cal_mape(Yobs,Ycal):
96     Mape = round(np.mean(np.abs((Yobs - Ycal)/Yobs))*100,6)
97     print("MAPE =" , Mape)
98     return Mape
99 def nse(Yobs,Ycal):
100     global NSE
101     NSE = round(1-(np.sum((Yobs-Ycal)**2)/np.sum((Yobs-
102 np.mean(Ycal))**2)),4)
103     print("NSE =" , NSE)
104     return NSE
```

ภาพที่ 3.31 แสดงโค้ดที่จะใช้ในการทดสอบชุดข้อมูลที่ดีที่สุด

บรรทัดที่ 64	หมายถึง	ใช้คำสั่ง Normalization เพื่อให้ข้อมูลไม่มีการคลาดเคลื่อนให้ข้อมูลเป็นปกติก่อนที่จะทำการนำเข้ากระบวนการ

บรรทัดที่ 65	หมายถึง	นำเข้าชุดคำสั่ง sklearn preprocessing ในขั้นตอนของ Data Analysis หรือการเข้าใจและวิเคราะห์ข้อมูล อาจต้องมีการแก้ไขให้ข้อมูลอยู่ในรูปแบบที่เรานำไปใช้งานได้ เครื่องมือตัวนี้จะสามารถช่วยจัดการกับข้อมูลได้
บรรทัดที่ 66	หมายถึง	กำหนดช่วงข้อมูลค่ามากสุดน้อยสุดให้อยู่ในช่วง 0-1
บรรทัดที่ 67-72	หมายถึง	ให้คำนวณทางสถิติทาง scaler X , scaler Y
บรรทัดที่ 73	หมายถึง	นำเข้าชุดคำสั่ง sklearn metric มาคำนวณค่าทางสถิติ
ของ		R^2 , RMSE , EI และ MAPE
บรรทัดที่ 75-81	หมายถึง	คำนวณค่า R^2 (Coefficient of determination)โดยคำนวณจาก Ycal และ Yobs
บรรทัดที่ 82-86	หมายถึง	คำนวณค่า Rmse (Root mean square error) โดยคำนวณจาก Ycal และ Yobs
บรรทัดที่ 87-94	หมายถึง	คำนวณค่า Ei(ค่าประสิทธิภาพจากแบบจำลอง Efficiency Index) โดยคำนวณจาก Ycal และ Yobs
บรรทัดที่ 95- 98	หมายถึง	คำนวณค่า Mape (Mean absoulte percent error) โดยคำนวณจาก Ycal และ Yobs
บรรทัดที่ 99-104	หมายถึง	คำนวณค่า NSE (Nash Sutcliffe efficiency)) โดยคำนวณจาก Ycal และ Yobs

```
109 # Import ANNs
110 from sklearn.neural_network import MLPRegressor
111 # Create lists for collecting the R2, RMSE,
112 MAPE (Validation period)
113 A =[]
114 # compile the model
115 Ann=MLPRegressor(hidden_layer_sizes=14,random_state=1,
116 verbose=1,batch_size=16).fit(scalerX,scalerY)
117 import pickle
118 # Save to file in the current working directory
119 pkl_filename = "pickle_model.pkl"
120 with open(pkl_filename, 'wb') as file:
121     pickle.dump(Ann, file)
122
123 # Load from file
124 with open(pkl_filename, 'rb') as file:
125     pickle_model = pickle.load(file)
126 # Calculate the accuracy score and predict target values
127 Ypredict = pickle_model.predict(xtest)
128 Ypredict = pd.DataFrame(Ypredict)
129 Ypredict.index = ytest.index
130 Y_Ans = pd.merge(ytest,Ypredict,how='inner',left_index=True
131 ,right_index=True)
132 Y_Ans.columns = ['Yobs_test', 'Ycal']
133
134 R2 = Cal_R2(Y_Ans) # Calculate R2
135 Rmse= Cal_rmse(Y_Ans['Yobs_test'],Y_Ans['Ycal'])#Calculate
136 Root mean square error (RMSE)
137 Mape= Cal_mape(Y_Ans['Yobs_test'],Y_Ans['Ycal'])#Calculate
138 Mean absolute percent error (MAPE)
139 nse = nse(Y_Ans['Yobs_test'],Y_Ans['Ycal'])
140 A.append((R2,Rmse,Mape,nse))
141 df1 = pd.DataFrame(A)
142 df1.columns=['R2','Rmse','Mape','nse',]df1=pd.DataFrame(A)
143 df1.columns=['R2','Rmse','Mape','nse']
144 df1.to_csv('D:/Project75 Data/Test Data/Test_1.csv')
```

ภาพที่ 3.32 แสดงโค้ดที่จะใช้ในการทดสอบชุดข้อมูลที่ดีที่สุด

บรรทัดที่ 109-110	หมายถึง	นำเข้าชุดคำสั่ง ANNs มาวิเคราะห์การประมวลผลข้อมูลจากขั้นตอนที่ทำ MLP
บรรทัดที่ 115	หมายถึง	จากที่ทำ MLP ของชุดข้อมูลนี้ผลสรุปมาได้ว่า Node ตัวที่ดีสุดคือ ครั้งที่ 14 ดังนั้น ANNs จะประมวลผลใน Hidden Layer ครั้งนี้

บรรทัดที่ 117-118	หมายถึง	ทำการ Save Model ที่ Run ได้เป็นค่า Xtest ไว้ในตัวไฟล์ที่จัดเก็บ
บรรทัดที่ 119-121	หมายถึง	นำข้อมูล Model ที่ Run ไว้มาใช้ในการพยากรณ์ต่อไป
บรรทัดที่ 124-132	หมายถึง	นำข้อมูลที่ Xtest ที่จัดเก็บไว้มาคำนวณใส่ไว้ใน Y predict โดยให้ ANNs คำนวณ ค่าที่ได้ออกมาจะเป็นค่าพยากรณ์ที่ถูกต้อง
บรรทัดที่ 134-138	หมายถึง	คำนวณค่าทางสถิติโดย ANNs ค่าสถิติที่ได้จะมีความแม่นยำมากขึ้น
บรรทัดที่ 142-144	หมายถึง	Save ข้อมูลเป็นคอลัมน์แยกค่าสถิติแต่ละตัวไว้ และไฟล์เป็น .csv

3.2.2.5 โค้ดภาษา Python สำหรับกราฟ Time series plot observed and forecasted and Scatter plot Observed and Forecasted

```
1  import numpy as np
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  import math
5  # import hydroeval as he
6  #Load the data
7  data='D:/Project75 Data/Test Data/Timeseries total.csv'
8
9  TimeSeries=pd.read_csv(data,header=None)
10 TimeSeries.columns = ['Yobs','Day1','Day2','Day3','Day4']
11
12 TimeSeries_1=TimeSeries[['Yobs','Day1']].copy()
13 TimeSeries_2=TimeSeries[['Yobs','Day2']].copy()
14 TimeSeries_3=TimeSeries[['Yobs','Day3']].copy()
15 TimeSeries_4=TimeSeries[['Yobs','Day4']].copy()
16
17 TimeSeries_1.columns = ['Yobs', 'Day1']
18 TimeSeries_2.columns = ['Yobs', 'Day2']
19 TimeSeries_3.columns = ['Yobs', 'Day3']
20 TimeSeries_4.columns = ['Yobs', 'Day4']
```

ภาพที่ 3.33 แสดง Code สร้างกราฟความสัมพันธ์ Timeseries

บรรทัดที่ 1-4	หมายถึง	การนำเข้าแพคเกจที่จำเป็นสำหรับการใช้งาน ได้แก่ pandasเป็นสำหรับจัดการข้อมูล , numpy เป็นการคำนวณทางคณิตศาสตร์ , Matplotlib.pyplot เป็นแพคเกจสำหรับการสร้างกราฟและ แพคเกจ math เป็นฟังก์ชันที่ใช้คำนวณทางคณิตศาสตร์ เพื่อหาค่าทางคณิตศาสตร์ โดยฟังก์ชัน math ใน Python
บรรทัดที่ 7	หมายถึง	นำเข้าไฟล์ข้อมูล .csv จากคอมพิวเตอร์
บรรทัดที่ 9-10	หมายถึง	อ่านข้อมูลจากไฟล์ .csv โดยให้อ่านเป็นคอลัมน์ตาม code

บรรทัดที่ 12-15	หมายถึง	ให้ Timeseries แต่ละตัวอ่านค่า Yobs , Day ของแต่ละตัว
บรรทัดที่ 17-20	หมายถึง	คอลัมน์ Timeseries แต่ละตัวประกอบด้วย Yobs ,Day ของแต่ละตัว

```
from sklearn.metrics import mean_squared_error
def Cal_R2(Yobs_Ycal):# Yobs_Ycal must be
dataframe 2 columns global R2
    R = Yobs_Ycal.corr(method="pearson")
    R = round(R.iloc[0,1],4)
    R2 = round(R**2,4)
    # print("R =" , R)
    print("R\u00b2 =" , R2)
    return R2

def Cal_ei(Yobs,Ycal):
    global Ei
    avg_Yobs = sum(Yobs) / len(Yobs)
    st= sum((Yobs-(avg_Yobs))**2)
    se= sum((Yobs-Ycal)**2)
    sr= st-se
    Ei= round(sr/st,4)
    print("EI =" , Ei)
    return Ei

def Cal_rmse(Yobs,Ycal):
    global RMSE
    mse = mean_squared_error(Yobs, Ycal)
    Rmse = mse**0.5
    print("RMSE =" , round(Rmse,4))
    return Rmse
#Kling-Gupta Efficiency
# kge, r, alpha, beta = he.evaluator(he.kge,
TimeSeries_1['Yobs'],TimeSeries_1['Model1'])

def nse(targets,predictions):
    global NSE
    NSE = round(1-(np.sum((targets-predictions)**2)
/np.sum((targets-np.mean(predictions))**2)),4)
    print("NSE =" , NSE)
    return NSE

def rmse(Yobs,Ycal):
    global RMSE
    MSE = np.square(np.subtract(Yobs, Ycal)).mean()
    RMSE = round(math.sqrt(MSE),3)
    print("RMSE =" , RMSE, "m\u00b3\s " )
    return RMSE
```

ภาพที่ 3.34 แสดง Code สร้างกราฟความสัมพันธ์ Timeseries

บรรทัดที่ 22	หมายถึง	การนำเข้าแพคเกจจที่จำเป็นสำหรับการใช้งาน ได้แก่ Skelearn.metric ของ R^2,RMSE,EI,MAPE
บรรทัดที่ 23-30	หมายถึง	คำนวณค่า R^2 (Coefficient Of Determination) โดยคำนวณจากค่า Ycal และ Yobs
บรรทัดที่ 32-40	หมายถึง	คำนวณค่า EI (Efficiency Index) โดยคำนวณจากค่า Ycal และ Yobs
บรรทัดที่ 42-47	หมายถึง	คำนวณค่า RMSE (Root Mean Square Error) โดยคำนวณจากค่า Ycal และ Yobs
บรรทัดที่ 53-58	หมายถึง	คำนวณค่า NSE (Nash Sutcliffe Efficiency)

```
66 #%%
67 Lists = [TimeSeries_1, TimeSeries_2, TimeSeries_3,
68 TimeSeries_4]
69 A,B,C =[],[],[]
70 for i in range(0, len(Lists)):
71     R2 = Cal_R2(Lists[i])
72     Ei = Cal_ei(Lists[i].iloc[:,0], Lists[i].iloc[:,1])
73     Rmse=Cal_rmse(Lists[i].iloc[:,0],
74 Lists[i].iloc[:,1])
75     A.append(R2)
76     B.append(Ei)
77     C.append(Rmse)
78 #%%
79 import matplotlib.gridspec as gridspec
80 import matplotlib.pyplot as plt
81 import numpy as np
82 plt.figure(figsize=(20,10))
83 ax1 = plt.subplot2grid(shape=(4,3), loc=(0,0),colspan=2)
84 ax2 = plt.subplot2grid(shape=(4,3), loc=(0,2),colspan=1)
85 ax3 = plt.subplot2grid(shape=(4,3), loc=(1,0),colspan=2)
86 ax4 = plt.subplot2grid(shape=(4,3), loc=(1,2),colspan=1)
87 ax5 = plt.subplot2grid(shape=(4,3), loc=(2,0),colspan=2)
88 ax6 = plt.subplot2grid(shape=(4,3), loc=(2,2),colspan=1)
89 ax7 = plt.subplot2grid(shape=(4,3), loc=(3,0),colspan=2)
90 ax8 = plt.subplot2grid(shape=(4,3), loc=(3,2),colspan=1)
```

ภาพที่ 3.35 แสดง Code สร้างกราฟความสัมพันธ์ Timeseries

บรรทัดที่ 67-68	หมายถึง	เก็บ Timeseries ทุกตัวไว้ในลิสต์

บรรทัดที่ 69	หมายถึง	สร้างตัวแปรแทนค่า คือ A,B,C
บรรทัดที่ 70	หมายถึง	ให้ค่าที่คำนวณอยู่ในช่วง 0 ถึง ค่าที่อยู่ในลิสต์
บรรทัดที่ 71-73	หมายถึง	คำนวณค่า R^2 จากค่าที่ลิสต์ไว้ , คำนวณค่า EI RMSE จากค่าที่ลิสต์และใช้ loc. เพื่อดึงค่าจากแถว 0กับ 1
บรรทัดที่ 74-76	หมายถึง	แทนค่า R^2 , NSE , RMSE ลงในตัวแปร A,B,C ตามลำดับ
บรรทัดที่ 78-80	หมายถึง	นำเข้าแพคเกจ mathplotlib แบบ gridspec ซึ่งสามารถระบุพิกัดที่จะวางแผนผังย่อยได้และนำเข้าแพคเกจ numpyที่ไว้ใช้คำนวณทางคณิตศาสตร์
บรรทัดที่ 82	หมายถึง	กำหนดของของผังที่จะให้พล็อตออกมา
บรรทัดที่ 83-90	หมายถึง	ระบุพิกัดตำแหน่งระยะห่างระหว่างแผนผังแต่ละตัว

```
# Model 1
ax1.plot(TimeSeries_1['Day1'], color='deeppink', label='1
Day',marker = (5,2))
ax1.fill_between(TimeSeries_1.index,TimeSeries_1['Yobs'],
facecolor='Cyan',edgecolor='black',alpha=0.5,label='observed')
ax1.title.set_text('Observed and Forecasted')
ax1.set_xlabel('Days')
ax1.set_ylabel('Discharge ($m^3$/s)')
ax1.grid(True)
ax1.set_ylim(0,700)
ax1.legend(loc='upper right')
#ax1.text(5, 120, 'EI = ' + str(round(B[0],3)))
#ax1.text(10,120, 'RMSE = ' + str(round(C[0],3)) +'$m^3$/s')

ax2.scatter(TimeSeries_1['Yobs'], TimeSeries_1['Day1'],
color='deeppink', marker='o', edgecolors='k')
ax2.title.set_text('Scatter plot Observed and Forecasted')
ax2.set_xlabel('Observed discharge ($m^3$/s)')
ax2.set_ylabel('Forecasted Discharge ($m^3$/s)')
ax2.set_ylim(0,700)
ax2.grid(True)
m,b=np.polyfit(TimeSeries_1['Yobs'],TimeSeries_1['Day1'],1)
ax2.plot(TimeSeries_1['Yobs'], m*TimeSeries_1['Yobs']+b,'k')
equation = 'y ='+str(round(m,4))+'x'+'+' +str(round(b,4))
#ax2.text(310,80, str(equation), color = 'k')
ax2.text(360,70,'$R^2$='+ str(round(A[0],3)),color = 'k')
```

ภาพที่ 3.36 แสดง Code สร้างกราฟความสัมพันธ์ Timeseries

บรรทัดที่ 93-103	หมายถึง	แสดงกราฟย่อยใช้ marker แบบจุดต่อกันเป็นเส้นโดยแสดงทับระหว่างค่าจริงที่อยู่ (Yobs) พร้อมทั้งกำหนดขนาดชื่อแกนและขอบเขตแกน
บรรทัดที่ 107-118	หมายถึง	คำสั่งเพื่อสร้างกราฟ scatter จากค่า Yobs ตั้งชื่อแกน กำหนดขอบเขต ให้ตัวแปร m,b เป็นค่าที่รับจาก Yobsใช้ฟังก์ชัน polyfit() เพื่อรับค่าเส้นแนวโน้มใน mathplotlib เพื่อสร้างเส้นแนวโน้มในการกระจายตัวและให้ผลลัพธ์ของค่า y,x ที่รวมกันแปลงมาจากการปัดเลขของตัวแปร m,b

```
120
121 #Model 2
122 ax3.plot(TimeSeries_2['Day2'],color='forestgreen',label='2
123 Day',marker = (5,2))
124 ax3.fill_between(TimeSeries_2.index,TimeSeries_2['Yobs'],
125 facecolor='cyan',edgecolor='black',alpha=0.5,label='observed')
126 ax3.title.set_text('Observed and Forecasted')
127 ax3.set_xlabel('Days')
128 ax3.set_ylabel('Discharge ($m^3$/s)')
129 ax3.grid(True)
130 ax3.set_ylim(0,700)
131 ax3.legend(loc="upper right")
132 #ax3.text(5, 130,'EI ='+ str(round(B[1],3)))
133 #ax3.text(10,130,'RMSE ='+ str(round(C[1],3))+'$m^3$/s')
134
135 ax4.scatter(TimeSeries_2['Yobs'],TimeSeries_2['Day2'],color
136 ='forestgreen', marker="o", edgecolor='k')
137 ax4.grid(True)
138 ax4.title.set_text('Scatter plot Observed and Forecasted')
139 ax4.set_ylim(0,700)
140 ax4.set_xlabel("Observed discharge ($m^3$/s)")
141 ax4.set_ylabel("Forecasted  discharge ($m^3$/s)")
142 m,b=np.polyfit(TimeSeries_2['Yobs'],TimeSeries_2['Day2'],1)
143 ax4.plot(TimeSeries_2['Yobs'], m*TimeSeries_2['Yobs']+b,'k')
144 equation = 'y = ' + str(round(m,3)) + 'x' ' + ' +
145 str(round(b,3))
146 # ax4.text(310,80, str(equation),  color='black')
147 ax4.text(360,70, '$R^2$ = ' + str(round(A[1],3)),
148 color='black')
```

ภาพที่ 3.37 แสดง Code สร้างกราฟความสัมพันธ์ Timeseries

บรรทัดที่ 121-130	หมายถึง	แสดงกราฟย่อยใช้ marker แบบจุดต่อกันเป็นเส้นโดยแสดงทับระหว่างค่าจริงที่อยู่ (Yobs) พร้อมทั้งกำหนดขนาดชื่อแกนและขอบเขตแกน
บรรทัดที่ 134-148	หมายถึง	คำสั่งเพื่อสร้างกราฟ scatter จากค่า Yobs ตั้งชื่อแกน กำหนดขอบเขต ให้ตัวแปร m,b เป็นค่าที่รับจาก Yobsใช้ฟังก์ชัน polyfit() เพื่อรับค่าเส้นแนวโน้มใน mathplotlib เพื่อสร้างเส้นแนวโน้มในการกระจายตัวและให้ผลลัพธ์ของค่า y,x

```
# Model 3
ax5.plot(TimeSeries_3['Day3'],color='royalblue',label='3
Day',marker = (5,2))
ax5.fill_between(TimeSeries_3.index,TimeSeries_3['Yobs'],
facecolor='cyan',edgecolor='black',alpha=0.5
,label='observed')
ax5.title.set_text('Observed and Forecasted')
ax5.set_xlabel('Days')
ax5.set_ylabel('Discharge ($m^3$/s)')
ax5.grid(True)
ax5.set_ylim(0,700)
ax5.legend(loc="upper right")
#ax5.text(5,130,'EI ='+ str(round(B[2],3)))
#ax5.text(10,130,'RMSE ='+ str(round(C[2],3)) +'$m^3$/s')
ax6.scatter(TimeSeries_3['Yobs'],TimeSeries_3['Day3']
,color= 'royalblue', marker="o",edgecolor='k')
ax6.grid(True)
ax6.set_ylim(0,700)
ax6.title.set_text('Scatter plot Observed
and Forecasted')
ax6.set_xlabel("Observed discharge ($m^3$/s)")
ax6.set_ylabel("Forecasted  discharge ($m^3$/s)")
m,b = np.polyfit(TimeSeries_3['Yobs']
,TimeSeries_3['Day3'],1)
ax6.plot(TimeSeries_3['Yobs'], m*TimeSeries_3
['Yobs'] +b,'k')
equation = 'y ='+ str(round(m,3))+ 'x' ' + '
+ str(round(b,3))
#ax6.text(310,80,str(equation),  color='black')
ax6.text(360,70,'$R^2$ = '+ str(round(A[2],3)),
color='black')
```

ที่รวมกันแปลงมาจากการปัดเลขของตัวแปร m,b

<u>ภาพที่ 3.38</u> แสดง Code สร้างกราฟความสัมพันธ์ Timeseries

บรรทัดที่ 150-159	หมายถึง	แสดงกราฟย่อยใช้ marker แบบจุดต่อกันเป็นเส้นโดยแสดงทับระหว่างค่าจริงที่อยู่ (Yobs) พร้อมทั้งกำหนดขนาดชื่อแกนและขอบเขตแกน
บรรทัดที่ 163-176	หมายถึง	คำสั่งเพื่อสร้างกราฟ scatter จากค่า Yobs ตั้งชื่อแกนกำหนดขอบเขต ให้ตัวแปร m,b เป็นค่าที่รับจาก Yobsใช้ฟังก์ชัน polyfit() เพื่อรับค่าเส้นแนวโน้มใน mathplotlib เพื่อสร้างเส้นแนวโน้มในการกระจายตัวและให้ผลลัพธ์ของค่าที่รวมกันแปลงมาจากการปัดเลขของตัวแปร m,b

```
# Model 4
ax7.plot(TimeSeries_4['Day4'],color='brown',label='4
Day',marker = (5,2))
ax7.fill_between(TimeSeries_4.index,TimeSeries_4
['Yobs'],facecolor='cyan',
edgecolor='black',alpha=0.5,label='observed')
ax7.title.set_text('Observed and Forecasted')
ax7.set_xlabel('Days')
ax7.set_ylabel('Discharge ($m^3$/s)')
ax7.grid(True)
ax7.set_ylim(0,700)
ax7.legend(loc="upper right")

# ax7.text(5, 130, 'EI = ' + str(round(B[3],3)))
# ax7.text(10,130, 'RMSE = ' + str(round(C[3],3))
+'$m^3$/s')

ax8.scatter(TimeSeries_4['Yobs'], TimeSeries_4['Day4']
,color = 'brown', marker="o", edgecolor='k')
ax8.grid(True)
ax8.set_ylim(0,700)
ax8.title.set_text('Scatter plot Observed and
Forecasted')
ax8.set_xlabel("Observed discharge ($m^3$/s)")
ax8.set_ylabel("Forecasted  discharge ($m^3$/s)")
m,b = np.polyfit(TimeSeries_4['Yobs'],
TimeSeries_4['Day4'],1)
ax8.plot(TimeSeries_4['Yobs'], m*TimeSeries_4['Yobs']+
b,'k')
equation = 'y ='+ str(round(m,3)) + 'x' ' + ' +
str(round(b,3))
# ax8.text(310,80, str(equation),  color='black')
ax8.text(360,70, '$R^2$ = ' + str(round(A[3],3)),
color='black')
plt.tight_layout()
# plt.savefig('D:/Research_IRID/Plot_complete.png'
,dpi=800)
# plt.savefig('Timeseries.eps', format='eps')
```

ภาพที่ 3.39 แสดง Code สร้างกราฟความสัมพันธ์ Timeseries

บรรทัดที่ 180-191	หมายถึง	แสดงกราฟย่อยใช้ marker แบบจุดต่อกันเป็นเส้นโดยแสดงทับระหว่างค่าจริงที่อยู่ (Yobs) พร้อมทั้งกำหนดขนาด ชื่อแกน และขอบเขตแกน

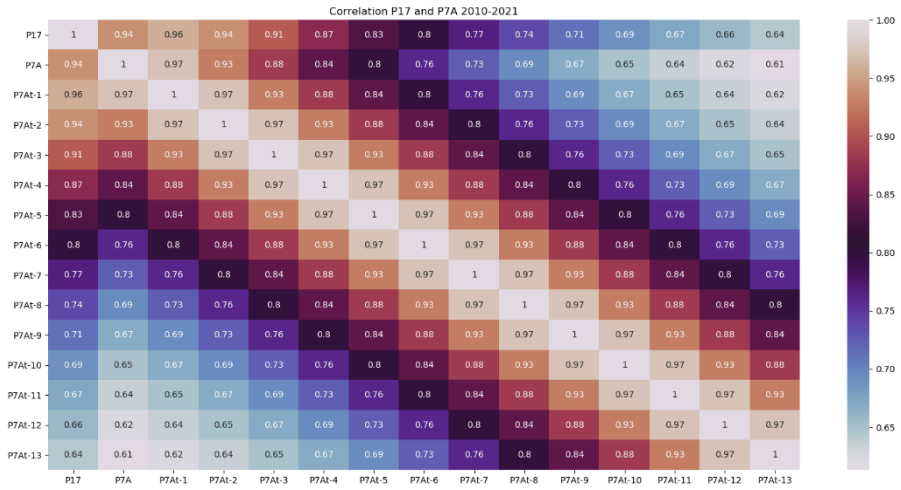
บรรทัดที่ 197-214	หมายถึง	คำสั่งเพื่อสร้างกราฟ scatter จากค่า Yobs ตั้งชื่อแกนกำหนดขอบเขต ให้ตัวแปร m,b เป็นค่าที่รับจาก Yobsใช้ฟังก์ชัน polyfit() เพื่อรับค่าเส้นแนวโน้มใน mathplotlib เพื่อสร้างเส้นแนวโน้มในการกระจายตัวและให้ผลลัพธ์ของค่า y,x ที่รวมกันแปลงมาจากการปัดเลขของตัวแปร m,b

บทที่ 4

ผลและการวิจารณ์

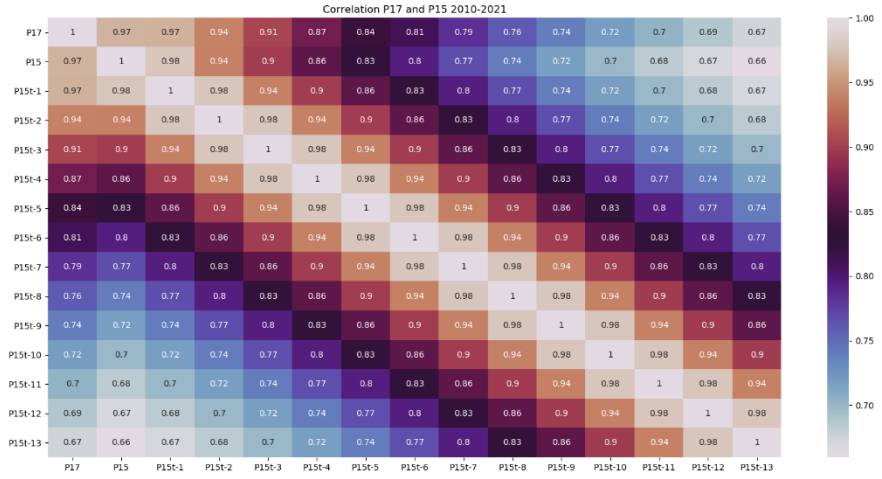
การพยากรณ์การไหลของปริมารณน้ำท่าด้วยโครงข่ายประสาทเทียมโดย โปรแกรมPython กรณีศึกษาสถานีวัดน้ำท่า P.17 ในลุ่มน้ำปิงตอนล่าง ประเทศไทย

4.1 ผลการหาค่าสัมประสิทธิ์สหสัมพันธ์



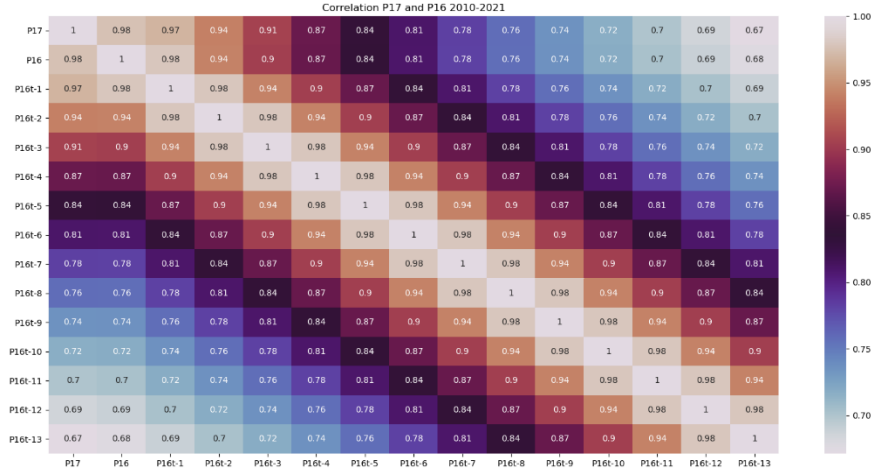
<u>ภาพที่ 4.1</u> แผนภาพ Heatmap แสดงค่าสัมประสิทธิ์สหสัมพันธ์ระหว่างสถานี P.17 กับ P.7A ในปี 2010-2021

จากการหาค่าสัมประสิทธิ์สหสัมพันธ์ (Correlation) ระหว่าง สถานี P.17 กับ P.7A ในปี 2010-2021 ซึ่งแสดงผลโดยใช้ Module Seaborn ใน Python แสดงผลเป็น Heatmap เห็นได้ว่าจะใช้ข้อมูลย้อนหลัง 1 วันเพื่อใช้ในการพยากรณ์ซึ่งสามารถนำข้อมูลย้อนหลัง 1 วันนี้ไปเข้าขั้นตอน MLP Regressor ในกระบวนการฝึกสอนและทวนสอบแบบจำลองเพื่อหา Model ที่ดีที่สุดไปวิเคราะห์ต่อไป



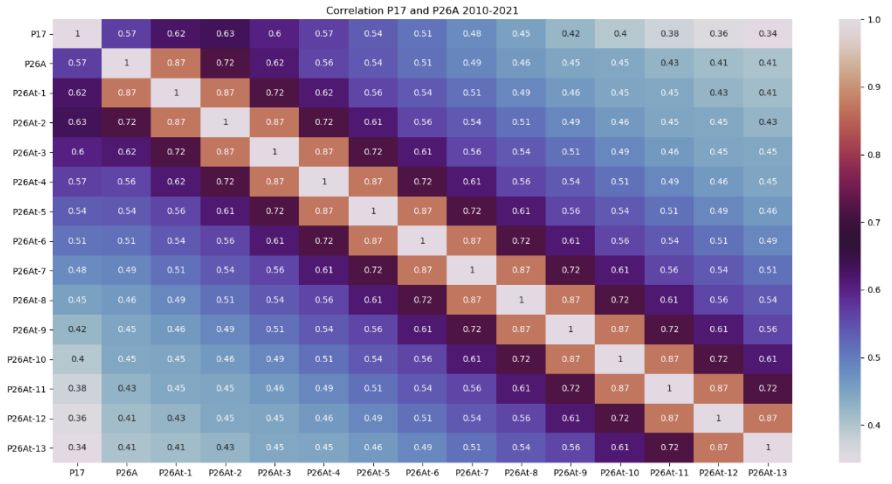
ภาพที่ 4.2 แผนภาพ Heatmap แสดงค่าสัมประสิทธิ์สหสัมพันธ์ระหว่างสถานี P.17 กับ P.15 ในปี 2010-2021

จากการหาค่าสัมประสิทธิ์สหสัมพันธ์ (Correlation) ระหว่าง สถานี P.17 กับ P.15 ในปี 2010-2021 ซึ่งแสดงผลโดยใช้ Module Seaborn ใน Python แสดงผลเป็น Heatmap เห็นได้ว่าจะใช้ข้อมูลย้อนหลัง 1 วันเพื่อใช้ในการพยากรณ์ซึ่งสามารถนำข้อมูลย้อนหลัง 1 วันนี้ไปเข้าขั้นตอน MLP Regressor ในกระบวนการฝึกสอนและทวนสอบแบบจำลองเพื่อหา Model ที่ดีที่สุดไปวิเคราะห์ต่อไป



ภาพที่ 4.3 แผนภาพ Heatmap แสดงค่าสัมประสิทธิ์สหสัมพันธ์ระหว่างสถานี P.17 กับ P.16 ในปี 2010-2021

จากการหาค่าสัมประสิทธิ์สหสัมพันธ์ (Correlation) ระหว่าง สถานี P.17 กับ P.16 ในปี 2010-2021 ซึ่งแสดงผลโดยใช้ Module Seaborn ใน Python แสดงผลเป็น Heatmap เห็นได้ว่าจะใช้ข้อมูลย้อนหลัง 1 วันเพื่อใช้ในการพยากรณ์ซึ่งสามารถนำข้อมูลย้อนหลัง 1 วันนี้ไปเข้าขั้นตอน MLP Regressor ในกระบวนการฝึกสอนและทวนสอบแบบจำลองเพื่อหา Model ที่ดีที่สุดไปวิเคราะห์ต่อไป



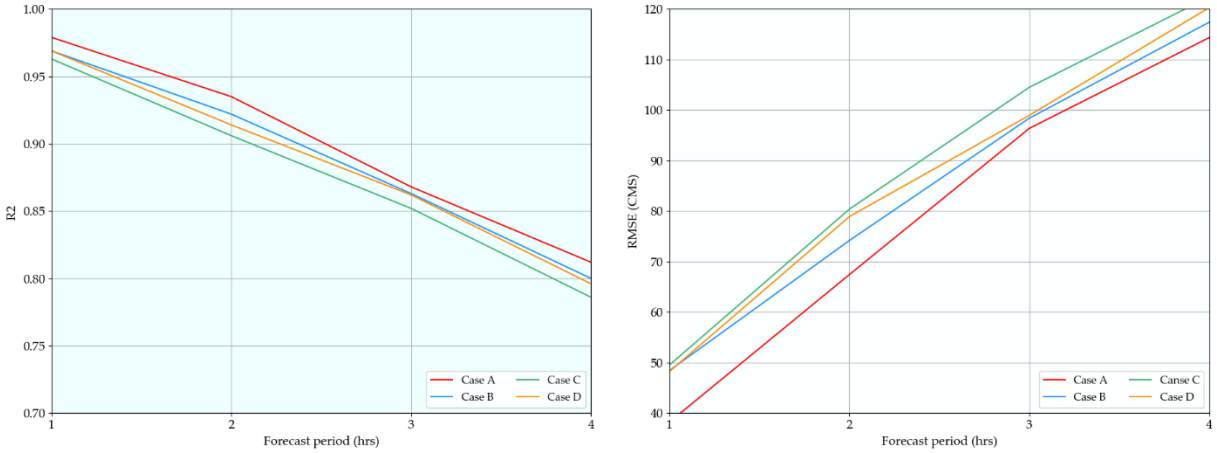
<u>ภาพที่ 4.4</u> แผนภาพ Heatmap แสดงค่าสัมประสิทธิ์สหสัมพันธ์ระหว่างสถานี P.17 กับ P.26A ในปี 2010-2021

จากการหาค่าสัมประสิทธิ์สหสัมพันธ์ (Correlation) ระหว่าง สถานี P.17 กับ P.26A ในปี 2010-2021 ซึ่งแสดงผลโดยใช้ Module Seaborn ใน Python แสดงผลเป็น Heatmap เห็นได้ว่าจะใช้ข้อมูลย้อนหลัง 1 วันเพื่อใช้ในการพยากรณ์ซึ่งสามารถนำข้อมูลย้อนหลัง 1 วันนี้ไปเข้าขั้นตอน MLP Regressor ในกระบวนการฝึกสอนและทวนสอบแบบจำลองเพื่อหา Model ที่ดีที่สุดไปวิเคราะห์ต่อไป

<u>ภาพที่ 4.5</u> แผนภาพ Heatmap แสดงค่าสัมประสิทธิ์สหสัมพันธ์ระหว่างสถานี P.17 กับ RF ในปี 2010-2021

จากการหาค่าสัมประสิทธิ์สหสัมพันธ์ (Correlation) ระหว่าง สถานี P.17 กับ RF ในปี 2010-2021 ซึ่งแสดงผลโดยใช้ Module Seaborn ใน Python แสดงผลเป็น Heatmap เห็นได้ว่าจะใช้ข้อมูลย้อนหลัง 1 วันเพื่อใช้ในการพยากรณ์ซึ่งสามารถนำข้อมูลย้อนหลัง 1 วันนี้ไปเข้าขั้นตอน MLP Regressor ในกระบวนการฝึกสอนและทวนสอบแบบจำลองเพื่อหา Model ที่ดีที่สุดไปวิเคราะห์ต่อไป

4.2 ผลของการ Validation ข้อมูล



ภาพที่ 4.6 แสดงผลการ Validation ข้อมูลของทั้ง 4 ชุดข้อมูล

จากกราฟคือผลจากการรันโค้ดเพื่อดูว่าแบบจำลองไหนมีผลดีที่สุดที่จะสามารถนำไปใช้พยากรณ์ปริมาณน้ำต่อไป ซึ่งผลที่ได้คือแบบจำลองของ Case A ให้ค่าความถูกต้องของค่า R^2 มีค่าเข้าใกล้ 1 และค่า RMSE มีค่าเข้าใกล้ 0 มากที่สุดเมื่อเทียบกับกรณีอื่นจึงเลือกฟังก์ชันของกรณี A มาทำการทดสอบ ANNs เพื่อคาดการณ์ปริมาณน้ำท่าต่อไป

เมื่อโมเดลของแต่ละ Case คือ

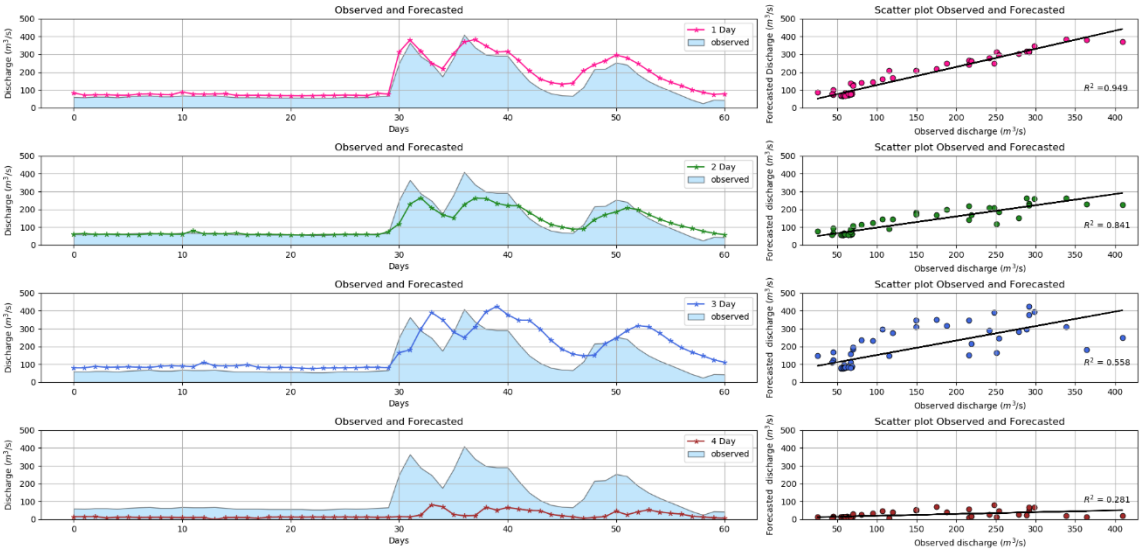
Case A = f ($P.17_{t-1}$, $P.15_{t-1}$, $P.16_{t-1}$, $P.7A_{t-1}$, $P.26A_{t-1}$, RF_{t-2},)

Case B = f ($P.17_{t-1}$, $P.15_{t-1}$, $P.16_{t-1}$, RF_{t-2},)

Case C = f ($P.17_{t-1}$, $P.15_{t-1}$, $P.16_{t-1}$)

Case D = f ($P.17_{t-1}$, $P.15_{t-1}$, $P.7A_{t-1}$)

4.3 ผลการทดสอบแบบจำลองโดยการ Run เป็น Timeseries



<u>ภาพที่ 4.7</u> กราฟแสดง Timeseries Plot Observed and Forecasted and Scatter Plot Observed and Forecasted

จากรูปที่ 4.7 เป็นการพล็อตระหว่างค่าตรวจวัดจริง (Observed) กับค่าการพยากรณ์ (Forecasted) ในรูปแบบสุ่มตัวอย่าง (Samples) ไม่ได้เรียงตาม Datetime Index และการสร้างสมการเส้นตรงเพื่อนำไป Plot Trend line และแสดงค่า R^2 ในกราฟ Plot Scatter ระหว่างค่าตรวจวัดจริง (Observed) กับค่าการพยากรณ์ (Forecasted) ในกราฟนี้เป็นการแสดงข้อมูลของช่วงที่มีปริมาณน้ำทั้งในฤดูแล้งและฤดูฝน

ตารางที่ 4.1 แสดงค่า ค่า R^2 , RMSE , MAPE และ NSE ในการพยากรณ์ปริมาณน้ำท่า

จำนวนวันในการพยากรณ์	R^2	RMSE	MAPE	NSE
1	0.949	36.321	33.559	0.872
2	0.841	46.392	20.984	0.776
3	0.558	86.918	70.271	0.357
4	0.281	132.235	79.306	0.079

จากกราฟ Timeseries จะสังเกตได้ว่าข้อมูลจาก Model ที่เลือกนั้นให้ผลดัชนีทางคณิตศาสตร์ออกมาดังตารางที่ 4.1 ซึ่งผลจากตารางสามารถสรุปได้ว่า การคาดการณ์ปริมาณน้ำท่าที่จะเกิดขึ้นของสถานีน้ำท่า P.17 โดยใช้ข้อมูลของสถานีน้ำท่าอื่นๆที่เกี่ยวข้องสามารถคาดการณ์ปริมาณล่วงหน้าแม่นยำและทันท่วงทีใน 2 วัน

บทที่5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการศึกษา

จากการศึกษาพยากรณ์ปริมาณน้ำท่าของสถานีน้ำท่า P.17 เมื่อนำข้อมูลของสถานีน้ำท่าและสถานีน้ำฝนในพื้นที่ใกล้เคียงมาพยากรณ์ได้แก่สถานีน้ำท่า P.15 , P.16 , P.7A , P.26A และสถานีน้ำฝน 120161 มาทำการหาค่าสัมประสิทธิ์สหสัมพันธ์ (Correlation) ที่ดีที่สุดของสถานีน้ำท่า P.17 คือชุดสถานีน้ำท่า P.17 กับ P.7A ซึ่งค่า Lag Time อยู่ที่ 1 วัน และของสถานีน้ำฝน 120161 Lag Time อยู่ที่ 1 วัน ซึ่งเมื่อนำข้อมูลของแต่ละชุดสถานีไปทดสอบค่าก่อนนำไปใช้พยากรณ์พบว่าทุกค่าของทุกสถานีที่เกี่ยวข้องนั้นล้วนส่งผลต่อการพยากรณ์จึงนำทุกสถานีไปทำการทดสอบ เมื่อได้ชุดข้อมูลที่ต้องการจึงนำไปทดสอบโดยใช้วิธีการวิเคราะห์โครงข่ายประสาทเทียม Artificial neural networks (ANNs) มาช่วยในการพยากรณ์ปริมาณน้ำที่สถานีน้ำท่า P.17 ซึ่งผลลัพธ์ที่ได้สามารถคาดการปริมาณน้ำล่วงหน้าได้ถึง 1-2 วัน ได้ค่า R^2 , RMSE และ NSE ของแบบจำลองดังนี้

ในการพยากรณ์น้ำล่วงหน้า 1 วัน แบบจำลองมีค่า R^2 เท่ากับ 0.949 ค่า RMSE เท่ากับ 36.321 m^3/s และค่า NSE เท่ากับ 0.872

ในการพยากรณ์น้ำล่วงหน้า 2 วัน แบบจำลองมีค่า R^2 เท่ากับ 0.841 ค่า RMSE เท่ากับ 46.392 m^3/s และค่า NSE เท่ากับ 0.776

ดังนั้นจึงสรุปได้ว่า แบบจำลอง Artificial Neural Networks (ANNs) สามารถพยากรณ์น้ำล่วงหน้าที่ 1 วัน ได้มีความแม่นยำมากที่สุด รองลงมาคือ 2 วัน ซึ่งตรงกับสมมติฐานที่คาดการณ์ไว้ว่าสามรถพยากรณ์ได้ล่วงหน้า 1-3 วัน

5.2 ปัญหาและอุปสรรค

การใช้โปรแกรม Spyder ในการเขียนภาษา Python ด้วยความที่เป็นภาษาคอมพิวเตอร์จึงค่อนข้างที่จะซับซ้อนเล็กน้อย การเขียนผิดหลักภาษา หรือใช้คำสั่งผิด Library จะทำให้เกิดการชะงักของโปรแกรม และทำให้รันชุดคำสั่งไม่ได้ การติดตั้งคำสั่งบางอย่างไม่สามารถติดตั้งได้จึงต้องลองใช้ชุดคำสั่งอื่นในการทำงานซึ่งอาจทำให้ค่าคลาดเคลื่อนขึ้นได้ และเนื่องจากข้อมูลมีจำนวนมากจึงใช้เวลาในการคัดเลือกค่อนข้างนานทำให้การวิจัยล่าช้ายิ่งขึ้น

อีกหนึ่งอุปสรรคคือข้อมูลที่ได้มาจากส่วนกลางซึ่งบางช่วงข้อมูลของแต่ละสถานีนั้นไม่ได้จัดเก็บหรือบันทึกนั้นทำให้เวลานำข้อมูลเหล่านั้นมาทำการพยากรณ์ ความแม่นยำในการคาดการณ์จะลดลงหรือคลาดเคลื่อนได้

5.3 ข้อเสนอแนะ

5.3.1 เพื่อให้ผลการพยากรณ์มีความแม่นยำควรทำการบันทึกข้อมูลใหม่ทุกๆปี

5.3.2 การพยากรณ์ สิ่งที่ควรคำนึงคือข้อมูลที่ใช้ควรเป็นข้อมูลที่อิงมาจากธรรมชาติของลุ่มน้ำหรือสภาพภูมิประเทศ เพื่อการพยากรณ์ที่แม่นยำใกล้เคียงความจริง

5.3.3 อาจใช้แบบจำลองอื่นหรือโปรแกรมอื่นนอกเหนือจาก Python ในการทำพยากรณ์เผื่ออาจจะได้โมเดลที่ประสิทธิภาพที่มากขึ้น

5.3.4 เมื่อได้แบบจำลองพยากรณ์น้ำท่าที่มีความแม่นยำ ควรลองนำไปประยุกต์ใช้กับสถานีวัดน้ำท่าอื่นในประเทศไทย เพื่อที่จะพยากรณ์ได้มากขึ้นและแม่นยำมากขึ้น

5.3.4 ควรพัฒนาแบบจำลองให้มีการพยากรณ์ได้เป็นรายชั่วโมง จะได้มีความละเอียดและแม่นยำมากขึ้น

5.3.5 ในส่วนโมเดลต่อไปถ้าสามารถพัฒนาจัดทำ Application ขึ้นมาได้จะสะดวกต่อการพยากรณ์ในอนาคตยิ่งขึ้น

5.3.6 ควรแยกโมเดลให้ชัดเจนของช่วงหน้าฝนและช่วงหน้าแล้ง เพราะถ้ารวมกันโมเดลจะทำการหาค่าเฉลี่ยตรงกลางจากค่าต่ำสูงที่คำนวณ โมเดลจึงจำค่าเหล่านั้นมาพยากรณ์ทำให้ค่าคลาดเคลื่อน

5.3.7 เนื่องจากอินพุตมีหลายจำนวนควรเลือกสถานีน้ำท่าและสถานีน้ำฝนใหม่ 2-3 สถานีในการวิเคราะห์เพื่อได้ค่าที่แม่นยำ

5.3.8 ปริมาณฝนอาจไม่ใช่ Lag Time ที่ดีที่สุดเนื่องจากวันที่ฝนตกหนักสุดกลับมีปริมาณน้ำไม่มากที่สุดทั้งนี้อาจเป็นเพราะฝนที่ตกไม่ได้อยู่ในที่เก็บกักหรืออาจซึมผ่านลงไปในดินแต่ผลการของ Validation มีปริมาณฝนเข้ามาเกี่ยวข้องกับการพยากรณ์ดังนั้นในขั้นตอนวิเคราะห์จึงต้องนำฝนมาคำนวณซึ่งค่าที่ได้อาจทำให้ผลพยากรณ์คาดเคลื่อน

5.3.9 นำแบบจำลองโครงสร้างที่ดีที่สุดและตัวแปรที่ใช้ไปรายงานสำนักบริหารการจัดการน้ำและอุทกวิทยาเพื่อเป็นแนวทางในการพยากรณ์และเพื่อเตือนภัยต่อไป

เอกสารและสิ่งอ้างอิง

ทวี ชัยพิมลผลิน ,ทวีศักดิ์ วงไพศาล ,. **แบบจำลองโครงข่ายประสาทเทียมสำหรับการพยากรณ์น้ำท่วมในลุ่มน้ำมูลตอนล่าง .** Journal of Science & Technology MSU, 35(5) . (2558)

ทศพล ภูผิวฟ้า . **แบบจำลองพารามิเตอร์บ่งรูปร่างของการแจกแจงค่าสุดขีดนัยทั่วไปจากข้อมูลน้ำฝน-น้ำท่า และข้อมูลภาพถ่ายดาวเทียมโดยใช้วิธีโครงข่ายประสาทเทียม** . สาขาวิชาวิทยาการจัดการสถิติ มกราคม 2565 มหาวิทยาลัยมหาสารคาม . (2565)

ธิษณ์ปัณฑา คนโทฉิมพลี . **การประยุกต์ใช้โครงข่ายใยประสาทเทียมในการพยากรณ์ปริมาณน้ำไหลเข้าอ่างเก็บน้ำที่สำคัญในจังหวัดระยอง** . ปริญญาโท ภาคพิเศษ สาขาวิศวกรรมทรัพยากรน้ำ มหาวิทยาลัยเกษตรศาสตร์ . (2560)

พรรณปพร บุญแปง และคณะ **. ความเป็นไปได้ สำหรับการคาดการณ์แผ่นดินไหวในประเทศไทยด้วยแบบจำลองโครงข่ายประสาทเทียม .** วารสาร วิทยาศาสตร์ และ เทคโนโลยี มหาวิทยาลัย มหาสารคาม , 39(4), 400-413 . (2563)

ยุโชติ อมรศักดิ์ , ยุพิน ไชยสมภาร , ทวีชัย พิมลผลิน . **การประยุกต์ใช้แบบจำลองโครงข่ายประสาทเทียมเพื่อคาดการณ์น้ำท่วมในอนาคต: กรณีศึกษาเทศบาลนครเชียงใหม่ .** วารสาร สังคมศาสตร์ มหาวิทยาลัย ศรีนครินทรวิโรฒ, 20(1) . (2560)

รติพร จันทร์กลั่น . **การสร้างแบบจำลองด้วยเทคนิคการเรียนรู้ของเครื่องเพื่อคาดการณ์ปริมาณน้ำท่า (Doctoral dissertation)** . สาขาวิชาวิศวกรรมคอมพิวเตอร์สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัย เทคโนโลยี สุรนารี . (2560)

วิรัช กองสิน . **การศึกษาระบบควบคุมปริมาณน้ำให้กับพืชระยะสั้นด้วยโครงข่ายประสาทเทียม .** RMUTSB ACADEMIC JOURNAL, 7(1), 40-51. (2562)

Alireza Sharifi., Rasoul Mirabbasi. , Yagob Dinpashoh... **Daily runoff prediction using the linear and non-linear models.** Water Science and Technology, 76(4), 793-805. (2017)

Imen Aichouri. . **River flow model using artificial neural networks.** Energy Procedia , *74*, 1007-1014 . (2015)

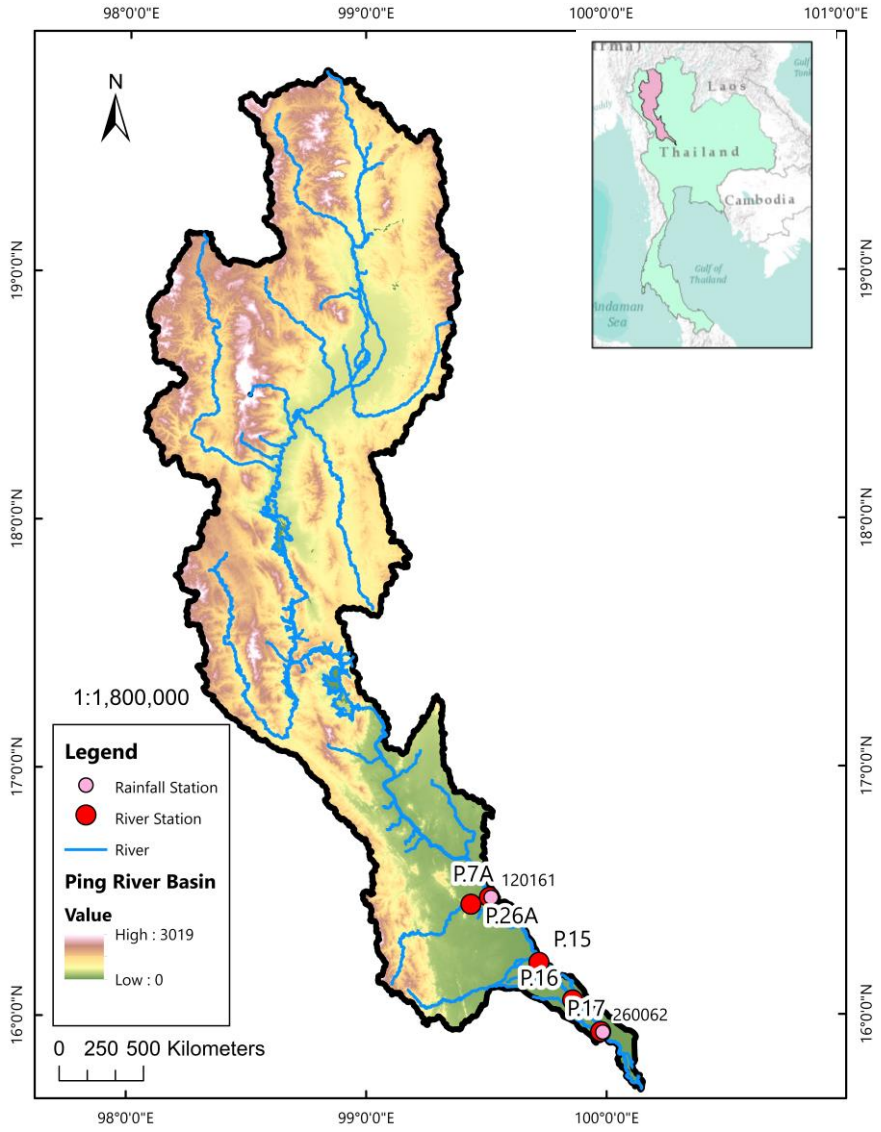
Gokcen Uysal. . **Streamflow forecasting using different neural network models with satellite data for a snow dominated region in Turkey .** Procedia Engineering, 154, 1185-1192 . (2016)

Imen Aichouri. . **River flow model using artificial neural networks.** Energy Procedia , *74*, 1007-1014 . (2015)

ภาคผนวก

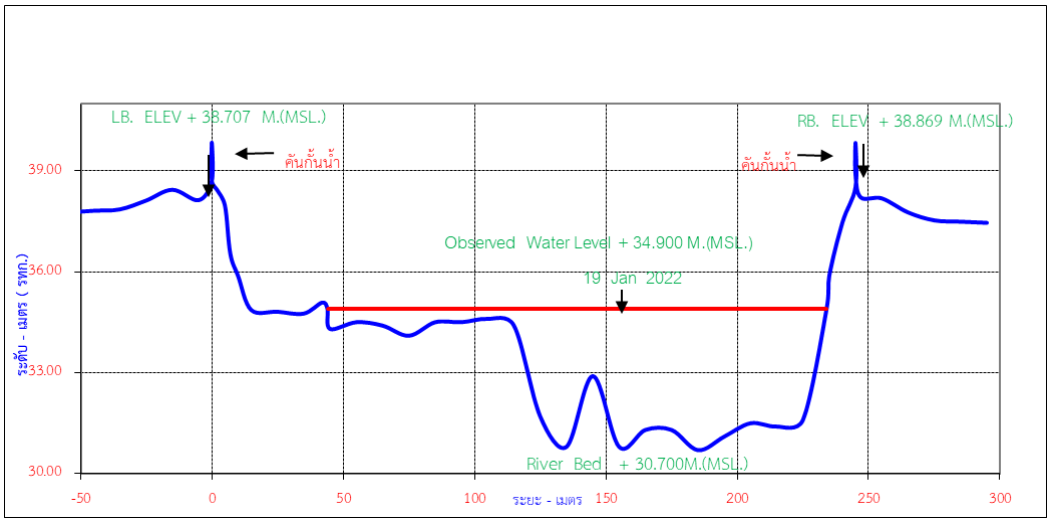
ภาคผนวก ก

ข้อมูลเพิ่มเติม สถานีวัดน้ำท่า P.17



ภาค นวก ก1 แสดงตำแหน่งที่ตั้งของสถานีวัดน้ำท่าและสถานีวัดน้ำฝนที่เกี่ยวข้อง

<u>ภาค นวก ก2</u> ภาพถ่ายแนวสำรวจปริมาณน้ำสถานี P.17 แม่น้ำปิง อ.บรรพตพิสัย จ.นครสวรรค์



<u>ภาค นวก ก3</u> รูปตัดขวางลำน้ำสถานี P.17 แม่น้ำปิง บ้านท่างิ้ว ต.ท่างิ้ว อ.บรรพตพิสัย จ.นครสวรรค์

อธิบายโค้ดโปรแกรม Data1y1Col (เรียงข้อมูลอัตราการไหลรายวันหลายปีในไฟล์เดียวให้เป็นไฟล์ .csv รวมทุกปีแยกเป็นปีละ 1 คอลัมน์ในไฟล์เดียว

หน้าต่างตัวอย่างไฟล์ที่ได้

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Datetime	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
2	1-Apr	85.6	72.05	17.8	2.11	17.8	6.7	11.35	34.5	31.5	25.1	18.6	5.56
3	2-Apr	70.2	72.6	21.4	2	14	7	15.8	34	29.4	32.35	12.8	4.76
4	3-Apr	59.1	74.25	27.5	2	12.05	5.65	16.6	34.25	26.2	31.7	15.1	3.52
5	4-Apr	86	83.05	27.75	1.9	11.3	4.8	20	29.5	25.4	23.9	13.9	2.96
6	5-Apr	108	74.25	24.6	2	10.55	4.8	17.6	25	24.8	17.75	17.3	2.12
7	6-Apr	99.2	69.85	21.2	2.11	6.95	4.7	21.5	24.6	34.8	13	25.25	9.8
8	7-Apr	89.6	67.1	19	6.05	4.7	4.4	19.4	23.2	26.4	9.8	20	23.2
9	8-Apr	101.6	66	18.2	6.44	3.9	4.2	14.8	17.8	18.1	12.2	17.2	10.8
10	9-Apr	125.5	63	19	6.31	3.4	4.2	11.95	17.28	17.2	36.2	18.8	1.7
11	10-Apr	171.9	52.85	24.2	6.05	3.3	4.3	18.8	22.2	20.2	23.6	20.3	3.52
12	11-Apr	177.1	75.35	24	5.66	3.3	4.4	17.6	23.6	23.4	31.1	16.2	7.6
13	12-Apr	152.4	153.4	24	5.27	3.3	4.8	14.8	43.48	33	35.85	15	11
14	13-Apr	126.5	296	22.6	4.75	3.3	5	10.15	46.8	48.3	24.5	19.8	12.75
15	14-Apr	98.4	324.5	20.2	4.36	3.3	5.1	9.85	58.03	36.6	18.75	31.25	15.75
16	15-Apr	72.3	264	18	4.1	3.3	5.35	9.7	55.75	30.3	14.4	29.9	16
17	16-Apr	61.2	186.25	16.4	3.89	3.3	6.25	9.4	30.75	28.6	21.8	27.05	15.5
18	17-Apr	51.3	127.6	15.2	4.75	3.3	5.95	8.8	17.1	26.6	20	21.2	11.75
19	18-Apr	42	115.25	14	4.88	3.2	6.1	8.95	22	25.4	12	19.5	12.25
20	19-Apr	36.2	117.85	13.55	4.88	3.5	5.65	11.05	21.65	37.2	25.7	19	14.25
21	20-Apr	31	109	13.4	4.88	4.5	5.5	10.15	39.25	27.6	37.95	15	20.2
22	21-Apr	29.6	175.75	13.25	4.88	4.6	6.1	11.2	42.38	27	36.9	9.5	16
23	22-Apr	28.2	182.5	12.95	4.88	4.4	6.4	10.15	40.75	27.2	12.8	17.2	11.25
24	23-Apr	26.4	182.5	12.2	4.88	5	5.95	11.35	39	26.6	5.05	16.9	6.68
25	24-Apr	24.2	191.5	12.65	5.01	4.7	5.65	11.95	51.6	26	8.6	12.7	5.56
26	25-Apr	21.6	172.75	14	5.14	4.5	8.96	10.9	48	26	14.6	9.5	6.84
27	26-Apr	19.6	128.25	14.6	5.27	4.4	7.45	11.05	42.38	32.7	7.2	13.8	7.6
28	27-Apr	17.4	103	15.6	6.31	3.8	6.4	13.2	50.7	27.4	14	16.3	8.4
29	28-Apr	16.2	92.2	17	8.65	3.1	7.15	15.2	42.38	28.2	40.2	15	7.8
30	29-Apr	15.45	82.5	18.8	10.02	2.75	7.3	11.35	38.5	28	38.65	11.4	12.5
31	30-Apr	15.15	69.85	21.4	10.02	2.35	8.77	10.45	36.75	28	39.8	10.3	11.75

ส่วนที่ 1: การนำเข้าเครื่องมือ (Libraries)

```python
import pandas as pd
import re
import calendar
```

- import pandas as pd: นำเข้าไลบรารี Pandas (ตั้งชื่อย่อว่า pd) ซึ่งเป็นเครื่องมือหลักที่ใช้จัดการข้อมูลในรูปแบบตาราง (DataFrame)
- import re: นำเข้าไลบรารี Regular Expression ใช้สำหรับค้นหาและจัดการข้อความที่มีรูปแบบเฉพาะ (ในที่นี้ใช้หาคำว่า "Water Year")
- import calendar: นำเข้าเครื่องมือจัดการปฏิทิน ใช้ตรวจสอบว่าเดือนนั้นๆ มีกี่วัน (ช่วยแก้ปัญหาวันที่ 29 ก.พ.)

ส่วนที่ 2: ประกาศฟังก์ชันและการอ่านไฟล์

```
def parse_and_pivot_hydrological_data(file_path):
    # 1. อ่านและแปลงข้อมูลดิบ (Parse Data)
    with open(file_path, 'r', encoding='utf-8') as f:
        lines = f.readlines()
```

- def parse_...: ประกาศฟังก์ชันชื่อ parse_and_pivot_hydrological_data โดยรับค่า file_path (ชื่อไฟล์) เข้ามา
- with open(...) as f:: เปิดไฟล์ข้อความ (Text file) แบบอ่านอย่างเดียว ('r') และรองรับภาษาไทย/สัญลักษณ์ (encoding='utf-8')
- lines = f.readlines(): อ่านข้อมูลทั้งหมดในไฟล์ แล้วเก็บแยกไว้ทีละบรรทัดในรูปแบบ List ชื่อ lines

ส่วนที่ 3: การตั้งค่าตัวแปรเริ่มต้นและกฎของ "ปีน้ำ"

```
data = []
current_year = None

# ลำดับเดือนของปีน้ำ (เม.ย. เริ่มก่อน)
# (เดือน, ตัวทดปี) -> เม.ย.-ธ.ค. ใช้ปีเดิม, ม.ค.-มี.ค. ใช้ปีถัดไปในปฏิทิน (แต่เป็นปีน้ำเดียวกัน)
month_map = [
    (4, 0), (5, 0), (6, 0), (7, 0), (8, 0), (9, 0),
    (10, 0), (11, 0), (12, 0), (1, 1), (2, 1), (3, 1)
]
```

- data = []: สร้างลิสต์ว่างๆ เตรียมไว้สำหรับเก็บข้อมูลที่คัดแยกแล้ว
- current_year = None: ตัวแปรสำหรับจดจำว่าตอนนี้กำลังอ่านข้อมูลของ "ปีน้ำ" อะไรอยู่
- month_map = [...]: สร้างกฎการอ่านคอลัมน์เดือนตามรูปแบบของกรมชลประทาน โดยจับคู่เป็น (เดือน, ตัวทดปี)
    - (4, 0) หมายถึง เดือนที่ 4 (เม.ย.) อยู่ในปีปฏิทินเดียวกับปีน้ำ (บวก 0)
    - (1, 1) หมายถึง เดือนที่ 1 (ม.ค.) จะข้ามไปเป็นปีปฏิทินถัดไปแล้ว (บวก 1) แต่ยังนับเป็นปีน้ำเดิม
    -

ส่วนที่ 4: กระบวนการคัดกรองข้อมูลทีละบรรทัด

```
for line in lines:
    line = line.strip()
    # หาบรรทัดที่ระบุปีน้ำ เช่น "Water Year - 1999"
    m_year = re.search(r'Water Year\s*-\s*(\d{4})', line)
    if m_year:
        current_year = int(m_year.group(1))
        continue
```

- for line in lines:: นำข้อความที่อ่านมาวนลูปทำทีละบรรทัด
- line = line.strip(): ตัดช่องว่างหรือการขึ้นบรรทัดใหม่ที่อยู่หัว/ท้ายข้อความทิ้งไป

- m_year = re.search(...): ใช้ Regex หาคำว่า Water Year - ตามด้วยตัวเลข 4 หลัก (\d{4})
- if m_year:: ถ้าพบบรรทัดที่มีคำนี้
- current_year = int(m_year.group(1)): ให้ดึงเอาตัวเลขปีนั้น (เช่น 1999) มาแปลงเป็นตัวเลข (int) แล้วเก็บไว้ใน current_year
- continue: ข้ามไปอ่านบรรทัดถัดไปทันที (ไม่ต้องทำโค้ดด้านล่าง)

```
if current_year is None:
    continue

# ข้ามบรรทัดที่ไม่ใช่ข้อมูลวันที่
if not re.match(r'^\d+\s+', line):
    continue
```

- if current_year is None:: ถ้ายังไม่เจอข้อมูลปีน้ำเลย ให้ข้ามบรรทัดนั้นไปก่อน
- if not re.match(...): เช็คว่าบรรทัดนี้ขึ้นต้นด้วยตัวเลข (วันที่) และตามด้วยช่องว่างหรือไม่ ถ้าไม่ใช่ (เช่น เป็นหัวตาราง) ให้ข้ามไป

```
parts = line.split()
if not parts[0].isdigit():
    continue

day = int(parts[0])
values = parts[1:]
```

- parts = line.split(): หั่นข้อความในบรรทัดนั้นแยกออกจากกัน (ใช้ช่องว่างเป็นจุดตัด)
- if not parts[0].isdigit(): continue: ตรวจสอบซ้ำเพื่อความชัวร์ว่าคำแรกเป็นตัวเลข (วันที่) จริงๆ
- day = int(parts[0]): กำหนดให้คำแรก (index 0) เป็นตัวแปรวันที่ (day)
- values = parts[1:]: กำหนดให้คำที่เหลือตั้งแต่คำที่ 2 เป็นต้นไป เป็นชุดข้อมูลระดับน้ำ (values)
-

ส่วนที่ 5: การจัดการปัญหาวันที่ 29 ก.พ. และ 31 ในเดือนที่มี 30 วัน

```
# คำนวณเดือนที่มีจริงสำหรับวันนี้
valid_months = []
for month, year_offset in month_map:
    y = current_year + year_offset
    _, max_d = calendar.monthrange(y, month) # เช็ควันสุดท้ายของเดือน (รวม 29 ก.พ.)
    if day <= max_d:
        valid_months.append((y, month))
```

- valid_months = []: สร้างลิสต์เพื่อเก็บเฉพาะ "เดือนที่มีวันที่นี้อยู่จริง"
- for month...: วนลูป 12 เดือนตาม month_map
- y = current_year + year_offset: คำนวณปี ค.ศ. จริงๆ ของคอลัมน์นั้น (เช่น ปีน้ำ 1999 เดือนมกราคม คือ ค.ศ. 2000)

- _, max_d = calendar.monthrange(y, month): ให้โปรแกรมเช็คว่า เดือนนั้นของปีนั้น มีจำนวนวันสูงสุดกี่วัน (max_d)
- if day <= max_d:: เช็คว่าวันที่กำลังอ่านอยู่ มีค่าเกินวันสุดท้ายของเดือนไหม (เช่น วันที่ 31 ไปเช็คกับเดือน ก.ย. ซึ่งมีแค่ 30 วัน) ถ้าไม่เกิน แปลว่าเป็นเดือนที่ valid ให้เก็บลงลิสต์

```
# ตัดข้อมูลส่วนเกิน (เช่น คอลัมน์ Annual)
if len(values) > len(valid_months):
     values = values[:len(valid_months)]
```

- ในบางไฟล์จะมีคอลัมน์ Annual (สรุปรายปี) ห้อยท้ายมา โค้ดส่วนนี้จะตรวจสอบว่ามีข้อมูลเกินจำนวนเดือนที่มีจริงหรือไม่ ถ้าเกินจะตัดส่วนท้ายทิ้งไป

```
# เก็บข้อมูลลง list
for i, (y, m) in enumerate(valid_months):
    if i < len(values):
        try:
            val = float(values[i].replace(',', ''))
            data.append({
                'WaterYear': current_year,
                'Month': m,
                'Day': day,
                'Value': val
            })
        except ValueError:
            continue
```

- for i, (y, m)...: นำลิสต์เดือนที่ถูกต้องมาจับคู่กับค่าระดับน้ำ (ดึงมาตามลำดับ i)
- if i < len(values):: ป้องกัน Error กรณีข้อมูลในแถวนั้นมาไม่ครบ
- val = float(values[i].replace(',', '')): ดึงข้อมูลมาลบลูกน้ำ (,) ออก แล้วแปลงเป็นตัวเลขทศนิยม (float)
- data.append(...): เก็บข้อมูลที่แปลงสำเร็จเป็น Dictionary ใส่ลงในตัวแปร data (ประกอบด้วย ปีน้ำ, เดือน, วัน, และค่าที่วัดได้)
- except ValueError:: ถ้าข้อมูลแปลงเป็นตัวเลขไม่ได้ (เช่น เป็นเครื่องหมายขีด - หรือช่องว่าง) ให้ข้ามไปไม่ต้องสนใจ

ส่วนที่ 6: จัดรูปแบบตารางด้วย Pandas

```
df = pd.DataFrame(data)

# 2. สร้างรูปแบบการแสดงผลและการเรียงลำดับ
# สร้างลำดับเดือนเพื่อใช้ sort (เม.ย.=0 ... มี.ค.=11)
month_order = {4:0, 5:1, 6:2, 7:3, 8:4, 9:5, 10:6, 11:7, 12:8, 1:9, 2:10, 3:11}
df['MonthOrder'] = df['Month'].map(month_order)
```

- df = pd.DataFrame(data): แปลงลิสต์ที่ได้มาทั้งหมดให้กลายเป็นตาราง (DataFrame) ของ Pandas
- month_order = {...}: สร้างพจนานุกรมเพื่อกำหนดลำดับการเรียงเดือน (บังคับให้เดือน 4 หรือ เม.ย. มาเป็นอันดับ 0 เพื่อให้อยู่บนสุด)
- df['MonthOrder'] = ...: สร้างคอลัมน์ใหม่ชื่อ MonthOrder เพื่อใช้เป็นคีย์ในการจัดเรียงตาราง

```
# สร้างข้อความวันที่ (เช่น 01-Apr)
month_abbr = {i: calendar.month_abbr[i] for i in range(1, 13)}
df['DateStr'] = df.apply(lambda x: f"{int(x['Day']):02d}-{month_abbr[int(x['Month'])]}", axis=1)
```

- month_abbr = ...: สร้างพจนานุกรมชื่อเดือนย่อ (1=Jan, 2=Feb, ...)
- df['DateStr'] = ...: นำเลขวัน และ ชื่อเดือนย่อ มาต่อกัน (เช่น วันที่ 1 เดือน 4 จะกลายเป็น 01-Apr) และเก็บในคอลัมน์ใหม่ชื่อ DateStr

```
# 3. Pivot ตาราง (เปลี่ยนจากแถวเป็นคอลัมน์ตามปี)
pivot = df.pivot_table(index=['MonthOrder', 'Month', 'Day', 'DateStr'],
                       columns='WaterYear',
                       values='Value',
                       aggfunc='first')
```

- pivot = df.pivot_table(...): เป็นพระเอกของงานนี้ คือคำสั่งหมุนตาราง
    - index=...: กำหนดให้คอลัมน์ วัน/เดือน เป็นแถวแนวตั้ง (เรียงตาม MonthOrder)
    - columns='WaterYear': เอาปีน้ำไปกางเป็นคอลัมน์แนวนอน
    - values='Value': เอาค่าระดับน้ำไปใส่ตามช่องตัดระหว่างวันและปี
    - aggfunc='first': ถ้ามีข้อมูลซ้ำให้ดึงค่าแรกมาใช้

ส่วนที่ 7: การบังคับสร้างแกนเวลาให้ครบ 366 วัน (Reindexing)

```
# 4. สร้าง Index ให้ครบทุกวัน (รวม 29 ก.พ.)
# เพื่อให้ตารางมีแถว 29-Feb เสมอ แม้ปีนั้นจะไม่มีข้อมูล
full_index = []
months_in_order = [4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3]

for m in months_in_order:
    # กำหนดจำนวนวันสูงสุด (บังคับให้ ก.พ. มี 29 วัน เพื่อสร้างแถวรอไว้)
    if m == 2:
        max_d = 29
    elif m in [4, 6, 9, 11]:
        max_d = 30
    else:
        max_d = 31

    m_order = month_order[m]

    for d in range(1, max_d + 1):
        date_s = f"{d:02d}-{month_abbr[m]}"
        full_index.append((m_order, m, d, date_s))

# Reindex เพื่อเติมแถวที่ขาด
pivot = pivot.reindex(full_index)
```

- เป้าหมายของส่วนนี้คือ ถึงแม้ข้อมูลที่เราอ่านมาจะไม่มีวันอธิกสุรทินเลย เราก็อยากให้โครงสร้างตารางมีช่องวันที่ 29-Feb รอเอาไว้
- full_index = []: เตรียมลิสต์ว่างสำหรับโครงสร้างเวลาที่สมบูรณ์
- for m ...: วนลูปเดือนตามลำดับปีน้ำ
- if m == 2: max_d = 29 ...: **บังคับ** ให้เดือนกุมภาพันธ์ (2) มี 29 วันเสมอ
- for d in range(1, max_d + 1):: นำมาสร้างเป็นโครงสร้าง (ลำดับเดือน, เลขเดือน, เลขวัน, ข้อความวันที่) เก็บลงใน full_index
- pivot = pivot.reindex(full_index): สั่งให้ตารางที่เราหมุนไว้ (Pivot) จัดเรียงแถวใหม่ตาม full_index (แถวไหนไม่เคยมีข้อมูลมาก่อน มันจะถูกสร้างขึ้นมาใหม่โดยมีค่าเป็นค่าว่างหรือ NaN)

ส่วนที่ 8: การตกแต่งและส่งออกผลลัพธ์

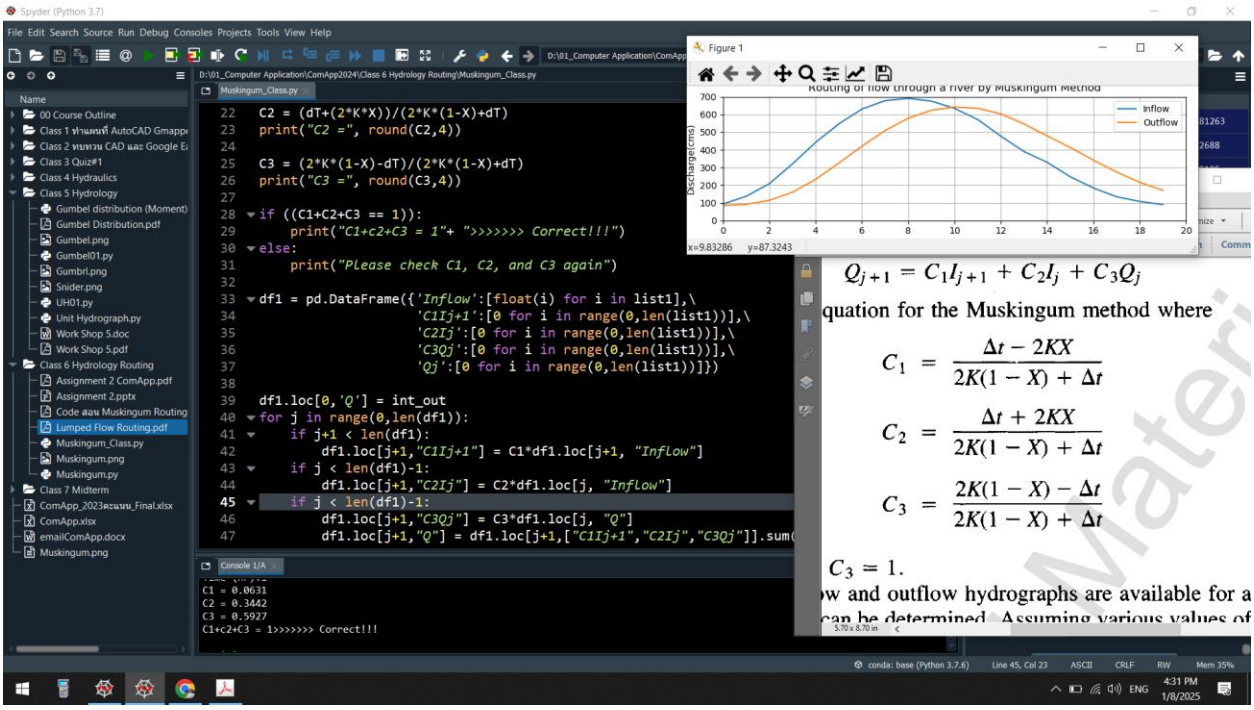
```
# 5. ตกแต่งตาราง
# ใช้แค่ DateStr เป็น Index
pivot.index = [x[3] for x in pivot.index]
pivot.index.name = 'Datetime'

# เติมช่องว่าง (รวมถึง 29 ก.พ. ในปีปกติ) ด้วยเครื่องหมายขีด (-)
pivot = pivot.fillna('-')

return pivot
```

- pivot.index = ...: ก่อนหน้านี้ Index ของเรามันซ้อนกันหลายชั้น (มีทั้ง MonthOrder, Month, Day) บรรทัดนี้คือการสั่งให้ดึงเฉพาะตัวที่ 3 (คือ DateStr เช่น 01-Apr) มาแสดงเป็นชื่อแถวเพียงอย่างเดียว
- pivot.index.name = ...: ตั้งชื่อหัวคอลัมน์ของแถวว่า Datetime
- pivot = pivot.fillna('-'): ค้นหาช่องที่ไม่มีข้อมูล (NaN) ซึ่งรวมถึงวันที่ 29-Feb ของปีปกติ และเติมเครื่องหมายขีดกลาง (-) ลงไปแทน
- return pivot: ส่งตารางที่เสร็จสมบูรณ์กลับไปให้ผู้ใช้งาน

อธิบายโค้ดโปรแกรม Muskingum Routing



ส่วนที่ 1: การนำเข้าเครื่องมือและรับข้อมูลน้ำไหลเข้า (Inflow)

```
import numpy as np
import pandas as pd
n = int(input("Length of list (Inflow): "))
list1 =[]
for i in range(n):
    element1 = input("Inflow: ")
    print(i+1)
    list1.append(element1)
df1 = pd.DataFrame(list1)
df1.columns = ["Inflow"]
df1 = df1.astype(float)
```

- บรรทัดที่ 1-2: นำเข้าไลบรารีคณิตศาสตร์ numpy และไลบรารีจัดการตาราง pandas
- บรรทัดที่ 3-8: โปรแกรมจะถามผู้ใช้ว่ามีข้อมูล Inflow กี่ตัว (n) จากนั้นจะวนลูป for เพื่อให้ผู้ใช้พิมพ์ป้อนค่า Inflow เข้ามาทีละค่าจนครบ และนำไปเก็บไว้ในลิสต์ที่ชื่อ list1
- บรรทัดที่ 9-11: นำลิสต์ข้อมูลมาแปลงเป็นตาราง Pandas DataFrame ตั้งชื่อคอลัมน์ว่า "Inflow" และแปลงชนิดข้อมูลให้เป็นตัวเลขทศนิยม (float) เพื่อเตรียมคำนวณ

ส่วนที่ 2: รับค่าพารามิเตอร์ของ Muskingum

```
int_out = float(input("Initial Outflow (cfs)"))
K = float(input("Propotionality Coefficient (K,(hr):"))
X = float(input("Weighting factor (X):"))
dT = float(input("Time (hr):"))
```

ส่วนนี้คือการรับค่าตัวแปรสำคัญ 4 ตัวในสมการ Muskingum จากผู้ใช้งาน:

- int_out (Initial Outflow): ปริมาณน้ำไหลออกเริ่มต้น (ณ เวลาที่ 0)
- K (Storage Time Constant): เวลาที่มวลน้ำเดินทางผ่านช่วงลำน้ำที่พิจารณา (หน่วยเป็นชั่วโมง)
- X (Weighting Factor): สัมประสิทธิ์การถ่วงน้ำหนัก (ค่าตั้งแต่ 0 ถึง 0.5) บ่งบอกถึงการยุบตัวของยอดน้ำ (Attenuation)
- dT (Delta t): ช่วงเวลา (Time step) ของข้อมูลน้ำ (เช่น ทุก 1 ชั่วโมง หรือ 2 ชั่วโมง)
-

ส่วนที่ 3: คำนวณค่าสัมประสิทธิ์ C1, C2, C3

```
# Calculation of C1, C2, and C3
C1 = (dT-(2*K*X))/(2*K*(1-X)+dT)
print("C1 =", round(C1,4))

C2 = (dT+(2*K*X))/(2*K*(1-X)+dT)
print("C2 =", round(C2,4))

C3 = (2*K*(1-X)-dT)/(2*K*(1-X)+dT)
print("C3 =", round(C3,4))

if ((C1+C2+C3 == 1)):
    print("C1+c2+C3 = 1"+ ">>>>>>> Correct!!!")
else:
    print("Please check C1, C2, and C3 again")
```

- บรรทัดที่ 1-8: แปลงสูตรทางคณิตศาสตร์ของ Muskingum เพื่อหาค่าสัมประสิทธิ์ C_1, C_2, C_3และพิมพ์ออกมาดูโดยปัดเศษทศนิยม 4 ตำแหน่ง (round(..., 4))
- บรรทัดที่ 10-13: จุดตรวจสอบสำคัญ (Quality Check): ตามหลักการสมดุลของมวล (Mass Balance) ผลรวมของ C_1 + C_2 + C_3 จะต้องเท่ากับ 1 เสมอ โค้ดส่วนนี้จะเช็คว่าสูตรคำนวณถูกต้องหรือไม่

ส่วนที่ 4: สร้างตารางเตรียมคำนวณ

```
df1 = pd.DataFrame({'Inflow':[float(i) for i in list1],\
                    'C1Ij+1':[0 for i in range(0,len(list1))],\
                    'C2Ij':[0 for i in range(0,len(list1))],\
                    'C3Qj':[0 for i in range(0,len(list1))],\
                    'Qj':[0 for i in range(0,len(list1))]})
```

- สร้างตาราง DataFrame ใหม่ โดยมีคอลัมน์ต่างๆ เพื่อเก็บผลลัพธ์ย่อยของสมการ ได้แก่:
  - Inflow: ข้อมูลน้ำไหลเข้า
  - C1Ij+1: เก็บผลลัพธ์ของ C_1 j+1
  - C2Ij: เก็บผลลัพธ์ของ C_ j
  - C3Qj: เก็บผลลัพธ์ของ C_3 Q_j
  - Qj: ช่องเตรียมไว้ (แต่ในลูปด้านล่าง ผู้เขียนโค้ดได้ไปสร้างคอลัมน์ชื่อ "Q" แทน)

ส่วนที่ 5: ลูปคำนวณปริมาณน้ำไหลออก (The Routing Loop) – *หัวใจสำคัญ*

```
df1.loc[0,'Q'] = int_out
for j in range(0,len(df1)):
    if j+1 < len(df1):
        df1.loc[j+1,"C1Ij+1"] = C1*df1.loc[j+1, "Inflow"]
    if j < len(df1)-1:
        df1.loc[j+1,"C2Ij"] = C2*df1.loc[j, "Inflow"]
    if j < len(df1)-1:
        df1.loc[j+1,"C3Qj"] = C3*df1.loc[j, "Q"]
        df1.loc[j+1,"Q"] = df1.loc[j+1,["C1Ij+1","C2Ij","C3Qj"]].sum()
```

- บรรทัดที่ 1: กำหนดให้ปริมาณน้ำไหลออกเวลาเริ่มต้น (บรรทัดแรก หรือ Index 0) มีค่าเท่ากับ Initial Outflow ที่ผู้ใช้ป้อนมา
- บรรทัดที่ 2-9: ลูปคำนวณสมการ Muskingum Q_j+1 = (C1 j+1) + (C2 j) + (C_3 Q_j)
  - ค่อยๆ ดึงค่า Inflow ของเวลาถัดไป (j+1) และเวลาปัจจุบัน (j) มาคูณกับ C1 และ C2
  - ดึงค่า Outflow เวลาปัจจุบัน (Q แถว j) มาคูณ C3
  - นำผลลัพธ์ทั้ง 3 ช่องมาบวกกัน (.sum()) แล้วเก็บเป็นค่า Outflow ในเวลาถัดไป (Q แถว j+1)

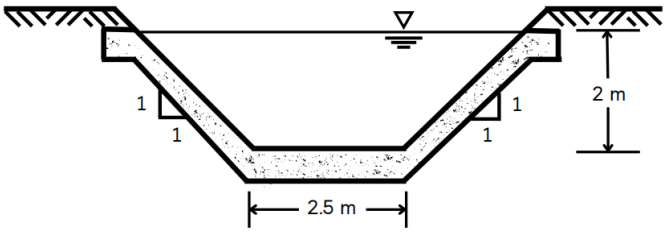
ส่วนที่ 6: การพล็อตกราฟ (Plotting Section)

```
# Plot Section
import matplotlib.pyplot as plt
plt.figure(figsize=(10,5))
plt.plot(df1["Inflow"],label="Inflow")
plt.plot(df1["Q"],label="Outflow")
plt.xlabel("Time(hr)")
plt.ylabel("Discharge(cms)")
plt.title("Routing of flow through a river by Muskingum Method")
plt.xticks(np.arange(0,21,2))
plt.xlim(0,20)
plt.yticks(np.arange(0,800,100))
plt.ylim(0,700)
plt.legend(ncol=1,loc="best")
plt.grid(True)
plt.tight_layout()
plt.savefig("Muskingum",dpi=300)
```

- ส่วนนี้ใช้ matplotlib.pyplot ในการวาดกราฟ Hydrograph เปรียบเทียบระหว่างมวลน้ำก้อนที่ไหลเข้า (Inflow) และมวลน้ำก้อนที่ไหลออก (Outflow)
- การตั้งค่าแกน: บังคับให้แกน X (เวลา) โชว์สเกล 0 ถึง 20 (ทีละ 2) และแกน Y โชว์สเกล 0 ถึง 700 (ทีละ 100)
- plt.grid(True): เปิดเส้นตารางหลังกราฟ
- plt.savefig("Muskingum",dpi=300): เซฟรูปกราฟที่ได้เป็นไฟล์ภาพความละเอียดสูง (300 dpi) ชื่อ "Muskingum.png" ไว้ในเครื่อง

การประยุกต์ใช้โปรแกรมคอมพิวเตอร์ด้วยภาษา Python

ข้อ 1. จงหาอัตราการไหลผ่านหน้าตัดคลองส่งน้ำดาดด้วยคอนกรีตที่มีความลาด 1 ต่อ 1000



ภาพที่ 1 : หน้าตัดคลองส่งน้ำดาดด้วยคอนกรีตที่มีความลาด 1 ต่อ 1000

- **วิธีทำ (Python Code)**

```
1  import math
2  n = float(input('n = Manning Value = '))
3  b = float(input('b (m) = '))
4  y = float(input('y (m) = '))
5  s = float(input('s = Channel Slope = '))
6  z = float(input('z = Side Slope of Channel = '))
7  # n = 0.014 , b = 2.5 , y = 2 , s = 0.001 , z = 1
8  A = 1/2 * (b + 6.5) * y
9  P = b + 2 * y * math.sqrt(1 + z**2)
10 R = A / P
11 Q = (1/n) * A * (R**(2/3)) * (s**0.5)
12 print("Q = ", round(Q, 2), "m\u00B3/s")
```

- **คำอธิบาย Python Code**

บรรทัดที่ 1 : นำเข้าฟังก์ชัน math เพื่อให้สามารถใช้ฟังก์ชันทางคณิตศาสตร์ได้
บรรทัดที่ 2 : รับค่า Manning (n) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 3 : รับค่าความกว้างท้องน้ำ (b) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 4 : รับค่าความลึกของน้ำ (y): โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 5 : รับค่าความลาดชันของร่องน้ำ (s): โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 6 : รับค่าความลาดชันด้านข้าง (z): โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 7 : คอมเมนต์ค่าต่างๆที่โจทย์กำหนดให้
บรรทัดที่ 8 : คำนวณพื้นที่หน้าตัด (A)
บรรทัดที่ 9 : คำนวณความยาวขอบเขตเปียก (P)
บรรทัดที่ 10 : คำนวณรัศมีชลศาสตร์ (R)
บรรทัดที่ 11 : คำนวณอัตราการไหล (Q)
บรรทัดที่ 12 : แสดงผลลัพธ์ค่าอัตราการไหล (Q) โดยให้แสดงเป็นเลขทศนิยม 2 ตำแหน่ง และมีหน่วยเป็น m^3/s

- **<u>ผลลัพธ์</u>**
    - n = Manning Value = 0.014
    - b (m) = 2.5
    - y (m) = 2
    - s = Channel Slope = 0.001
    - z = Side Slope of Channel = 1

    ∴ Q = 21.71 m^3/s

<u>ตอบ</u> อัตราการไหลผ่านหน้าตัดคลองส่งน้ำดาดด้วยคอนกรีตที่มีความลาด 1 ต่อ 1000 (Q) = 21.71 m^3/s

ข้อ 2. ถ้าต้องการระบายน้ำออกจากหมู่บ้านจัดสรรด้วยอัตราการไหล 0.5 cms โดยใช้รางคอนกรีตรูปสี่เหลี่ยมผืนผ้ากว้าง 0.50 m วางในแนวความลาด 20 cm ต่อความยาว 100 cm จะต้องออกแบบความลึกของรางคอนกรีตเท่าใด

- **วิธีทำ (Python Code)**

```
import math
n = float(input('n = Manning Value = '))
b = float(input('b(m) = '))
choice = input("คุณมีค่า Slope & Length หรือมีค่า Slope of Channel แล้ว? (หากมีค่า
Slope & Lengthให้ตอบ sl/หากมีค่า Slope of Channelให้ตอบ sc): ")
if choice.lower() == 'sl':
    slope = float(input("Slope (m) = "))
    length = float(input("Length (m) = "))
    s = slope / length
    print("s = Slope Channel = ", s)
elif choice.lower() == 'sc':
    s = float(input("s = Slope Channel = "))
else:
    print("กรุณาเลือก 'sl' หรือ 'sc' เท่านั้น")
Q = float(input('Q (m\u00B3/s) = '))

from scipy import optimize
def RectangularY(y):
    error = ((1/n) * (b*y) * (((b*y)/(b + (2*y)))**(2/3)) * (s**(1/2))) - Q
    return error
sol = optimize.root_scalar(RectangularY, bracket = [0,10], method='brentq')
y = sol.root
print('y = ',round(y,4),' m')
```

- **<u>คำอธิบาย Python Code</u>**

บรรทัดที่ 1 : นำเข้าโมดูล math ของ Python เพื่อใช้ฟังก์ชันคณิตศาสตร์ต่าง ๆ

บรรทัดที่ 2 : นำเข้าค่าจากผู้ใช้เป็น String ผ่าน input() โดยข้อความที่แสดงคือ 'n = Manning Value = ' แล้วแปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร n

บรรทัดที่ 3 : นำเข้าค่าความกว้างรางคอนกรีตเป็นเมตรจากผู้ใช้ เป็น String ผ่าน input() โดยข้อความที่แสดงคือ 'b(m) = ' แล้ว แปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร b

บรรทัดที่ 4-5 : ถามผู้ใช้ว่ามีค่า slope แบบใด โดยนำเข้าคำตอบจากผู้ใช้เป็น String ผ่าน input() โดยข้อความที่แสดง คือ ' คุณมีค่า Slope & Length หรือมีค่า Slope of Channel แล้ว? (หากมีค่า Slope & Lengthให้ตอบ sl/หากมีค่า Slope of Channelให้ตอบ sc): ' แล้วเก็บไว้ในตัวแปร choice

บรรทัดที่ 6 : ตรวจสอบเงื่อนไขว่าค่าที่ผู้ใช้ป้อน (.lower() คือ แปลงตัวอักษรที่ตอบเป็นพิมพ์เล็ก) เท่ากับ 'sl' หรือไม่ หากใช่ โปรแกรมจะเข้าไปทำตามคำสั่งภายในบล็อก if ต่อไป

บรรทัดที่ 7 : เมื่อผู้ใช้เลือก 'sl' โปรแกรมจะขอนำเข้าค่า Slope (ความลาด) ของรางคอนกรีต เป็นเมตรจากผู้ใช้ เป็นString ผ่าน input() โดยข้อความที่แสดง คือ 'Slope (m) = ' แล้ว แปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร Slope

บรรทัดที่ 8 : โปรแกรมจะขอนำเข้าค่า Length (ความยาว) ของรางคอนกรีต เป็นเมตรจากผู้ใช้ เป็น String ผ่าน input() โดยข้อความที่แสดง คือ 'Length (m) = ' แล้ว แปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร length

บรรทัดที่ 9 : คำนวณความชันของรางคอนกรีต (s) โดยใช้สูตร $s = \frac{\text{Slope (m)}}{\text{Length (m)}}$ ซึ่งให้ค่า slope เป็นอัตราส่วน (ไม่มีหน่วย)แล้วเก็บผลลัพธ์ไว้ในตัวแปร s

บรรทัดที่ 10 : แสดงค่าความชันของรางน้ำ s ที่คำนวณได้ให้ผู้ใช้ดู เพื่อยืนยันค่าว่าระบบรับค่าและคำนวณถูกต้อง โดยข้อความที่แสดง คือ ' s = Slope Channel = ' ตามด้วยค่าความชันของรางน้ำที่โปรแกรมคำนวณได้

บรรทัดที่ 11 : ถ้าเงื่อนไขแรกไม่เป็นจริง และคำตอบของผู้ใช้ (พิมพ์เล็ก) เท่ากับ 'sc' โปรแกรมจะทำตามคำสั่งในบล็อก elif ต่อไป

บรรทัดที่ 12 : เมื่อผู้ใช้เลือก 'sc' โปรแกรมจะขอนำเข้าค่า Slope of channel (ความลาดของท้องน้ำ) ของรางคอนกรีต จากผู้ใช้ เป็น String ผ่าน input() โดยข้อความที่แสดง คือ ' s = Slope Channel = ' แล้ว แปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร s

บรรทัดที่ 13-14 : บรรทัดนี้กำหนดกรณีสำรอง (default) หากคำตอบที่ผู้ใช้ป้อนใน choice ไม่ใช่ 'sl' หรือ 'sc' โปรแกรมจะเข้าสู่บล็อก else เพื่อจัดการกับข้อมูลนำเข้าที่ไม่ถูกต้องตามเงื่อนไข โดยจะแสดงข้อความเตือนผู้ใช้ว่าต้องเลือกเพียงหนึ่งในสองตัวเลือกคือ 'sl' หรือ 'sc' เท่านั้น

บรรทัดที่ 15 : นำเข้าค่าอัตราการไหลในรางคอนกรีตเป็น m^3/s จากผู้ใช้ เป็น String ผ่าน input() โดยข้อความที่แสดง คือ 'Q (m^3/s) = ' แล้ว แปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร Q

บรรทัดที่ 17 : นำเข้าโมดูล optimize จากไลบรารี scipy เพื่อใช้ฟังก์ชันเชิงตัวเลข

บรรทัดที่ 18 : ประกาศฟังก์ชันชื่อ RectangularY ที่รับพารามิเตอร์ y (ความลึกของรางคอนกรีต)

บรรทัดที่ 19 : ภายในฟังก์ชันคำนวณค่าอัตราการไหลตามสมการ Manning สำหรับรางคอนกรีตรูปสี่เหลี่ยมผืนผ้าดังนี้ :

$Q = \frac{1}{n} A R^{\frac{2}{3}} S^{\frac{1}{2}}$ โดยที่

พื้นที่หน้าตัดของรางน้ำ (A) = by

เส้นขอบเปียก (P) = b + 2y

รัศมีชลศาสตร์ (R) = $\frac{A}{P}$

ความลาดท้องน้ำ (s)

โดยในบรรทัดนี้ จะคำนวณ $Q_{\text{ที่คำนวณได้}} - Q_{\text{ที่ผู้ใช้ป้อน}}$ แล้วเก็บผลลัพธ์ไว้ในตัวแปร error เพื่อใช้เป็นฟังก์ชันสำหรับการหา root เพื่อหาค่าความลึกของรางน้ำ y (ค่าที่ทำให้ error = 0 คือ คำตอบสำหรับ y)

บรรทัดที่ 20 : ให้ฟังก์ชัน RectangularY คืนค่า error ซึ่งจะใช้เพื่อหาค่า y ที่ทำให้ค่า error = 0

บรรทัดที่ 21 : เรียกตัวแก้สมการเชิงตัวเลข root_scalar ของ scipy.optimize เพื่อหา y ที่ทำให้ RectangularY (y) = 0

- RectangularY เป็นฟังก์ชันที่ต้องการหา root เพื่อหาค่า y
- Bracket = [0,10] ระบุช่วงเริ่มต้นที่คาดว่ามีค่ารากอยู่ (0 ถึง 10 เมตร)
- method = 'brentq' ใช้วิธี Brent's method (brentq) ซึ่งเป็นวิธีที่แม่นยำและเสถียร แต่ต้องการให้ค่าฟังก์ชันที่ขอบทั้งสองมีสัญลักษณ์ต่างกัน (+, -) มิฉะนั้นจะเกิดข้อผิดพลาด

แล้วเก็บค่าในตัวแปร sol

บรรทัดที่ 22 : ดึงค่ารากที่พบออกมาเป็น sol.root แล้วเก็บไว้ในตัวแปร y ซึ่งเป็นค่าความลึกของรางน้ำที่ต้องการ

บรรทัดที่ 23 : แสดงค่าความลึกของรางน้ำ y ที่คำนวณได้บนหน้าจอ โดยใช้ round() ปัดเศษทศนิยมให้เหลือ 4 ตำแหน่ง และแสดงหน่วยเป็นเมตร โดยข้อความที่แสดงคือ 'y = ... m '

- **<u>ผลลัพธ์</u>**
  - n = Manning Value = 0.014
  - b(m) = 0.50
  - คุณมีค่า Slope & Length หรือมีค่า Slope of Channel แล้ว? (หากมีค่าSlope & Lengthให้ตอบ sl/หากมีค่า Slope of Channelให้ตอบ sc): sl
  - Slope (m) = 0.2
  - Length (m) = 100
  - s = Slope Channel = 0.002
  - Q (m^3/s) = 0.5

    ∴ y = 0.9252 m

<u>ตอบ</u> จะต้องออกแบบความลึกของรางคอนกรีต (y) = 0.9252 m

ข้อ 3. จากโจทย์ ข้อ 2 ถ้าออกแบบระบบระบายน้ำโดยใช้ท่อคอนกรีตเสริมเหล็กแทนรางคอนกรีต จะต้องวางท่อที่มีขนาดเส้นผ่าศูนย์กลางเท่าใด จึงจะเพียงพอต่อการระบายน้ำได้

- **วิธีทำ (Python Code)**

```
# When n = 0.014, Q = 0.5, s = 0.002
import math
n = float(input('n = Manning Value = '))
s = float(input('s = Channel Slope = '))
Q = float(input('Q (cms) = '))
pi = math.pi
from scipy import optimize
def Diameter(D):
    error = (1/n)*((pi * D**2) / 4) * (D/4)**(2/3) * (s**0.5) - Q
    return error
sol = optimize.root_scalar(Diameter, bracket=[0, 10], method='brentq')
D = sol.root
print('D =', round(D, 3), 'm')
```

- **คำอธิบาย Python Code**

บรรทัดที่ 1 : คอมเมนต์: ระบุค่าพารามิเตอร์ที่ใช้เป็นกรณีศึกษา ได้แก่ สัมประสิทธิ์ความขรุขระ (n), อัตราการไหล (Q), และความลาดชัน (s)

บรรทัดที่ 2 : การนำเข้าโมดูล: นำเข้าโมดูล math เพื่อให้สามารถใช้ค่าคงที่ π (math.pi) ในการคำนวณพื้นที่หน้าตัดท่อ

บรรทัดที่ 3 : การรับอินพุต: รับค่าสัมประสิทธิ์ความขรุขระของแมนนิ่ง (n) จากผู้ใช้ โดยแปลงเป็นตัวเลขทศนิยม

บรรทัดที่ 4 : การรับอินพุต: รับค่าความลาดชันของท่อ (s) จากผู้ใช้ โดยแปลงเป็นตัวเลขทศนิยม

บรรทัดที่ 5 : การรับอินพุต: รับค่าอัตราการไหลที่ต้องการ (Q) ในหน่วยลูกบาศก์เมตรต่อวินาที (cms หรือ m^3/s)

บรรทัดที่ 6 : การกำหนดค่าคงที่: กำหนดตัวแปร pi ให้มีค่าเท่ากับค่าคงที่ π

บรรทัดที่ 7 : การนำเข้าโมดูล: นำเข้าโมดูล optimize จากไลบรารี scipy ซึ่งมีเครื่องมือสำหรับแก้สมการที่ไม่สามารถหาคำตอบโดยตรงได้ (Non-linear Equation Solver)

บรรทัดที่ 8 : การสร้างฟังก์ชัน: กำหนดฟังก์ชันชื่อ Diameter ที่จะใช้ในการแก้สมการ โดยรับอินพุตเพียงตัวแปรเดียวคือ D (เส้นผ่านศูนย์กลาง)

บรรทัดที่ 9 : สมการหลักของแมนนิ่ง : คำนวณค่า error ซึ่งคือส่วนต่างระหว่าง Q_calculated กับ Q_target เป้าหมายคือการหาค่า D ที่ทำให้ error = 0

บรรทัดที่ 10 : การส่งค่ากลับ: ส่งค่า error ที่คำนวณได้กลับไปยังฟังก์ชัน root_scalar เพื่อปรับค่า D ในการวนซ้ำครั้งต่อไป

บรรทัดที่ 11 : การเรียกใช้ตัวแก้สมการ: ใช้ฟังก์ชัน root_scalar เพื่อค้นหาค่า D ที่เป็นรากของฟังก์ชัน Diameter โดย Diameter คือ ฟังก์ชันที่ต้องการให้ผลลัพธ์เป็น 0
bracket=[0,10] คือการกำหนดช่วงเริ่มต้นที่คาดว่าคำตอบจะอยู่ (ตั้งแต่ 0 ถึง 10 เมตร)
method='brentq' คือวิธีการค้นหาค่าราก ซึ่งเป็นวิธีการที่มีความแม่นยำสูงและรวดเร็ว

บรรทัดที่ 12 : การดึงผลลัพธ์: ดึงค่ารากที่พบ (D) จากอ็อบเจกต์ผลลัพธ์ (sol) มาเก็บในตัวแปร D

บรรทัดที่ 13 : การแสดงผลลัพธ์: แสดงค่าเส้นผ่านศูนย์กลาง D ที่คำนวณได้ โดยปัดเศษทศนิยมให้เหลือ 3 ตำแหน่ง พร้อมระบุหน่วยเป็นเมตร

- **<u>ผลลัพธ์</u>**
  - n = Manning Value = 0.014
  - s = Channel Slope = 0.002
  - Q (cms) = 0.5

∴ D = 0.772 m

<u>ตอบ</u> จะต้องวางท่อที่มีขนาดเส้นผ่าศูนย์กลาง (D) = 0.772 m

ข้อ 4. จงหาอัตราการไหลผ่านรางระบายน้ำคอนกรีตข้างถนนที่มีน้ำลึก 10 cm และความลาด 10 cm ต่อความยาว 100 m

ภาพที่ 2 : รางระบายน้ำคอนกรีตข้างถนน

- **วิธีทำ (Python Code)**

```python
import math

n = float(input('n = Manning Value = '))
b = float(input('b (m) = '))
y = float(input('y (m) = '))
s = float(input('s = Channel Slope = '))
# n = 0.014 , b = 0 , y = 0.1 , s = 0.001

A = 1/2 * (b * y)
P = math.sqrt((b**2)+(y**2)) + y
R = A / P
Q = (1/n) * A * R**(2/3) * (s**0.5)
print('Q =', round(Q, 4), 'm\u00B3/s')
```

- **คำอธิบาย Python Code**

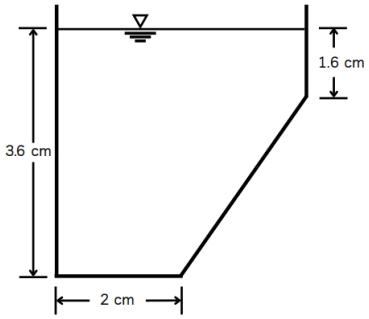
บรรทัดที่ 1 : นำเข้าฟังก์ชัน math เพื่อให้สามารถใช้ฟังก์ชันทางคณิตศาสตร์ได้
บรรทัดที่ 3 : รับค่า Manning (n) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 4 : รับค่าความกว้างท้องน้ำ (b) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 5 : รับค่าความลึกของน้ำ (y): โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 6 : รับค่าความลาดชันของร่องน้ำ (s): โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 7 : ค่าต่างๆที่โจทย์กำหนดให้
บรรทัดที่ 9 : คำนวณพื้นที่หน้าตัด (A)
บรรทัดที่ 10 : คำนวณความยาวขอบเขตเปียก (P)
บรรทัดที่ 11 : คำนวณรัศมีชลศาสตร์ (R)
บรรทัดที่ 12 : คำนวณอัตราการไหล (Q)
บรรทัดที่ 13 : แสดงผลลัพธ์ค่าอัตราการไหล (Q)

- **<u>ผลลัพธ์</u>**
  - n = Manning Value = 0.014
  - b (m) = 100
  - y (m) = 0.1
  - s = Channel Slope = 0.001

∴ Q = 1.5318 m^3/s

<u>ตอบ</u> อัตราการไหลผ่านรางระบายน้ำคอนกรีตข้างถนน (Q) = 1.5318 m^3/s

ข้อ 5. คลองคอนกรีต (n = 0.013) ดังรูป ถ้าน้ำไหลแบบสม่ำเสมอมีอัตราการไหลของน้ำในคลอง 30 cms จงหาความลาดในแนวดิ่งของพื้นคลองที่ลดระดับลงต่อความยาวคลอง 1 km



ภาพที่ 3 : คลองคอนกรีต

- **วิธีทำ (Python Code)**

```
import math

Q = float(input("discharge Q (cms): "))
n = float(input("Manning (n): "))
L_km = float(input("canal length (km): "))
L = L_km * 1000.0

print("cross-section dimensions (meters):")
bottom_width = float(input("Bottom width (m) = "))
top_width = float(input("Top width (m) = "))
left_depth = float(input("Left depth (m) = "))
right_wall_above_slope = float(input("Right depth above slope (m) = "))


def polygon_area(bottom_width, top_width, left_depth, right_wall_above_slope):
    area_rec = top_width * right_wall_above_slope
    area_trapezoidal = 0.5 * (top_width + bottom_width) * (left_depth –
    right_wall_above_slope)
    area = area_rec + area_trapezoidal
    return area
A = polygon_area(bottom_width, top_width, left_depth, right_wall_above_slope)
slope = math.sqrt(((top_width - bottom_width)**2) + ((left_depth -
right_wall_above_slope)**2))
P = left_depth + bottom_width + slope + right_wall_above_slope
```

```
24 R = A / P
25 Sf = ((Q * n )/ (A * R**(2.0 / 3.0)))**2
26 drop = Sf * L
27
28 print("Water surface drop over", round(L_km, 3), "km =", round(drop, 3), "m")
```

- **<u>คำอธิบาย Python Code</u>**

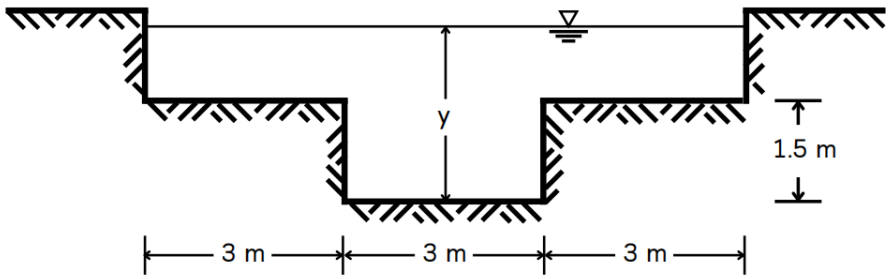
บรรทัดที่ 1 : นำเข้าโมดูล math เพื่อใช้ในการคำนวณ
บรรทัดที่ 3 : รับค่าอัตราการไหล (Q) เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 4 : รับค่า Manning (n) เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 5 : รับค่า ความยาวคลอง(km) เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 6 : แปลงความยางคลองจาก กิโลเมตรเป็นเมตร
บรรทัดที่ 9 : รับค่าความยาวด้านข้าง เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 10 : รับค่าความยาวด้านบน เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 11 : รับค่าความสูงด้านซ้าย เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 12 : รับค่าความสูงด้านขวาเหนือส่วนที่เอียง เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 15 : นิยามฟังก์ชันชื่อ polygon_area เพื่อหาค่าพื้นที่รูปหลายเหลี่ยม
บรรทัดที่ 16 : คำนวณพื้นที่ช่วงที่เป็นสี่เหลี่ยมผืนผ้าด้านบน
บรรทัดที่ 17 : คำนวณพื้นที่ช่วงที่เป็นสี่เหลี่ยมคางหมูด้านล่าง
บรรทัดที่ 18 : นำพื้นที่สี่เหลี่ยมด้านบนและด้านล่างมารวมกัน
บรรทัดที่ 19 : ส่งคืนค่าพื้นที่รวมออกไปยังฟังก์ชัน
บรรทัดที่ 21 : ให้ค่า A รับค่าพื้นที่จากฟังก์ชัน polygon_area
บรรทัดที่ 22 : หาความยาวส่วนที่เป็นทางลาดของขอบคลอง
บรรทัดที่ 23 : คำนวณหาความยาวขอบเขตเปียก
บรรทัดที่ 24 : คำนวณหารัศมีชลศาสตร์
บรรทัดที่ 25 : คำนวณหาค่า friction slope
บรรทัดที่ 26 : คำนวณหาความลาดในแนวดิ่งของพื้นคลอง
บรรทัดที่ 28 : แสดงผลความลาดในแนวดิ่งของพื้นคลอง

- **<u>ผลลัพธ์</u>**
  - discharge Q (cms): 30
  - Manning (n): 0.013
  - canal length (km): 1
  - cross-section dimensions (meters):
  - Bottom width (m) = 2
  - Top width (m) = 4
  - Left depth (m) = 3.6
  - Right depth above slope (m) = 1.6

∴ Water surface drop over 1.0 km = 0.745 m

<u>ตอบ</u> ความลาดในแนวดิ่งของพื้นคลองที่ลดระดับลงต่อความยาวคลอง 1 km = 0.745 m

ข้อ 6. คลองส่งน้ำแห่งหนึ่งมีสัมประสิทธิ์ความขรุขระ 0.017 และความลาด 0.0009 เมื่อมีอัตราการไหลเท่ากับ 34 cms จงหาความลึกน้ำในคลองส่งน้ำนี้



ภาพที่ 4 : คลองส่งน้ำ

- **วิธีทำ (Python Code)**

```
#%% หาค่า y ในคลองส่งน้ำรูปหน้าตัดที่โจทย์กำหนด
n = float(input('n = Manning Value = '))
s = float(input('s = Channel Slope = '))
Q = float(input('Q(cms) = '))
i = float(input('i ความสูงส่วนล่าง(m) = '))
j = float(input('j ความยาวฐานฝั่งซ้าย(m) = '))
k = float(input('k ความยาวฐานตรงกลาง(m) = '))
l = float(input('l ความยาวฐานฝั่งขวา(m) = '))
def Area_R(y):
    if y >= i:
        A = (y-i)*(j+k+l)+k*i
        P = 2*(y-i)+(j+k+l)+2*i
    else:
        A = i*k
        P = 2*i+k
    R = A/P
    return A,R
from scipy import optimize
def TshapeY(y):
    A,R = Area_R(y)
    error = (1/n)*A*(R**(2/3))*(s**0.5)-Q
    return error
sol = optimize.root_scalar(TshapeY, bracket=[0, 10], method='brentq')
y = sol.root
print('y =',round(y,3),'m')
```

- **<u>คำอธิบาย Python Code</u>**

บรรทัดที่ 1 : คอมเมนต์ (comment) ใช้เพื่ออธิบายว่าบล็อกโค้ดถัดไปเป็นการคำนวณ “หาค่า y ในคลองส่งน้ำรูปห้านาตัดที่โจทย์กำหนด” Python จะ ไม่ประมวลผลบรรทัดนี้

บรรทัดที่ 2 : รับค่า Manning (n) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 3 : รับค่าความลาดชันของร่องน้ำ (s): โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 4 : รับค่าอัตราการไหล (Q) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 5 : รับค่าความสูงส่วนล่าง (i) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 6 : รับค่าความยาวฐานฝั่งซ้าย (j) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 7 : รับค่าความยาวฐานตรงกลาง (k) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 8 : รับค่าความยาวฐานฝั่งขวา (l) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 9-17 : ตั้งเงื่อนไขตัวแปรที่เกี่ยวกับภาพหน้าตัด โดยแบ่งเป็นกรณีที่ค่า y มากกว่าและน้อยกว่าค่า l (ความสูงส่วนล่าง)

บรรทัดที่ 18 : นำเข้าโมดูล optimize จาก Scipy เพื่อใช้ฟังก์ชันหา root

บรรทัดที่ 19-22 : สร้างฟังก์ชัน TshapeY(y) โดยใช้สมการ Manning : $\frac{1}{n} AR^{\frac{2}{3}} S^{\frac{1}{2}}$ และส่งกลับค่าความคลาดเคลื่อน (error)

บรรทัดที่ 23-24 : ใช้การหา root จากฟังก์ชัน TshapeY และกำหนดช่วงของคำตอบ [0,10] จากนั้นใช้วิธี Brent’s method ซึ่งคำตอบที่ได้อยู่ใน sol.root

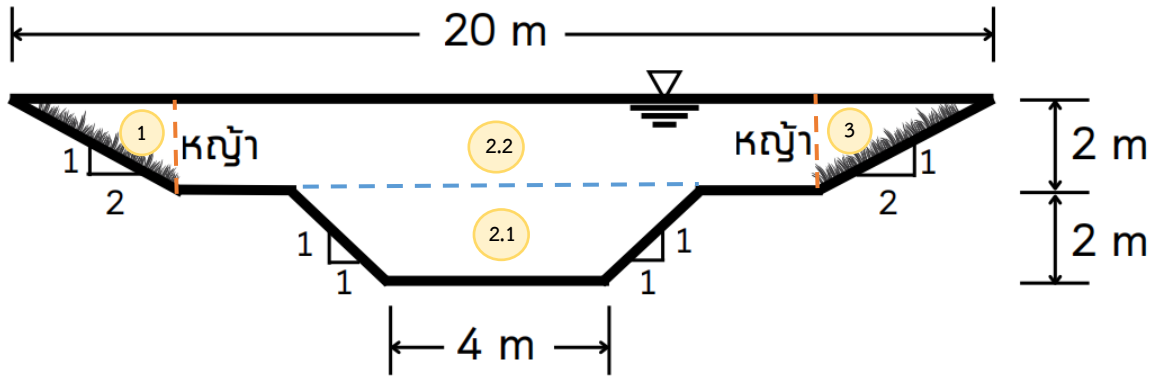
บรรทัดที่ 25 : แสดงผลลัพธ์ค่าความสูงของคลองส่งน้ำ (y) โดยใช้ round (y,3) ปัดเศษให้เหลือทศนิยม 3 ตำแหน่ง หน่วยเป็น m

- **<u>ผลลัพธ์</u>**
  - n = Manning Value = 0.017
  - s = Channel Slope = 0.0009
  - Q(cms) = 34
  - i ความสูงส่วนล่าง(m) = 1.5
  - j ความยาวฐานฝั่งซ้าย(m) = 3
  - k ความยาวฐานตรงกลาง(m) = 3
  - l ความยาวฐานฝั่งขวา(m) = 3

  ∴ y = 2.929 m

<u>ตอบ</u> ความลึกน้ำในคลองส่งน้ำนี้ (y) = 2.929 m

ข้อ 7. จงหาอัตราการไหลผ่านทางน้ำเปิดที่มีความลาด 0.001



ภาพที่ 5 : หน้าตัดทางน้ำ

- **วิธีทำ (Python Code)**

แบ่งรูปหน้าตัดออกเป็น 4 รูป ตามภาพ

```
import math
n1 = float(input('n1 = Manning Value(grass) = '))
n2 = float(input('n2 = Manning Value(clay) = '))
b1 = float(input('b-Triangle left (m) = '))
y1 = float(input('y-Triangle left (m) = '))
b211 = float(input('b-Trapezoid long edge (m) = '))
b212 = float(input('b-Trapezoid short edge (m) = '))
y21 = float(input('y-Trapezoid (m) = '))
b22 = float(input('b-Rectangle (m) = '))
y22 = float(input('y-Rectangle (m) = '))
b3 = float(input('b-Triangle Right (m) = '))
y3 = float(input('y-Triangle Right (m) = '))
s = float(input('s = Slope Channel = '))
z1 = float(input('z (Triangle) = Side Slope of Canal = '))
z2 = float(input('z (Trapezoid) = Side Slope of Canal = '))
A1 = (1/2)*b1*y1
A2 = ((1/2)*(b211+b212)*y21)+(b22*y22)
A3 = (1/2)*b3*y3
P1 = y1*(math.sqrt(1+(z1**2)))
P2 = (b212 + ((2*y21*(math.sqrt(1+(z2**2))))))+(b22-b211)
P3 = y3*(math.sqrt(1+(z1**2)))
R1 = A1/P1
R2 = A2/P2
```

```
24 R3 = A3/P3
25 Q1 = (1/n1) * A1 * (R1**(2/3)) * (s**(1/2))
26 Q2 = (1/n2) * A2 * (R2**(2/3)) * (s**(1/2))
27 Q3 = (1/n1) * A3 * (R3**(2/3)) * (s**(1/2))
28 Qall = Q1 + Q2 + Q3
29 print('Qall = ',round(Qall, 4),' m\u00b3/s')
```

- <u>คำอธิบาย Python Code</u>

บรรทัดที่ 1 : นำเข้าโมดูล math ของ Python เพื่อใช้ฟังก์ชันคณิตศาสตร์ต่าง ๆ

บรรทัดที่ 2 : นำเข้าค่า Manning ของพื้นหญ้าจากผู้ใช้เป็น String ผ่าน input() โดยข้อความที่แสดงคือ 'n1 = Manning Value (grass) = ' แล้วแปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร n1

บรรทัดที่ 3 : นำเข้าค่า Manning ของพื้นดินเหนียวบดอัดแน่นจากผู้ใช้เป็น String ผ่าน input() โดยข้อความที่แสดงคือ 'n2 = Manning Value (clay) = ' แล้วแปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร n2

บรรทัดที่ 4-5 : นำเข้าค่าความกว้าง (b1) และความลึก (y1) ของทางน้ำ (รูปที่ 1) จากผู้ใช้เป็น String ผ่าน input() โดยข้อความที่แสดงคือ 'b-Triangle left (m) = ' ตามด้วย 'y-Triangle left (m) = ' แล้วแปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร b1 และ y1 ตามลำดับ

บรรทัดที่ 6-8 : นำเข้าค่าความยาวด้านและความลึกของทางน้ำ โดย ความยาวด้านยาวของรูป (b211), ความยาวด้านสั้นของรูป (b212) และความลึกของรูป (y21) ของทางน้ำ (รูปที่ 2.1) จากผู้ใช้เป็น String ผ่าน input() โดยข้อความที่แสดงคือ 'b-Trapezoid long edge (m) = ' ตามด้วย 'b-Trapezoid short edge (m) = ' และ 'y-Trapezoid (m) = ' แล้วแปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร b211, b212 และ y21 ตามลำดับ

บรรทัดที่ 9-10 : นำเข้าค่าความกว้าง (b22) และความลึก (y22) ของทางน้ำ (รูปที่ 2.2) จากผู้ใช้เป็น String ผ่าน input() โดยข้อความที่แสดงคือ 'b- Rectangle (m) = ' ตามด้วย 'y- Rectangle (m) = ' แล้วแปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร b22 และ y22 ตามลำดับ

บรรทัดที่ 11-12 : นำเข้าค่าความกว้าง (b3) และความลึก (y3) ของทางน้ำ (รูปที่ 3) จากผู้ใช้เป็น String ผ่าน input() โดยข้อความที่แสดงคือ 'b-Triangle right (m) = ' ตามด้วย 'y-Triangle right (m) = ' แล้วแปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร b3 และ y3 ตามลำดับ

บรรทัดที่ 13 : นำเข้าค่า s (Slope channel) จากผู้ใช้เป็น String ผ่าน input() โดยข้อความที่แสดงคือ 's = Slope Channel = ' แล้วแปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร s

บรรทัดที่ 14-15 : นำเข้าค่า Side slope ของทางน้ำ จากผู้ใช้เป็น String ผ่าน input() โดยข้อความที่แสดงคือ 'z (Triangle) = Side Slope of Canal = ' ตามด้วย 'z (Trapezoid) = Side Slope of Canal = '
แล้วแปลงค่าที่รับเข้ามาเป็น float แล้วเก็บไว้ในตัวแปร z1 และ z2 ตามลำดับ

บรรทัดที่ 16 : คำนวณพื้นที่รูปที่ 1 เก็บไว้ในตัวแปร A_1 – พื้นที่รูปสามเหลี่ยม $A_1 = \frac{1}{2} b_1 y_1$

บรรทัดที่ 17 : คำนวณพื้นที่รูปที่ 2.1+2.2 เก็บไว้ในตัวแปร A_2 – พื้นที่รูปสี่เหลี่ยมคางหมู + พื้นที่รูปสี่เหลี่ยมผืนผ้า

$A_2 = (\frac{1}{2} \times (b_{211} + b_{212}) \times y_{21}) + (b_{22} \times y_{22})$

บรรทัดที่ 18 : คำนวณพื้นที่รูปที่ 3 เก็บไว้ในตัวแปร A_3 – พื้นที่รูปสามเหลี่ยม $A_3 = \frac{1}{2} b_3 y_3$

บรรทัดที่ 19 : คำนวณเส้นขอบเปียกรูปที่ 1 เก็บไว้ในตัวแปร P_1 – เส้นขอบเปียกรูปสามเหลี่ยม

$P_1 = y_1 \sqrt{1+z_1^2}$

บรรทัดที่ 20 : คำนวณเส้นขอบเปียกรูปที่ 2.1+2.2 เก็บไว้ในตัวแปร P_2 – เส้นขอบเปียกรูปสี่เหลี่ยมคางหมู + เส้นขอบเปียกรูปสี่เหลี่ยมผืนผ้า

$P_2 = (b_{212} + (2y_{21} \sqrt{1+z_2^2}) + (b_{22} - b_{211})$

บรรทัดที่ 21 : คำนวณเส้นขอบเปียกรูปที่ 3 เก็บไว้ในตัวแปร P_3 – เส้นขอบเปียกรูปสามเหลี่ยม

$P_3 = y_3 \sqrt{1+z_1^2}$

บรรทัดที่ 22-24 : คำนวณรัศมีชลศาสตร์ของรูปทั้ง 4 รูป แล้วเก็บไว้ในตัวแปร R_1, R_2 และ R_3 ตามลำดับ โดย

$R = \frac{A}{P}$

บรรทัดที่ 25-27 : คำนวณอัตราการไหลของรูปทั้ง 4 รูป แล้วเก็บไว้ในตัวแปร Q_1, Q_2 และ Q_3 ตามลำดับ โดย

$Q = \frac{1}{n} A R^{\frac{2}{3}} S^{\frac{1}{2}}$

บรรทัดที่ 28 : คำนวณอัตราการไหลรวมของทางน้ำ แล้วเก็บไว้ในตัวแปร Q_{all} โดย $Q_{all} = Q_1 + Q_2 + Q_3$

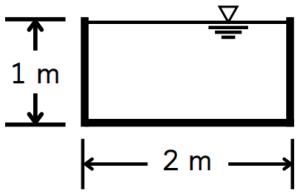
บรรทัดที่ 29 : แสดงค่า Q_{all} ออกทางหน้าจอ โดยปัดเศษเป็น 4 ตำแหน่งทศนิยม ใช้ \u00b3 เพื่อแสดงสัญลักษณ์ยกกำลังสาม (³) ในหน่วย m³/s โดยข้อความที่แสดงคือ 'Q_{all} = ... m³/s '

- **<u>ผลลัพธ์</u>**
  - n_1 = Manning Value (grass) = 0.030
  - n_2 = Manning Value (clay) = 0.014
  - b_1 = b-Triangle left (m) = 4 m
  - y_1 = y-Triangle left (m) = 2 m
  - b_{211} = b-Trapezoid long edge (m) = 8 m
  - b_{212} = b-Trapezoid short edge (m) = 4 m
  - y_{21} = y-Trapezoid (m) = 2 m
  - b_{22} = b- Rectangle (m) = 12 m
  - y_{22} = y- Rectangle (m) = 2 m
  - b_3 = b-Triangle right (m) = 4 m
  - y_3 = y-Triangle right (m) = 2 m
  - s = Slope Channel = 0.001
  - z (Triangle) = Side Slope of Canal = 2
  - z (Trapezoid) = Side Slope of Canal = 1

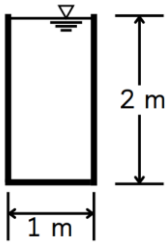
$\therefore Q_{all} = 162.9987\ m^3/s$

<u>ตอบ</u> อัตราการไหลผ่านทางน้ำเปิดนี้ คือ (Q_{all}) = 162.9987 m^3/s

ข้อ 8. รางส่งน้ำคอนกรีต (n = 0.012) วางตามแนวความลาด 1 m ต่อความยาว 1 km ดังรูป



ภาพที่ 6 : รางน้ำแบบที่ 1



ภาพที่ 7 : รางน้ำแบบที่ 2

(ก) จงหาอัตราการไหลในรางส่งน้ำทั้ง 2 ราง

(ข) จากผลในข้อ (ก) ถ้าต้องการออกแบบรางส่งน้ำที่มีพฤติกรรมการไหลที่ดีที่สุดในทางชลศาสตร์ วิศวกรจะเลือกออกแบบรางน้ำหน้าตัดใด จงอธิบายเหตุผลประกอบการ ตัดสินใจ

- **วิธีทำ (Python Code)**

```
# รางน้ำแบบที่ 1
n = float(input('n = Manning Value = '))
s = float(input('s = Channel Slope = '))
b1 = float(input('b1(m) = '))
y1 = float(input('y1(m) = '))
# n=0.012, b1=2, y1=1, s=0.001
A1 = b1*y1
P1 = b1+(2*y1)
R1 = A1/P1
Q1 = (1/n)*A1*R1**(2/3)*(s**0.5)
# รางน้ำแบบที่ 2
b2 = float(input('b2(m) = '))
y2 = float(input('y2(m) = '))
# n=0.012, b2=1, y2=2, s=0.001
A2 = b2*y2
P2 = b2+(2*y2)
R2 = A2/P2
Q2 = (1/n)*A2*R2**(2/3)*(s**0.5)
print('Q1 =',round(Q1,2),'m\u00b3/s') #cms
print('Q2 =',round(Q2,2),'m\u00b3/s') #cms
```

- **<u>คำอธิบาย Python Code</u>**

บรรทัดที่ 1 : คอมเมนต์ (comment) ใช้เพื่ออธิบายว่าบล็อกโค้ดถัดไปเป็นการคำนวณ “รางน้ำแบบที่ 1” Python จะ ไม่ประมวลผลบรรทัดนี้

บรรทัดที่ 2 : รับค่า Manning (n) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 3 : รับค่าความลาดชันของร่องน้ำ (s): โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 4 : รับค่าความกว้างท้องน้ำ (b1) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 5 : รับค่าความลึกของน้ำ (y1): โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 6 : คอมเมนต์ค่าต่างๆที่โจทย์กำหนดให้ เป็นคอมเมนต์ (ไม่ถูกทำงานจริง)

บรรทัดที่ 7 : คำนวณพื้นที่หน้าตัด (A1) สูตร = b1 * y1

บรรทัดที่ 8 : คำนวณความยาวขอบเขตเปียก (P1) สูตร = b1+ (2 * y1)

บรรทัดที่ 9 : คำนวณรัศมีชลศาสตร์ (R1) สูตร = A1/P1

บรรทัดที่ 10 : คำนวณอัตราการไหล (Q1) โดยใช้สมการ Manning ; $\frac{1}{n} A_1 R_1^{\frac{2}{3}} S^{\frac{1}{2}}$

บรรทัดที่ 11 : คอมเมนต์ (comment) ใช้เพื่ออธิบายว่าบล็อกโค้ดถัดไปเป็นการคำนวณ “รางน้ำแบบที่ 2” Python จะ ไม่ประมวลผลบรรทัดนี้

บรรทัดที่ 12 : รับค่าความกว้างท้องน้ำ (b2) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 13 : รับค่าความลึกของน้ำ (y2): โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 14 : คอมเมนต์ค่าต่างๆที่โจทย์กำหนดให้ เป็นคอมเมนต์ (ไม่ถูกทำงานจริง)

บรรทัดที่ 15 : คำนวณพื้นที่หน้าตัด (A2) สูตร = b2 * y2

บรรทัดที่ 16 : คำนวณความยาวขอบเขตเปียก (P2) สูตร = b2+ (2 * y2)

บรรทัดที่ 17 : คำนวณรัศมีชลศาสตร์ (R2) สูตร = A2/P2

บรรทัดที่ 18 : คำนวณอัตราการไหล (Q2) โดยใช้สมการ Manning ; $\frac{1}{n} A_2 R_2^{\frac{2}{3}} S^{\frac{1}{2}}$

บรรทัดที่ 19 : แสดงผลลัพธ์ค่าอัตราการไหล (Q1) โดยใช้ round (Q1,2) ปัดเศษให้เหลือทศนิยม 2 ตำแหน่ง หน่วยเป็น m^3/s (cms)

บรรทัดที่ 20 : แสดงผลลัพธ์ค่าอัตราการไหล (Q2) โดยใช้ round (Q1,2) ปัดเศษให้เหลือทศนิยม 2 ตำแหน่ง หน่วยเป็น m^3/s (cms)

- **<u>ผลลัพธ์</u>**
  - n = Manning Value = 0.012
  - s = Channel Slope = 0.001
  - b1(m) = 2
  - y1(m) = 1
  - b2(m) = 1
  - y2(m) = 2

∴ Q1 = 3.32 m^3/s และ Q2 = 2.86 m^3/s

<u>ตอบ</u>

(ก) อัตราการไหลของรางน้ำแบบที่ 1 (Q1) = 3.32 m^3/s และ รางน้ำแบบที่ 2 (Q2) = 2.86 m^3/s

(ข) อธิบายได้ว่า จากผลในข้อ (ก) ถ้าต้องการออกแบบรางส่งน้ำที่มีพฤติกรรมการไหลที่ดีที่สุดในทางชลศาสตร์ วิศวกรควรจะเลือกออกแบบ รางส่งน้ำแบบที่ 1 เพราะให้การไหลที่มีประสิทธิภาพดีกว่า และมีรัศมีไฮดรอลิกมากกว่า

ข้อ 9. จงออกแบบหน้าตัดดีที่สุดทางชลศาสตร์ของคลองส่งน้ำคอนกรีต (n = 0.012) รูปตัดสี่เหลี่ยมผืนผ้าเพื่อส่งน้ำด้วยความเร็วไม่เกิน 2.4 m/s ตามแนวความลาดที่ลดระดับลง 1 m ต่อความยาว 1 km และจงหาอัตราการไหลในคลองส่งน้ำนี้

- **วิธีทำ (Python Code)**

```
#%% Optimization find R for Rectangle Shape
# When n=0.012, s=0.001, V=2.4cms
n = float(input('n = Manning Value = '))
s = float(input('s = Channel Slope = '))
V = float(input('V(m/s) = '))
from scipy import optimize
def RectangR(R):
    error = 1/n*R**(2/3)*(s**0.5) - V
    return error
sol = optimize.root_scalar(RectangR, bracket=[0,1],method='brentq')
R = sol.root
print('R = ',round(R,2),'m')
# เนื่องจากหน้าตัดที่ดีที่สุดของรางน้ำสี่เหลี่ยมผืนผ้าคือ b=2y จะได้ A=2y^2, P=4y
# ดังนั้นค่า R=A/P จะได้ค่า R=y/2 ซึ่งหมายความว่าค่า y=2R
y = 2*R
b = 2*y
A = 2*(y**2)
Q = A*V
print('y = ',round(y,2),'m')
print('b = ',round(b,2),'m')
print('Q = ',round(Q,2),'m\u00b3/s')
```

- **คำอธิบาย Python Code**

บรรทัดที่ 1 : คอมเมนต์ (comment) ใช้เพื่ออธิบายว่าบล็อกโค้ดถัดไปเป็นการคำนวณ “Optimization เพื่อหาค่า R สำหรับรางน้ำรูปสี่เหลี่ยม” Python จะ ไม่ประมวลผลบรรทัดนี้

บรรทัดที่ 2 : คอมเมนต์ (comment) ใช้เพื่ออธิบายว่าบล็อกโค้ดถัดไป “ค่า n=0.012, s=0.001 และ V=2.4m/s” Python จะ ไม่ประมวลผลบรรทัดนี้

บรรทัดที่ 3 : รับค่า Manning (n) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 4 : รับค่าความลาดชันของร่องน้ำ (s): โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 5 : รับค่าความเร็วการไหลของน้ำ (V) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม

บรรทัดที่ 6 : นำเข้าโมดูล optimize จาก Scipy เพื่อใช้ฟังก์ชันหา root

บรรทัดที่ 7-9 : สร้างฟังก์ชัน RectangR(R) โดยใช้สมการ Manning : $\frac{1}{n} R_2^{\frac{2}{3}} S^{\frac{1}{2}}$ และส่งกลับค่าความคลาดเคลื่อน (error)

บรรทัดที่ 10-11 : ใช้การหา root จากฟังก์ชัน RectangR และกำหนดช่วงของคำตอบ [0,1] จากนั้นใช้วิธี Brent's method ซึ่งคำตอบที่ได้อยู่ใน sol.root

บรรทัดที่ 12 : แสดงผลลัพธ์ค่ารัศมีไฮดรอลิก (R) โดยใช้ round (Q1,2) ปัดเศษให้เหลือทศนิยม 2 ตำแหน่ง หน่วยเป็น m

บรรทัดที่ 13-14 : อธิบายเกี่ยวกับหน้าตัดที่ดีที่สุดของรางน้ำสี่เหลี่ยมผืนผ้า

บรรทัดที่ 15 : จากการอธิบายจะได้ว่าค่าความลึกรางน้ำ y = 2R

บรรทัดที่ 16 : จากการอธิบายจะได้ว่าค่าความกว้างรางน้ำ b = 2y

บรรทัดที่ 17 : จากการอธิบายจะได้ว่าค่าพื้นที่หน้าตัดของรางน้ำ $A = 2y^2$

บรรทัดที่ 18 : คำนวณอัตราการไหล (Q) จากสมการอัตราการไหล ; Q = AV

บรรทัดที่ 19 : แสดงผลลัพธ์ค่าความลึกรางน้ำ (y) โดยใช้ round (y1,2) ปัดเศษให้เหลือทศนิยม 2 ตำแหน่ง หน่วยเป็น m

บรรทัดที่ 20 : แสดงผลลัพธ์ค่าความกว้างรางน้ำ (b) โดยใช้ round (b1,2) ปัดเศษให้เหลือทศนิยม 2 ตำแหน่ง หน่วยเป็น m

บรรทัดที่ 21 : แสดงผลลัพธ์ค่าอัตราการไหล (R) โดยใช้ round (Q1,2) ปัดเศษให้เหลือทศนิยม 2 ตำแหน่ง หน่วยเป็น m^3/s (cms)

- **<u>ผลลัพธ์</u>**
  - n = Manning Value = 0.012
  - s = Channel Slope = 0.001
  - V(m/s) = 2.4
  - R = 0.87 m

    ∴ y = 1.74 m, b = 3.48 m และ Q = 14.5 m^3/s

<u>ตอบ</u> - หน้าตัดดีที่สุดทางชลศาสตร์ของคลองส่งน้ำคอนกรีตรูปตัดสี่เหลี่ยมผืนผ้า คือ y = 1.74 m และ b = 3.48 m

- อัตราการไหลในคลองส่งน้ำนี้ (Q) = 14.5 m3/s

ข้อ 21. ทางน้ำเปิดรูปสี่เหลี่ยมคางหมูแห่งหนึ่งมีฐานกว้าง 3 m ความลาดด้านข้างในแนวดิ่งต่อแนวราบเท่ากับ 2 ต่อ 3 และสัมประสิทธิ์ความขรุขระ 0.0149 ในขณะที่มีอัตราการไหล 17 cms ไปตามแนวความลาดของทางน้ำเปิด 0.0003 วัดความลึกของกระแสน้ำด้านเหนือน้ำได้ 1.50 m และความลึกของกระแสน้ำด้านท้ายน้ำ 1.80 m จงหาระยะทางระหว่างจุดที่วัดความลึกทั้ง 2 จุด

- **วิธีทำ (Python Code)**

```
import math
Q = float(input('Q(cms) = '))
n = float(input('n = manning value = '))
b1 = float(input('b1(m) = '))
y1 = float(input('y1(m) = '))
b2 = float(input('b2(m) = '))
y2 = float(input('y2(m) = '))
z = float(input('z = Side slope of canal = '))
s = float(input('s = channel slope = '))
V1 = Q / ((b1 * y1) + (z *( y1**2)))
V2 = Q / ((b2 * y2) + (z *( y2**2)))
A1 = (b1 * y1) + (z *( y1**2))
A2 = (b2 * y2) + (z *( y2**2))
Vavg = (V1 + V2) / 2
P1 = b1 + (2 * y1 * math.sqrt(1 + z**2))
P2 = b2 + (2 * y2 * math.sqrt(1 + z**2))
R1 = A1 / P1
R2 = A2 / P2
Ravg = (R1 + R2) / 2

from scipy import optimize
def slopeSf(sf):
    error = 1/n * Ravg**(2/3) * (sf**0.5) - Vavg
    return error
sol = optimize.root_scalar(slopeSf, bracket=[0, 10], method='brentq')
sf = sol.root
print('sf = ', round(sf, 7))

DeltaL = abs(((y1 + ((V1**2) / (2 * 9.81))) - (y2 + ((V2**2) / (2 * 9.81)))) / (sf - s))
print('DeltaL = ', round(DeltaL, 2), 'm')
```

- **<u>คำอธิบาย Python Code</u>**

บรรทัดที่ 1 : นำเข้าโมดูล math เพื่อใช้ในการคำนวณ
บรรทัดที่ 2 : รับค่าอัตราการไหล (Q) เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 3 : รับค่า Manning (n) เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 4 : รับค่าความกว้างท้องน้ำจุดที่ 1 (b1) เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 5 : รับค่าความลึกของกระแสน้ำจุดที่ 1 (y1) เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 6 : รับค่าความกว้างท้องน้ำจุดที่ 2 (b2) เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 7 : รับค่าความลึกของกระแสน้ำจุดที่ 2 (y2) เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 8 : รับค่าความลาดของขอบท้องน้ำ (z) เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 9 : รับค่าความลาดของทางน้ำ (s) เป็นตัวเลขทศนิยมโดยให้ user ป้อนค่า
บรรทัดที่ 10 : คำนวณหาความเร็วการไหลของน้ำในจุดที่ 1 (V1)
บรรทัดที่ 11 : คำนวณหาความเร็วการไหลของน้ำในจุดที่ 2 (V2)
บรรทัดที่ 12 : คำนวณหาพื้นที่หน้าตัดในจุดที่ 1 (A1)
บรรทัดที่ 13 : คำนวณหาพื้นที่หน้าตัดในจุดที่ 2 (A2)
บรรทัดที่ 14 : คำนวณหาความเร็วการไหลเฉลี่ย (Vavg)
บรรทัดที่ 15 : คำนวณหาความยาวขอบเขตเปียกในจุดที่ 1 (P1)
บรรทัดที่ 16 : คำนวณหาความยาวขอบเขตเปียกในจุดที่ 2 (P2)
บรรทัดที่ 17 : คำนวณหารัศมีชลศาสตร์ในจุดที่ 1 (R1)
บรรทัดที่ 18 : คำนวณหารัศมีชลศาสตร์ในจุดที่ 2 (R2)
บรรทัดที่ 19 : คำนวณหารัศมีชลศาสตร์เฉลี่ย (Ravg)
บรรทัดที่ 21 : นำเข้าโมดูล optimize จากไลบรารี scipy เพื่อใช้ฟังก์ชันช่วยหาค่ารากสมการเชิงตัวเลข
บรรทัดที่ 22 : นิยามฟังก์ชันชื่อ slopeSf เพื่อคำนวณหาระยะทางระหว่างจุดที่วัดความลึกทั้ง 2 จุด(L)
บรรทัดที่ 23 : คำนวณค่าความคลาดเคลื่อน error โดยเปรียบเทียบความเร็วกับความเร็วเฉลี่ย
บรรทัดที่ 24 : ส่งค่า error กลับไปยังฟังก์ชันด้วย return
บรรทัดที่ 25 : ใช้ฟังก์ชัน optimize.root_scalar() เพื่อหาค่า Sf ที่ทำให้ฟังก์ชัน slopeSf มีค่าเป็นศูนย์โดยกำหนดช่วงการค้นหาอยู่ระหว่าง 0 ถึง 10 และใช้วิธี brentq
บรรทัดที่ 26 : ดึงค่ารากของสมการออกมาจะได้ค่า sf ที่โปรแกรมคำนวณได้
บรรทัดที่ 27 : แสดงผลลัพธ์ของ Sf ที่คำนวณได้ออกมาเป็นทศนิยม 7 ตำแหน่ง
บรรทัดที่ 29 : คำนวณระยะทาง L จากจุด 1 ถึงจุด 2
บรรทัดที่ 30 : แสดงผลลัพธ์ L ที่คำนวณได้

- **<u>ผลลัพธ์</u>**
  - Q(cms) = 17
  - n = manning value = 0.0149
  - b1(m) = 3
  - y1(m) = 1.5
  - b2(m) = 3
  - y2(m) = 1.8
  - z = Side slope of canal = 1.5
  - s = channel slope = 0.0003
  - sf = 0.0007986

∴ DeltaL = 405.93 m

<u>ตอบ</u> ระยะทางระหว่างจุดที่วัดความลึกทั้ง 2 จุด = 405.93 m

ข้อ 22. ทางน้ำเปิดรูปสี่เหลี่ยมผืนผ้าแห่งหนึ่งกว้าง 3 เมตร มีความลาดท้องน้ำ 0.002 และมีสัมประสิทธิ์ความขรุขระ 0.0149 ถ้าวัดความลึกด้านเหนือน้ำได้ 0.60 เมตร ในขณะที่จุดวัดความลึกด้านท้ายน้ำที่ห่างกับจุดวัดความลึกด้านเหนือน้ำ 60 เมตร ปรากฏว่าวัดความลึกได้ 0.67 เมตร จงหาอัตราการไหลในทางน้ำเปิดนี้

- **วิธีทำ (Python Code)**

```
#%% Determine Discharge (Q)
# When b = 3, s = 0.002, n = 0.0149, y1 = 0.60, y2  = 0.67, L = 60
# User Input
n = float(input('n = Manning Value = '))                #Manning n
s = float(input('s = Channel Slope = '))                #Bed Slope
L = float(input('L = Distance Between Section L = '))       #Distance (m)
b = float(input('b (m) = '))                            #Channel width (m)
y1 = float(input('y1 (m) = '))                          #Upstream depth (m)
y2 = float(input('y2 (m) = '))                          #Downstream depth (m)

# Calculate
A1 = b*y1
A2 = b*y2
P1 = b+(2*y1)                                       #P = b + 2y for rectangular channel
P2 = b+(2*y2)
R1 = A1/P1
R2 = A2/P2

from scipy import optimize
def Discharge(Q):
    V1 = Q/A1
    V2 = Q/A2
    Aavg = (A1+A2)/2
    Ravg = (R1+R2)/2
    Sfavg = (Q * n / (Aavg * Ravg**(2/3)))**2
    H1 = y1 + (V1**2) / (2 * 9.81)
    H2 = y2 + (V2**2) / (2 * 9.81)
    Delta_H = H1 - H2
    error = Delta_H - L * (Sfavg - s)
    return error
sol = optimize.root_scalar(Discharge, bracket = [0.01,10], method='brentq')
Q = sol.root
print('Discharge Q = ', round(Q,3), 'm\u00b3/s')
```

- **<u>คำอธิบาย Python Code</u>**

บรรทัดที่ 4 : รับค่า Manning (n) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 5 : รับค่าความลาดชันของร่องน้ำ (s) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 6 : รับค่าระยะห่างระหว่างหน้าตัด (L) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 7 : รับค่าความกว้างท้องน้ำ (b) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 8 : รับค่าความลึกด้านเหนือน้ำ (y1) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 9 : รับค่าความลึกด้านท้ายน้ำ (y2) โดยให้ User ป้อนค่า จากนั้นแปลงค่าที่ได้เป็นตัวเลขทศนิยม
บรรทัดที่ 12 : คำนวณพื้นที่หน้าตัด (A1) เมื่อความลึกด้านเหนือน้ำเท่ากับ y1
บรรทัดที่ 13 : คำนวณพื้นที่หน้าตัด (A2) เมื่อความลึกด้านท้ายน้ำเท่ากับ y2
บรรทัดที่ 14 : คำนวณความยาวขอบเขตเปียก (P1) เมื่อความลึกด้านเหนือน้ำเท่ากับ y1
บรรทัดที่ 15 : คำนวณความยาวขอบเขตเปียก (P2) เมื่อความลึกด้านท้ายน้ำเท่ากับ y2
บรรทัดที่ 16 : คำนวณรัศมีชลศาสตร์ (R1) เมื่อความลึกด้านเหนือน้ำเท่ากับ y1
บรรทัดที่ 17 : คำนวณรัศมีชลศาสตร์ (R2) เมื่อความลึกด้านท้ายน้ำเท่ากับ y2
บรรทัดที่ 19 : นำเข้าโมดูล optimize จากไลบรารี scipy เพื่อใช้ฟังก์ชันช่วยหาค่ารากสมการเชิงตัวเลข
บรรทัดที่ 20 : สร้างฟังก์ชันชื่อ Discharge(Q) เพื่อใช้คำนวณความคลาดเคลื่อนของสมการพลังงานจากค่าอัตราการไหล (Q)
บรรทัดที่ 21 : คำนวณความเร็วกระแสน้ำที่หน้าตัดต้นน้ำ (V1)
บรรทัดที่ 22 : คำนวณความเร็วกระแสน้ำที่หน้าตัดท้ายน้ำ (V2)
บรรทัดที่ 23 : คำนวณพื้นที่หน้าตัดเฉลี่ย (A_{avg}) จากค่าเฉลี่ยของ A1 และ A2
บรรทัดที่ 24 : คำนวณรัศมีไฮดรอลิกเฉลี่ย (R_{avg}) จากค่าเฉลี่ยของ R1 และ R2
บรรทัดที่ 25 : คำนวณค่าความลาดชันพลังงานเฉลี่ย (Sf_{avg}) จากสมการ Manning โดยใช้ค่า Q, n, A_{avg} และ R_{avg}
บรรทัดที่ 26 : คำนวณระดับพลังงานที่หน้าตัดต้นน้ำ H1 จากระดับน้ำ y1 และพลังงานจลน์
บรรทัดที่ 27 : คำนวณระดับพลังงานที่หน้าตัดท้ายน้ำ H2 จากระดับน้ำ y2 และพลังงานจลน์
บรรทัดที่ 28 : คำนวณผลต่างพลังงานระหว่างต้นน้ำและท้ายน้ำ
บรรทัดที่ 29 : คำนวณค่าความคลาดเคลื่อน error โดยเปรียบเทียบผลต่างพลังงานกับการสูญเสียพลังงาน
บรรทัดที่ 30 : ส่งค่าความคลาดเคลื่อน error กลับไปยังโปรแกรมหลักด้วยคำสั่ง return
บรรทัดที่ 31 : ใช้ฟังก์ชัน optimize.root_scalar() เพื่อหาค่า Q ที่ทำให้ฟังก์ชัน Discharge(Q) มีค่าเป็นศูนย์ โดยกำหนดช่วงการค้นหาอยู่ระหว่าง 0.01 ถึง 10 และใช้วิธี brentq
บรรทัดที่ 32 : เก็บค่ารากของสมการ (ค่า Q ที่หาได้) ไว้ในตัวแปร Q
บรรทัดที่ 33 : แสดงผลลัพธ์อัตราการไหลที่คำนวณได้ โดยปัดเศษให้เหลือทศนิยม 3 ตำแหน่ง และแสดงหน่วยเป็น m^3/s

- **<u>ผลลัพธ์</u>**
  - n = Manning Value = 0.0149
  - s = Channel Slope = 0.002
  - L = Distance Between Section L = 60
  - b (m) = 3
  - y1 (m) = 0.6
  - y2 (m) = 0.67

∴ Discharge Q = 2.555 m^3/s

<u>ตอบ</u> อัตราการไหลในทางน้ำเปิดนี้ (Q) = 2.555 m^3/s


ROYAL IRRIGATION DEPARTMENT