



**การจำแนกป่าไม้โดยใช้ข้อมูล Time Series ในการสร้าง Phenotype จาก
ข้อมูลภาพถ่ายดาวเทียมด้วยโมเดล Deep Learning**

**Forest classification using time-series satellite imagery to construct
phenotypic signatures using Deep Learning Model**

โดย

**นาย ปณณวิชญ์ โคโตศรี รหัสนิสิต 6714500645
นิสิตปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเกษตรศาสตร์ กรุงเทพมหานคร**

Abstract

โครงการนี้นำเสนอแนวทางการจำแนกพื้นที่ป่าไม้จากข้อมูลดาวเทียม Sentinel-2 โดยใช้ข้อมูลอนุกรมเวลาของดัชนีพืชพรรณและดัชนีน้ำ ได้แก่ SAVI และ MNDWI รายเดือนเป็นเวลา 12 เดือน เพื่อจับสัญญาณเชิงฤดูกาล (vegetation phenology) ของพื้นที่ป่าเทียบกับพื้นที่อื่น ๆ ข้อมูลถูกนำเข้าโมเดล Deep Learning 3 รูปแบบ ได้แก่ LSTM, Bidirectional LSTM with Attention และ Transformer Encoder และประเมินด้วย Accuracy Precision Recall F1-Score และ AUC ผลลัพธ์พบว่าโมเดลทั้งสามมีความแม่นยำในระดับใกล้เคียงกัน โดย LSTM แบบมาตรฐานให้ผลดีที่สุดภายใต้ทรัพยากรที่ใช้ ทำให้เห็นว่าข้อมูลอนุกรมเวลาเพียงสองตัวชี้วัดก็สามารถสร้างโมเดลที่มีประสิทธิภาพสูงได้

Introduction

ประเทศไทยเป็นประเทศที่มีพื้นที่ป่าขนาดใหญ่ แต่ประสบปัญหาการบุกรุกพื้นที่ป่าและการเปลี่ยนแปลงการใช้ประโยชน์ที่ดินอย่างต่อเนื่อง การตรวจสอบสถานะป่าไม้แบบดั้งเดิม เช่น การสำรวจภาคสนามและการอ่านภาพถ่ายทางอากาศ มีข้อจำกัดด้านต้นทุน แรงงาน และความรวดเร็ว

ภาพถ่ายจากดาวเทียม เช่น Sentinel-2 ทำให้สามารถติดตามการเปลี่ยนแปลงของพื้นที่ได้อย่างต่อเนื่อง และเมื่อประกอบกับวิธี Deep Learning ช่วยเพิ่มศักยภาพในการแยกแยะประเภทพื้นที่โดยอัตโนมัติจากลักษณะอนุกรมเวลา

ปัญหาที่พบ

ป่าไม้มีพฤติกรรมเชิงฤดูกาลที่ซับซ้อน เช่น ใบไม้และความชื้นเปลี่ยนแปลงไปตามฤดู ทำให้ค่าดัชนีดาวเทียม (SAVI, MNDWI) ไม่คงที่ การใช้ภาพเพียงช่วงเวลาเดียวจึงเกิดความคลาดเคลื่อน เช่น พื้นที่ป่าหน้าแล้งอาจถูกตีความเป็นพื้นที่เสื่อมโทรม ขณะเดียวกันพื้นที่เกษตรฤดูฝนที่มีความเขียวสูงอาจถูกเข้าใจผิดเป็นป่า นอกจากนี้พื้นที่ป่ามีสัดส่วนต่ำกว่าพื้นที่อื่น ทำให้เกิดปัญหา Data Imbalance หากไม่จัดการ โมเดลจะค่อนข้างทำนาย Other แล้วให้ Accuracy สูงแต่ใช้งานจริงไม่ได้ ดังนั้นงานนี้จึงใช้ข้อมูลอนุกรมเวลา 12 เดือนและเทคนิค Balanced Sampling เพื่อแก้สองปัญหาหลัก และใช้โมเดล Deep Learning เชิงเวลา เช่น LSTM และ Transformer ในการจับ seasonal signature ของป่าอย่างมีประสิทธิภาพ

เหตุการณ์	ผลกระทบการจำแนก
หน้าแล้ง ใบไม้ร่วง	พื้นที่ป่าถูกเข้าใจผิดเป็นพื้นที่โล่ง / เกษตร
เกษตรฤดูฝนเขียวจัด	พื้นที่เกษตรถูกเข้าใจผิดเป็นป่า
แหล่งน้ำแห้ง/ขุ่น บางช่วงฤดู	ค่า MNDWI ผิดเพี้ยน

ดังนั้น ข้อมูลภาพแบบรายเดือนหรือรายปีจึงจำเป็น เพื่อให้โมเดลเข้าใจ ลักษณะตามฤดูกาลของป่า (Seasonal Forest Phenology)

การจำแนกป่าไม้จากข้อมูลระยะไกลต้องเผชิญกับประเด็นสำคัญสองข้อหลักๆ ได้แก่:

1. ลักษณะเชิงฤดูกาลซ้ำซ้อน (Seasonality)
พืชในป่ามีการเปลี่ยนแปลงค่า SAVI และ MNDWI ตามฤดูกาล เช่น
 - ฤดูฝน ความเขียวสูง (SAVI) ความชื้นสูง (MNDWI)
 - หน้าแล้ง SAVI และ MNDWI ลดลง
2. Data Imbalance
จำนวนพิกเซลที่เป็น forest น้อยกว่าพื้นที่อื่นมาก โมเดลสุ่มทำนาย other แล้ว Accuracy สูงได้
แก้ด้วย Balanced Sampling

วัตถุประสงค์

1. สร้างโมเดล Deep Learning เพื่อจำแนกป่าไม้จาก time-series ของ SAVI และ MNDWI
2. เปรียบเทียบโมเดล 3 รูปแบบ LSTM, BiLSTM+Attention, Transformer Encoder
3. ประเมินประสิทธิภาพและวิเคราะห์ความเหมาะสมเชิงปฏิบัติ

ข้อมูลและการเตรียมข้อมูล (Dataset & Preprocessing)

หมวดหมู่	รายละเอียด
แหล่งข้อมูล (Data Source)	<div><div>• Sentinel-2 Level-2A (Sentinel-2)</div><div>• LULC Map (Land-Use Land-Cover Thailand)</div></div>
Feature / Index	<div><div>• SAVI (Soil Adjusted Vegetation Index)</div><div>• MNDWI (Modified Normalized Difference Water Index)</div></div>
Spatial Resolution	20 m
พื้นที่ทดลอง	จังหวัดตาก

Methodology

1) ข้อมูลและการเตรียมข้อมูล (Data & Preprocessing)

1.1 แหล่งข้อมูลและ Index

- ภาพ Sentinel-2 Level-2A แปลงเป็นดัชนีรายเดือน:
 - SAVI (vegetation/greenness)
 - MNDWI (surface water/moisture)
- LULC raster ใช้เป็น label โดยกำหนด Forest = 1 เมื่อ LULC > 6000 และ Other = 0 เพื่อใช้ในการกรองข้อมูลดิบก่อน

1.2 การโหลดและซิงค์อนุกรมเวลา

- โหลดสแตกรายเดือนจากโฟลเดอร์ MAX_SAVI และ MAX_MNDWI
- กรอง NoData ด้วยเงื่อนไข $a \leq -9000$ คือให้เป็น NaN
- ซิงค์เดือน ระหว่าง SAVI/MNDWI ด้วย common_months เพื่อให้ input ตรงกันทุกแบนด์ทุกเดือน

1.3 การ Mask Data

- ใช้ Mask $lulc > 0$ เพื่อเอาเฉพาะ pixel ที่มีความหมาย
- สร้าง label $y = 1(lulc > 6000)$, $y \in \{0,1\}$
- กรองตัวอย่างที่ยังมี NaN ด้วย `good = np.all(np.isfinite(X), axis=1)`

1.4 จัดการ Data Imbalance

- Balanced undersampling: สุ่มให้จำนวน Forest เท่ากับ Other (1:1) เพื่อกันโมเดลลำเอียง
 - เลือกใช้ undersampling เพราะฝั่ง Other มักมีจำนวนมากช่วยให้ train เร็วและเข้าใจง่าย

1.5 รูปร่างอินพุตและการสเกล

- reshape เป็น $X \in \mathbb{R}^{N \times T \times 2}$ (N = ตัวอย่าง, T = เดือน, 2 features)
- Z-score per feature (รวมทุกเดือน):

$$X = \frac{x_i - \mu}{\sigma}$$

โดย μ, σ จำนวนบน train set (ในโค้ดใช้จากทั้ง x_i ที่สุ่มแล้ว ให้จำนวนบน train เท่านั้น)

1.6 Train/Validation Split

- train_test_split แบบ stratified 80/20
- หมายเหตุสำหรับงาน Spatial Data: หากเป็นการประเมินแบบข้ามงวด ควร spatial block split (แยกพื้นที่) ลด spatial leakage

2) สถาปัตยกรรมโมเดล (Model Architectures)

ทดสอบ Train 3 model : LSTM, BiLSTM + Attention, Transformer Encoder

ทุกโมเดลจบด้วย linear 2 logits แล้วใช้ CrossEntropyLoss (มี softmax ภายใน)

2.1 LSTM Classifier (Baseline)

- โครงสร้าง: LSTM(input_size=2, hidden_size=64, num_layers=3, dropout=0.25, batch_first=True)
- ใช้ last hidden state ของเลเยอร์บนสุด $h[-1] \in \mathbb{R}^{64}$ Linear(64 ไปที่ 2)
- จุดแข็ง: เรียบง่าย คำนวณพารามิเตอร์ไม่บวม เหมาะกับ series สั้น ($T \approx 12$)

พารามิเตอร์หลักที่เลือก

- hidden_size=64: พอสำหรับ pattern ฤดูกาลของ 2 ฟีเจอร์ 12 เดือน (ทดลองแล้วใหญ่กว่านี้ไม่คุ้มค่าในการคำนวณ)
- num_layers=3: ให้ capacity พอเรียนรู้ nonlinearity หลายชั้น โดยมี dropout = 0.25 กัน overfit
- เหตุผลใช้ AdamW (wd=1e-4): ควบคุม norm ของ weight ดีกว่า Adam ปกติในงาน sequence เล็ก

2.2 BiLSTM + Attention

- Bi-directional LSTM ช่วยอ่านสัญญาณจากอดีตและอนาคตในหน้าต่าง 12 เดือน
- Additive attention:

- คำนวณน้ำหนัก $\alpha_t = \text{softmax}(W_a h_t)$

- สร้าง context $C = \sum_t \alpha_t h_t$
- Linear(2*hid = 2) (เพราะ BiLSTM concat = 128)

เหตุผลที่เลือก

- เมื่อ series สั้น การมองสองทิศและ attention อาจช่วยโฟกัสเดือนสำคัญ (เช่น peak ฝน/แล้ง)
- ถ้า T ยาวกว่านี้มาก Attention จะเด่นชัดขึ้น

2.3 Transformer Encoder

- proj: Linear(2 = d_model), Sinusoidal Positional Encoding, nlayer=3, nhead=8, dim_ff=128
- Global average pooling ตลอดลำดับ = Linear(d_model=2)

เหตุผลที่เลือก

- d_model=64, nhead=8 (head size 8) สอดคล้องกับขนาดฟิเจอร์เล็กและ T สั้น
- dim_ff=128 ให้ capacity พอโดยไม่หนักเกิน

3) การ Optimization

3.1 Loss & Metrics

- **Loss:** CrossEntropyLoss() (2-logits เป็น softmax)
- **Metrics:** Accuracy, Precision, Recall, F1, ROC-AUC
 - เน้น Recall(Forest) เพื่อลด miss ป่า
 - รายงาน F1 เพื่อสมดุล precision/recall

3.2 Optimizer, Scheduler, Early Stopping

- AdamW lr เริ่มต้น: LSTM/BiLSTM = 1e-3, Transformer = 8e-4
- Weight decay = 1e-4 ลด overfit
- ReduceLROnPlateau (monitor val_loss, factor=0.5, patience=3) ถ้อย ๆ ลด lr เมื่อไม่ดีขึ้น
- Early stopping (patience=8-10) โหลด best checkpoint ตาม val_loss

3.3 Batch/Thread/Seed

- batch_size=512 (series สั้น จึงใช้ batch ใหญ่เพื่อความเสถียร)
- กลุ่มเธรด: OMP_NUM_THREADS, torch.set_num_threads, set_num_interop_threads
- Seed: torch, numpy, random เพื่อ reproducibility

4) ขั้นตอนการฝึกและประเมิน (Training & Evaluation Pipeline)

4.1 Train loop

- แต่ละ epoch: train หนึ่งชุดและบันทึกค่า train_loss
- วัด val_loss โดย step scheduler
- เก็บ model_best.pt เมื่อ val_loss ต่ำสุด
- Early stop เมื่อไม่ดีขึ้นติดต่อกัน \geq patience

4.2 หลัง Train เสร็จ

- โหลด best weight
- รวบรวม $y_prob = \text{softmax}(\text{logits})[:,1]$ และ $y_pred = (y_prob > 0.5)$
- คำนวณ acc precision recall f1 และ auc
- สร้าง Confusion Matrix ROC curve และ Loss curve
- บันทึก metrics.csv, loss_curve.png, confusion_matrix.png, roc_curve.png, model.pt

การเลือกพารามิเตอร์เริ่มต้น (Default Choices & Rationale)

กลุ่ม	ค่าที่ใช้	เหตุผล
LSTM hidden	64	เพียงพอสำหรับ 2 พียูเจอร์, $T \approx 12$
LSTM layers	3	เพิ่ม capacity เล็กน้อย + dropout
Dropout	0.25	กัน overfit จาก series สั้น
BiLSTM	64×2	มองสองทิศทาง เพิ่มบริบทใน T สั้น
Attention	additive (Linear ไป softmax)	เบา เร็ว ดีความได้
Transformer d_model	64	สมดุลพารามิเตอร์/ความเร็ว
Heads	8	head size 8 พอดี
FF dim	128	ใหญ่พอไม่ over-param
Epochs	50–60	พอให้ converge, มี early stop
LR	1e-3 / 8e-4	ค่ามาตรฐานสำหรับ AdamW งานนี้
WD	1e-4	ลด overfit, คงความเสถียร
Batch	512	เสถียร, ใช้หน่วยความจำเหมาะสม

6) Tune Hyperparameters (Practical Tuning Plan)

เฟส 0: ตรวจสอบการไหลของข้อมูล

- เช็คซิงค์เดือน, จำนวนตัวอย่างหลัง mask, สัดส่วนคลาส, NaN, สเตทพียูเจอร์, ค่า metrics เบื้องต้น

เฟส 1: Coarse Random Search (แนะนำ 20–30 trials)

- LSTM: hidden $\in \{32, 64, 96\}$, layers $\in \{1, 2, 3\}$, dropout $\in \{0.1, 0.25, 0.4\}$

- Transformer: $d_model \in \{48, 64, 96\}$, $nhead \in \{4, 8\}$, $nlayer \in \{2, 3, 4\}$, $dim_ff \in \{128, 192, 256\}$
- Optimizer: $lr \in \{5e-4, 1e-3, 2e-3\}$, $wd \in \{0, 1e-4, 5e-4\}$
- Batch: $\{256, 512, 1024\}$ (ขึ้นกับ RAM/GPU)
- เกณฑ์: val F1 (Forest) หรือ macro-F1 + ดู ROC-AUC

เฟส 2: Fine Grid/BO

- โฟกัสรอบค่าที่ดีจากเฟส 1
- เพิ่ม trial สำหรับ threshold tuning บน val set (maximize F1/Recall)

เฟส 3: Robustness & Leakage Checks

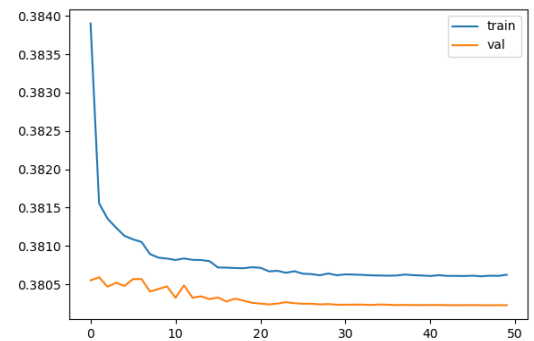
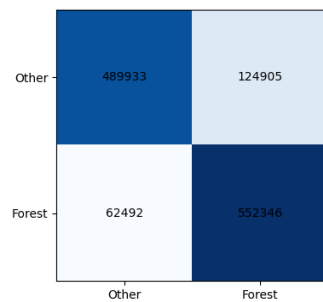
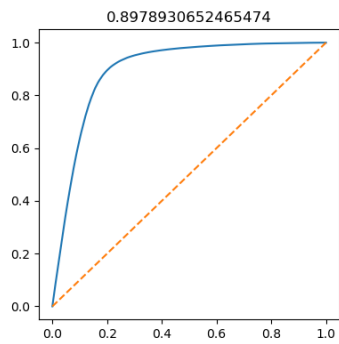
- เปลี่ยน random seed 3-5 ชุด โดยใช้ $median \pm IQR$ ของ metrics
- ถ้าเป็นพื้นที่จริง ใช้ spatial block split และทดสอบ cross-region

Result

ตารางแสดงผลลัพธ์

Model	Acc	Precision	Recall	F1	AUC
LSTM	0.8476	0.8154	0.8985	0.8550	0.8979
BiLSTM+Att	0.8476	0.8156	0.8984	0.85497	0.8979
Transformer	0.8476	0.8159	0.8977	0.85486	0.8979

BiLSTM+Att



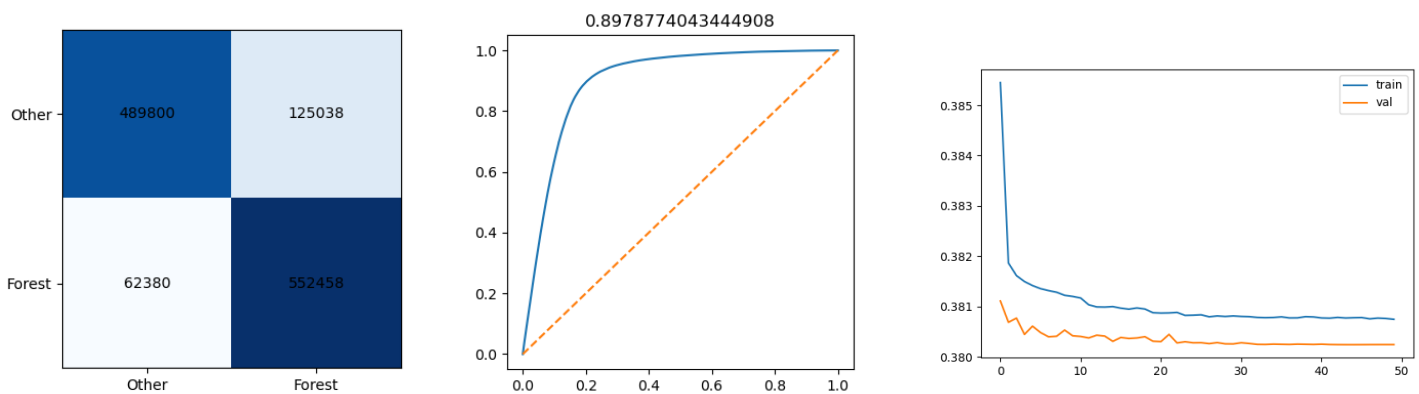
BiLSTM + Attention

Metric	Value
Accuracy	0.8476
Precision	0.8156
Recall	0.8984

Metric	Value
F1-Score	0.8550
AUC	0.8979

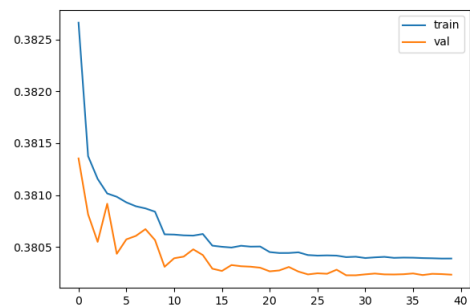
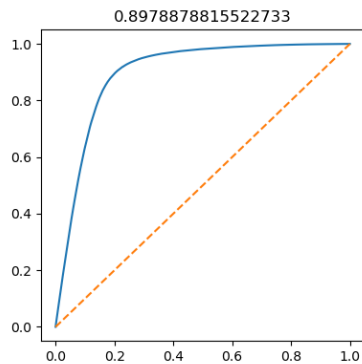
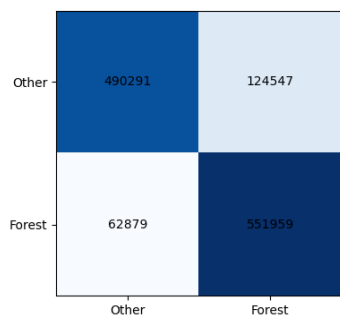
สรุป: จำแนกป่าได้ดีมาก (Recall สูงสุด) เหมาะกับงานพื้นที่ป่าที่เน้นไม่ให้พลาดป่าจริง เช่น อาจจะเป็นการทำเขตแดน หรือ ปักปัน อะไรคร่าวๆ

LSTM



Metric	Value
Accuracy	0.8476
Precision	0.8154
Recall	0.8985
F1-Score	0.8550
AUC	0.8979

สรุป: สมดุลดี ไกล่เคียง BiLSTM-Att มาก แต่ไม่มี Attention ความสามารถจับฤดูกาลอาจน้อยกว่าเล็กน้อย



Transformer

Metric	Value
Accuracy	0.8476
Precision	0.8159
Recall	0.8977
F1-Score	0.8549
AUC	0.8979

สรุป: ทำงานดีใกล้เคียงกันแต่ Precision สูงสุด ลดการทำนายป่าผิดเป็นพื้นที่อื่น

สรุปเทียบโมเดล

Model	จุดเด่น
BiLSTM-Attention	Recall สูงสุด เก่งมากในการตรวจจับป่าจริง
LSTM	baseline แข็งแกร่ง ใกล้เคียง BiLSTM
Transformer	Precision สูงสุด ลด false forest

ทั้งสามโมเดลให้ผลใกล้เคียงกันมาก (~0.848 accuracy และ ~0.898 AUC)

ข้อเสนอแนะเลือกโมเดลตามเป้าหมาย

เป้าหมาย	โมเดลแนะนำ
ลดพลาดป่าจริง (Conservation / Monitoring)	BiLSTM-Attention
ต้องการความเสถียรและง่าย	LSTM
ลด false alarm ป่าปลอม	Transformer

ที่มาของวิธีการต่างๆ และ ข้อมูล

แนวคิดของงานนี้มีพื้นฐานมาจากงานวิจัยระดับปริญญาโทของผม ซึ่งศึกษาแนวคิด Phenotype-based vegetation analysis หรือการวิเคราะห์ลักษณะเชิงลักษณะ-เชิงสัญญาณของพืช ผ่านข้อมูลดาวเทียมแบบอนุกรมเวลา (Time-Series Remote Sensing). โดยหลักการคือการทำความเข้าใจรูปแบบการเปลี่ยนแปลงของพืชตลอดช่วงฤดูกาลหนึ่งปี (Seasonal Cycle) เช่น การเติบโต ช่วงใบหนาแน่น ช่วงคายน้ำ หรือช่วงพักตัว ซึ่งสะท้อนออกมาในรูปของสัญญาณดัชนีพืชพรรณในแต่ละเดือนได้ ในงานนี้ ผมประยุกต์แนวคิดดังกล่าวกับข้อมูลป่าไม้ (Forest Phenology) เพื่อศึกษาพฤติกรรมการเปลี่ยนแปลงตามฤดูกาลของต้นไม้ และนำข้อมูลเชิงสัญญาณนี้มาใช้ฝึกโมเดล Deep Learning แบบ Long-Short Term (เช่น LSTM, BiLSTM-Attention และ Transformer) เพื่อจำแนกพื้นที่ป่ากับพื้นที่อื่นอย่างแม่นยำ จุดประสงค์หลักคือการสำรวจว่า โมเดลลำดับเวลา (Sequential Deep Learning) เหล่านี้สามารถเรียนรู้ลักษณะเชิงฤดูกาลของพืชในธรรมชาติได้ดีเพียงใด และสามารถนำแนวคิด Phenology/Phenotype ที่ผมพัฒนามาในงานวิจัยเดิม ไปประยุกต์ให้สอดคล้องกับโครงสร้างโมเดล Deep Learning สมัยใหม่ได้อย่างไร นี่จึงเป็นที่มาของโปรเจกต์นี้