Training Guide — CICIDS2017 Cyber Attack Classification

คู่มือเทรนโมเดล AI สำหรับ proposal โครงงาน เป้าหมาย: เทรน 3 โมเดล (RandomForest, XGBoost, LightGBM) บน CICIDS2017 ครบ 10 classes ด้วย accuracy + correctness สูงสุดเท่าที RAM อนุญาต

1. ตรวจเครื่อง

ต้องการ:

- Python 3.10 หรือใหม่กว่า
- RAM อย่างน้อย 8 GB (16-32 GB ดีกว่า — full corpus train ใน 30-45 นาที)
- พื้นที่ดิสก์ว่าง ~3 GB (dataset 880 MB + cache + artefacts)
- Windows / macOS / Linux ใช้ได้หมด (คู่มือนี้แสดงคำสั่ง PowerShell)

เช็ค Python:

```
python --version
```

ถ้าไม่ใช่ 3.10+ ดาวน์โหลดที่ python.org

2. Setup ครั้งเดียว

2.1 Clone repo

```
git clone <YOUR_GITHUB_URL>
cd cyber_attack_classification
```

2.2 สร้าง virtual environment + ติดตั้ง dependencies

```
python -m venv .venv
.\.venv\Scripts\Activate.ps1
pip install -r requirements.txt
```

ขั้นนี้ใช้เวลา ~3-5 นาที (ดาวน์โหลด ~500 MB)

2.3 ดาวน์โหลด dataset CICIDS2017

1. ไปที่ https://www.unb.ca/cic/datasets/ids-2017.html
2. กด "Download this dataset" → กรอกชื่อ + อีเมล + เหตุผล → รอประมาณ 15-30 นาที (กดอีเมลตอบกลับมี link)
3. ดาวน์โหลด `MachineLearningCSV.zip` (~230 MB)
4. แตก zip → ใน folder มี CSV 8 ไฟล์
5. ก็อปทั้ง 8 ไฟล์ใส่ `data/raw/` ของ repo

ตรวจว่าครบ:

```
ls data\raw\*.csv
```

ต้องเห็น 8 ไฟล์:

- Monday-WorkingHours.pcap_ISCX.csv
- Tuesday-WorkingHours.pcap_ISCX.csv

- Wednesday-workingHours.pcap_ISCX.csv
- Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv
- Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX.csv
- Friday-WorkingHours-Morning.pcap_ISCX.csv
- Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv
- Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv

3. เลือก RAM preset (ไม่ต้องแก้โค้ด)

`train.py` มี flag `--preset` ให้ปรับขนาด train ตาม RAM โดยไม่ต้องแก้ CONFIG เอง

Preset	subsample	HP search subset	n_iter	สำหรับ
`8gb` (default)	300k rows	80k	8	RAM 8 GB
`16gb`	800k rows	150k	12	RAM 16 GB
`32gb`	full ~2.5M	200k	20	RAM 32 GB ขึ้นไป

ใช้แบบนี้:

```
.\.venv\Scripts\python.exe -W error::Warning train.py --preset 32gb
```

ถ้าไม่ใส่ `--preset` = ใช้ default ของ CONFIG (8gb เทียบเท่า)

ถ้าอยากเปลี่ยนค่าเฉพาะอันใดอันหนึ่ง — เปิด `train.py` แก้ `RAM_PRESETS` (บรรทัด ~140) แทน

4. ทดสอบโค้ดก่อน (smoke test, ~2 นาที)

```
.\.venv\Scripts\python.exe -W error::Warning train.py --smoke
```

ตรวจทีบรรทัดสุดท้าย:

```
HH:MM:SS | INFO    | Done in XXX.Xs. All artefacts under ...\results\smoke
```

ถ้าเห็นบรรทัดนี + exit code 0 = ผ่าน

ถ้าเด้ง error/warning ใดๆ — แปลว่า environment ไม่พร้อม แก้ตามตาราง troubleshooting ด้านล่างก่อน

5. Train จริง

เลือก preset ตาม RAM:

RAM 8 GB (default — ไม่ต้องใส่ flag):

```
.\.venv\Scripts\python.exe -W error::Warning train.py
```

RAM 16 GB:

```
.\.venv\Scripts\python.exe -W error::Warning train.py --preset 16gb
```

RAM 32 GB ขึ้นไป (full corpus — accuracy ดีสุด):

```
.\.venv\Scripts\python.exe -W error::Warning train.py --preset 32gb
```

`-W error::Warning` แปลว่า warning ใดๆ จะถือเป็น error ทำให้สคริปต์หยุดทันที — ใช้พิสูจน์ว่าโค้ดเดิน clean จริง

เวลาทีใช้ (ประมาณ):

RAM	Subsample	เวลา
8 GB	300k	30-45 min
16 GB	800k	60-90 min
32 GB	full (2.5M)	90-120 min

ระหว่าง train ไม่ต้องปิดเครื่อง — ระวังเรื่อง laptop ร้อนเกิน

ถ้าหยุดกลางทาง: สคริปต์มี resumable — รันใหม่จะข้าม model ที train เสร็จแล้ว ไม่ต้องเริ่มจากศูนย์

6. ดูผลลัพธ์

หลังเสร็จ artefacts อยู่ที่ `results/latest/` ประกอบด้วย:

ไฟล์	คืออะไร
`random_forest.joblib`	โมเดล RandomForest พร้อม scaler
`random_forest_per_class.csv`	per-class precision / recall / f1
`random_forest_confusion_matrix.png`	รูป confusion matrix
`random_forest_metrics.json`	metric ทั้งหมดของ RF
`xgboost.*`	ชุดเดียวกัน 4 ไฟล์ สำหรับ XGBoost
`lightgbm.*`	ชุดเดียวกัน 4 ไฟล์ สำหรับ LightGBM
`label_encoder.joblib`	ตัวแปลง label string ↔ int
`feature_columns.json`	ลำดับ 77 features (ใช้ตอน inference)
`metrics.json`	สรุปรวมทุก model
`report.md`	รายงานหลัก อ่านได้เลย

อ่าน report.md

VS Code: เปิด `results/latest/report.md` แล้วกด Ctrl+Shift+V เพื่อ preview

ต้องเห็น:

1. Per-class sample sizes — Heartbleed มี train+test ครบ (8+3), Infiltration (29+7)
2. Headline metrics — ทุก model ควรได้ accuracy >0.99, f1_macro >0.85
3. Verdict — ตรวจ shuffled-labels f1_macro ใกล้ chance level (ถ้าใช่ = ไม่มี leakage)

7. Trust checks ที่ต้องเช็คก่อน defend อาจารย์

7.1 Shuffled-labels test (anti-leakage)

ใน `report.md` ดูคอลัมน์ `shuffled-labels f1_macro` ต้องใกล้ `1 / n_classes`

- 10 classes → chance = 0.10 → shuffle score ควร 0.08-0.12
- ถ้า shuffle score สูงกว่านี้มาก = pipeline leak

7.2 CV std

`CV f1_macro (mean +/- std)` — std ควร <0.02

- ถ้า std >0.05 = ผลแกว่ง อาจเป็น single-lucky-split

7.3 Majority baseline lift

`accuracy - majority_baseline_acc` ควร >+0.30

- ถ้า lift น้อย = model ไม่ได้เรียนรู้อะไรเกินกว่าทาย majority

8. หลัง train เสร็จ — เตรียม dashboard

ดูคู่มือเสริม `dashboard_guide.pdf`

สั้นๆ: ก็อปโมเดลไป canonical path

```
Copy-Item results\latest\random_forest.joblib models\
Copy-Item results\latest\xgboost.joblib models\
Copy-Item results\latest\lightgbm.joblib models\
Copy-Item results\latest\label_encoder.joblib models\
```

แล้วเปิด dashboard ตามคู่มือ

9. Troubleshooting

อาการ	สาเหตุ	แก้
FileNotFoundError data/raw	ลืมเอา CSV ใส่	ดูข้อ 2.3
MemoryError ตอน load	RAM ไม่พอ	ลด subsample_n
-W error เด้ง UserWarning	library version ไม่ตรง	pip install -r requirements.txt --upgrade
LightGBM โผล่ "feature_name" warning	venv เก่า, sklearn <1.3	pip install scikit-learn>=1.3 --upgrade
Train ช้าผิดปกติ	CPU โดน throttle ความร้อน	ลด hp_search_n_iter, ปิด background apps
Heartbleed ไม่อยู่ใน final distribution	smoke mode subsample เล็กไป	ใช้ full mode (ไม่ใส่ --smoke)
ModuleNotFoundError	ยังไม่ activate venv	.\\.venv\\Scripts\\Activate.ps1

10. คำสั่งสรุป

Setup ครั้งเดียว (วาง 8 CSV ใน `data\raw\` ก่อน):

```
python -m venv .venv
.\.venv\Scripts\Activate.ps1
pip install -r requirements.txt
```

ทดสอบ environment (~2 นาที):

```
.\.venv\Scripts\python.exe -W error::Warning train.py --smoke
```

Train จริง (เปลี่ยน `--preset 32gb` ตาม RAM เครื่อง):

```
.\.venv\Scripts\python.exe -W error::Warning train.py --preset 32gb
```

ดูผล:

```
code results\latest\report.md
```