

- 📁 **สรุปการนำเสนอระบบ AI ตรวจวัดคุณภาพอากาศ**
 - Air Quality AI Monitoring System - Comprehensive Presentation Summary
 - 📄 สารบัญ (Table of Contents)
 - ภาพรวมระบบ
 - 🎯 วัดผลกระทบ
 - 🏗️ สถาปัตยกรรมระบบ
 - 📊 ความสามารถหลัก
 - Hardware Diagram
 - 🔌 แผนผังการเชื่อมต่อพร้อมหมายเลข Pin
 - 📋 ตารางการเชื่อมต่อ Pin
 - ⚡ การจ่ายไฟ
 - Flowchart ระบบ
 - 🔄 ขั้นตอนการทำงานของระบบ
 - 📊 Data Flow Diagram
 - AI Model และการ Training
 - 🤖 การเลือกโมเดล AI แบบอัจฉริยะ
 - 🧠 LSTM Neural Network (ข้อมูลขนาดใหญ่)
 - สถาปัตยกรรมโมเดล:
 - คุณสมบัติ LSTM:
 - 🌳 Random Forest (ข้อมูลขนาดเล็ก)
 - พารามิเตอร์โมเดล:
 - ข้อดี Random Forest:
 - 🕒 ช่วงการพยากรณ์
 - 🔄 การฝึกโมเดลอัตโนมัติ
 - การจัดการ Dataset
 - 📊 โครงสร้างข้อมูล
 - Raw Data Format (air_quality_data.csv):
 - Prediction Data Format (predictions.csv):
 - 🔄 Data Pipeline
 - 📈 Data Preprocessing Steps
 - 💾 Data Storage Strategy
 - 🛠️ การจัดการข้อมูลอัตโนมัติ
 - เหตุผลในการเลือกใช้ AI
 - 🎯 ปัญหาที่ AI ช่วยแก้ไข
 - 🔬 เปรียบเทียบวิธีการ
 - 🎯 เหตุผลเลือก Hybrid Approach
 - 📊 ผลประโยชน์ที่ได้รับ

- 🚀 การปรับแต่งสำหรับ Jetson Nano
- ผลลัพธ์และการประเมิน
 - 📊 เมตริกการประเมิน
 - 1. Mean Absolute Error (MAE)
 - 2. Root Mean Square Error (RMSE)
 - 3. Accuracy by Time Horizon
 - 🎯 ผลการทดสอบ
 - LSTM vs Random Forest Performance:
 - 📈 Real-time Dashboard Features
 - 🌐 System Performance
 - Hardware Utilization (Jetson Nano):
 - Response Times:
 - 🎉 Key Achievements
 - 🖥️ Web Dashboard Features
 - การออกแบบทันสมัย:
 - ส่วนหลักของเว็บไซต์:
- 🎯 สรุป
 - ระบบ Air Quality AI ที่พัฒนาขึ้นประกอบด้วย:
 - ผลลัพธ์ที่ได้:
 - การประยุกต์ใช้:
 - จุดเด่นของระบบ:
- 🚀 Quick Start Commands
 - 🌐 เข้าถึงเว็บไซต์ที่:

สรุปการนำเสนอระบบ AI ตรวจวัดคุณภาพอากาศ

Air Quality AI Monitoring System - Comprehensive Presentation Summary



สารบัญ (Table of Contents)

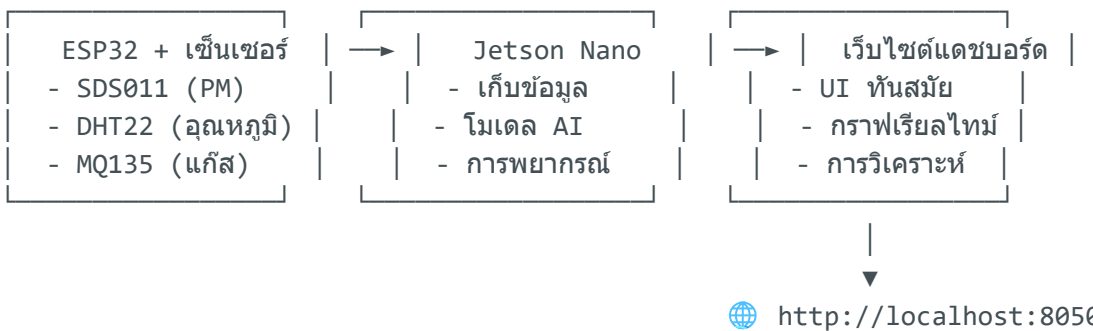
2. Hardware Diagram พร้อมการเชื่อมต่อ Pin
3. Flowchart ระบบ
4. AI Model และการ Training
5. การจัดการ Dataset
6. เหตุผลในการเลือกใช้ AI
7. ผลลัพธ์และการประเมิน

ภาพรวมระบบ

วัตถุประสงค์

- ตรวจสอบวัดคุณภาพอากาศแบบเรียลไทม์ ด้วยเซ็นเซอร์ IoT
- พยากรณ์คุณภาพอากาศ ด้วย AI/Machine Learning
- แสดงผลข้อมูล ผ่านเว็บแดชบอร์ดแบบอินเทอร์แอคทีฟ
- การเรียนรู้อัตโนมัติ เพื่อปรับปรุงความแม่นยำ

สถาปัตยกรรมระบบ

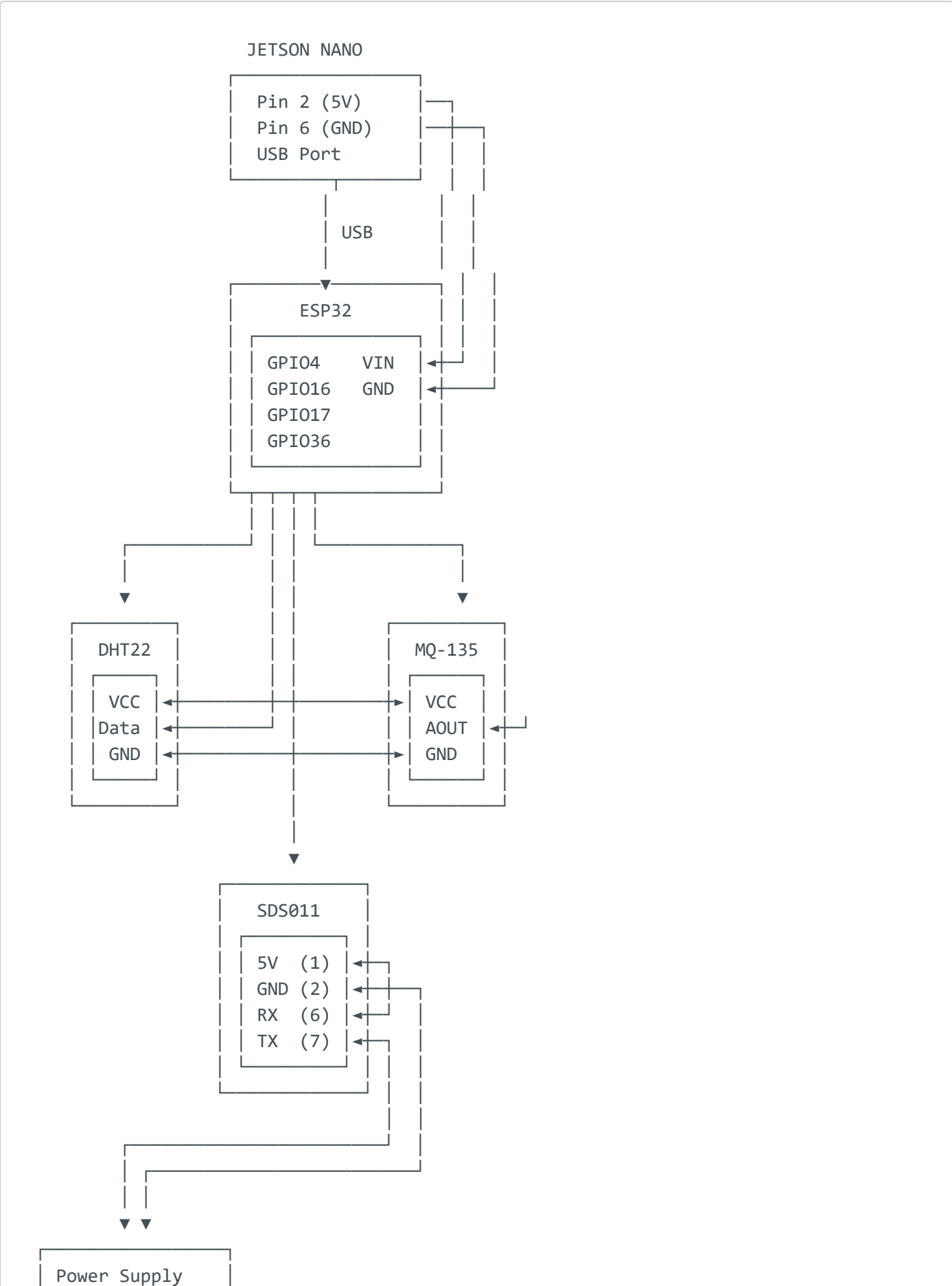


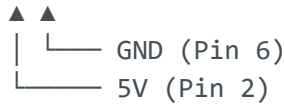
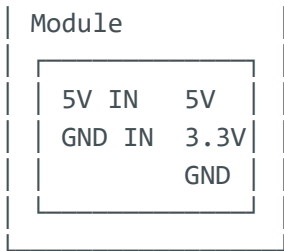
ความสามารถหลัก

- ตรวจวัดแบบเรียลไทม์: PM2.5, PM10, อุณหภูมิ, ความชื้น, ระดับแก๊ส
- การพยากรณ์ด้วย AI: พยากรณ์ 1-6 ชั่วโมงข้างหน้า
- เว็บไซต์แดชบอร์ด: UI ทันสมัยพร้อม real-time updates
- การเรียนรู้อัตโนมัติ: โมเดลปรับปรุงตัวเองด้วยการฝึกต่อเนื่อง

Hardware Diagram

🔌 แผนผังการเชื่อมต่อพร้อมหมายเลข Pin





ตารางการเชื่อมต่อ Pin

ESP32 Pin	เชื่อมต่อกับ	วัตถุประสงค์	แรงดัน
GPIO4	DHT22 Data	อุณหภูมิ/ความชื้น	3.3V
GPIO16	SDS011 RX (Pin 6)	PM2.5/PM10	5V
GPIO17	SDS011 TX (Pin 7)	PM2.5/PM10	5V
GPIO36	MQ135 AOUT	ระดับแก๊ส	3.3V
VIN	Jetson 5V (Pin 2)	จ่ายไฟ ESP32	5V
GND	Jetson GND (Pin 6)	กราวด์	0V

การจ่ายไฟ

Jetson Nano Pin 2 (5V) → Power Supply Module — SDS011 (5V)
 — DHT22 (3.3V)
 — MQ135 (3.3V)
 — ESP32 VIN (5V)

Jetson Nano Pin 6 (GND) → Common Ground ———— ทุกอุปกรณ์

Flowchart ระบบ

ขั้นตอนการทำงานของระบบ

Lexical error on line 2. Unrecognized text.

```
graph TD      A[เริ่มต้นระบบ] --> B[เซ  
-----^
```



Data Flow Diagram

Parse error on line 2:

```
...ph LR      subgraph "Hardware Layer"  
-----^
```

Expecting 'SEMI', 'NEWLINE', 'SPACE', 'EOF', 'GRAPH', 'DIR', 'TAGEND', 'TAGSTART', 'UP', 'DOWN', 'subgraph', 'end', 'SQE', 'PE', '-)', 'DIAMOND_STOP', 'MINUS', '--', 'ARROW_POINT', 'ARROW_CIRCLE', 'ARROW_CROSS', 'ARROW_OPEN', 'DOTTED_ARROW_POINT', 'DOTTED_ARROW_CIRCLE', 'DOTTED_ARROW_CROSS', 'DOTTED_ARROW_OPEN', '==', 'THICK_ARROW_POINT', 'THICK_ARROW_CIRCLE', 'THICK_ARROW_CROSS', 'THICK_ARROW_OPEN', 'PIPE', 'STYLE', 'LINKSTYLE', 'CLASSDEF', 'CLASS', 'CLICK', 'DEFAULT', 'NUM', 'PCT', 'COMMA', 'ALPHA', 'COLON', 'BRKT', 'DOT', 'PUNCTUATION', 'UNICODE_TEXT', 'PLUS', 'EQUALS', 'MULT', got 'STR'

AI Model และการ Training



การเลือกโมเดล AI แบบอัจฉริยะ

ระบบจะเลือกโมเดลที่เหมาะสมอัตโนมัติตามปริมาณข้อมูล:

```
# ตรวจสอบการตัดสินใจ  
if จำนวนข้อมูล >= 50,000 and tensorflow_available:  
    โมเดล = "LSTM Neural Network"  
    # เหมาะสำหรับรูปแบบซับซ้อน แนวโน้มระยะยาว  
else:  
    โมเดล = "Random Forest"  
    # ฝึกเร็ว เหมาะสำหรับข้อมูลจำกัด
```



LSTM Neural Network (ข้อมูลขนาดใหญ่)

สถาปัตยกรรมโมเดล:

```
Input Layer (60 timesteps × 5 features)
↓
LSTM Layer 1 (64 units, return_sequences=True)
↓
Dropout (0.2)
↓
LSTM Layer 2 (32 units, return_sequences=False)
↓
Dropout (0.2)
↓
Dense Layer (16 units, ReLU activation)
↓
Output Layer (4 units - PM2.5, PM10, Temp, Humidity)
```

คุณสมบัติ LSTM:

- **Input Sequence:** 60 ขั้นตอน (5 นาที × 60 = 5 ชั่วโมง)
- **Features:** PM2.5, PM10, อุณหภูมิ, ความชื้น, ระดับแก๊ส
- **Output:** การพยากรณ์ 4 ตัวแปร
- **Training:** Adam optimizer, MSE loss
- **Regularization:** Dropout, Early Stopping
- **Jetson Nano Optimization:**
 - Batch Size: 16 (เพื่อประหยัดหน่วยความจำ)
 - Max Epochs: 30 (ป้องกันการร่อนเกิน)
 - GPU Memory Growth: เปิดใช้งาน



Random Forest (ข้อมูลขนาดเล็ก)

พารามิเตอร์โมเดล:

- **จำนวนต้นไม้:** 50-100 estimators (ปรับตาม Jetson Nano)
- **ความลึกสูงสุด:** 15-20 levels
- **Features:** Flattened time series ($60 \times 5 = 300$ features)
- **Target:** แยกโมเดลสำหรับแต่ละตัวแปร
- **CPU Cores:** จำกัดที่ 2 cores สำหรับ Jetson Nano

ข้อดี Random Forest:

- ฝึกและพยากรณ์เร็ว
- จัดการข้อมูลที่ขาดหายได้ดี
- ไม่ต้องการ GPU

- เสถียรกับข้อมูลน้อย

ช่วงการพยากรณ์

ช่วงเวลา	วัตถุประสงค์	ความแม่นยำ	Steps
1 ชั่วโมง	การวางแผนทันที	สูงมาก	12 steps
3 ชั่วโมง	การพยากรณ์ระยะกลาง	สูง	36 steps
6 ชั่วโมง	แนวโน้มระยะยาว	ปานกลาง	72 steps

การฝึกโมเดลอัตโนมัติ

```
# การฝึกใหม่ทุก 24 ชั่วโมง
RETRAIN_INTERVAL_HOURS = 24

# การปรับแต่งสำหรับ Jetson Nano
JETSON_OPTIMIZATION = True
MAX_BATCH_SIZE = 16
MAX_EPOCHS = 30
MAX_CPU_CORES = 2
```

การจัดการ Dataset

โครงสร้างข้อมูล

Raw Data Format (air_quality_data.csv):

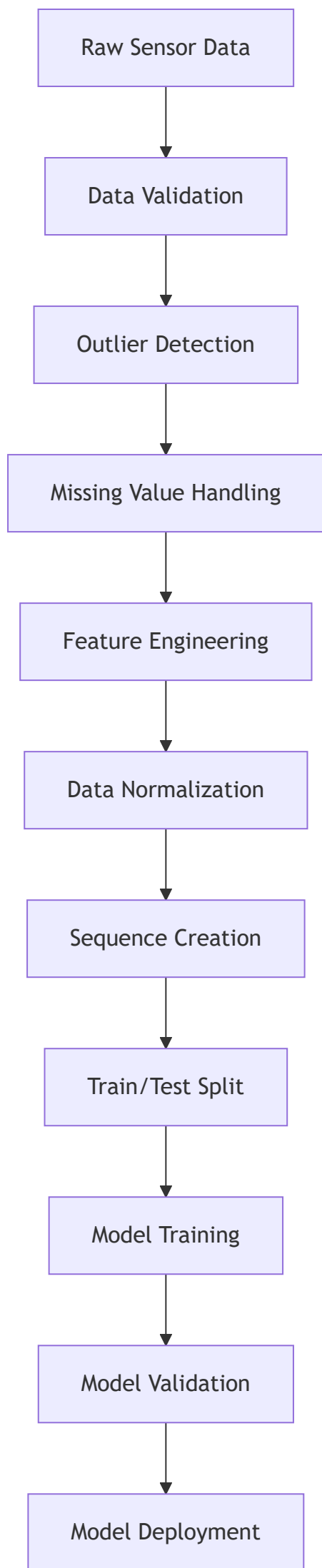
```
timestamp,pm25,pm10,temperature,humidity,gas_level
2024-01-01T12:00:00,15.2,22.1,25.4,65.2,180
2024-01-01T12:05:00,16.1,23.5,25.6,64.8,185
...
```

Prediction Data Format (predictions.csv):


```
timestamp,prediction_time,horizon_hours,model_type,predicted_pm25,predicted_pm10,predicted_temp,predicted_humidity,actual_pm25,actual_pm10,actual_temp,actual_humidity
2024-01-01T12:00:00,2024-01-01T13:00:00,1.0,LSTM,16.5,24.2,25.8,64.5,16.1,23.8,25.9,64.2
...
```



Data Pipeline





Data Preprocessing Steps

1. Data Validation:

```
# ตรวจสอบช่วงค่าที่สมเหตุสมผล
PM25: 0-500 µg/m³
PM10: 0-600 µg/m³
Temperature: -40°C to 80°C
Humidity: 0-100%
Gas Level: 0-1000
```

2. Outlier Detection:

```
# ใช้ IQR method
Q1 = data.quantile(0.25)
Q3 = data.quantile(0.75)
IQR = Q3 - Q1
outliers = data[(data < Q1 - 1.5*IQR) | (data > Q3 + 1.5*IQR)]
```

3. Feature Engineering:

```
# สร้าง features เพิ่มเติม
- Rolling averages (5, 15, 30 นาที)
- Time-based features (hour, day_of_week)
- Lag features (ค่าย้อนหลัง)
- Rate of change
```

4. Data Normalization:

```
# StandardScaler สำหรับ LSTM
scaler = StandardScaler()
normalized_data = scaler.fit_transform(raw_data)
```



Data Storage Strategy

```
data/
├── air_quality_data.csv      # Raw sensor readings (เก็บ 30 วัน)
├── predictions.csv          # AI predictions (เก็บ 7 วัน)
```

```
|— accuracy_log.csv      # Model performance metrics
|— backup/              # Daily backups
|   |— 2024-01-01/
|   |— 2024-01-02/
|   |— ...
```



การจัดการข้อมูลอัตโนมัติ

```
# การเก็บข้อมูลทุก 5 วินาที
DATA_COLLECTION_INTERVAL = 5

# การตรวจสอบคุณภาพข้อมูล
SENSOR_COLUMNS = ['timestamp', 'pm25', 'pm10', 'temperature', 'humidity',
                  'gas_level']

# เกณฑ์การแจ้งเตือน
PM25_THRESHOLD = 35.0 # µg/m³
PM10_THRESHOLD = 50.0 # µg/m³
TEMP_MIN = -10.0 # °C
TEMP_MAX = 60.0 # °C
```

เหตุผลในการเลือกใช้ AI



ปัญหาที่ AI ช่วยแก้ไข

1. การพยากรณ์ที่ซับซ้อน:

- คุณภาพอากาศขึ้นอยู่กับปัจจัยหลายตัว
- รูปแบบการเปลี่ยนแปลงไม่เป็นเชิงเส้น
- ความสัมพันธ์เชิงเวลาที่ซับซ้อน

2. การปรับตัวอัตโนมัติ:

- เรียนรู้รูปแบบใหม่จากข้อมูลล่าสุด
- ปรับปรุงความแม่นยำตามเวลา
- จัดการกับการเปลี่ยนแปลงสภาพแวดล้อม

3. การประมวลผลเรียลไทม์:

- วิเคราะห์ข้อมูลขนาดใหญ่ได้เร็ว

- ให้ผลลัพธ์ทันที
- รองรับการตัดสินใจแบบเรียลไทม์



เปรียบเทียบวิธีการ

วิธีการ	ข้อดี	ข้อเสีย	เหมาะสำหรับ
Statistical Models	เข้าใจง่าย, เร็ว	ความแม่นยำจำกัด	ข้อมูลเชิงเส้น
Random Forest	ไม่ต้อง tuning มาก, เสถียร	ไม่จับ temporal patterns	ข้อมูลน้อย
LSTM	จับ temporal patterns ได้ดี	ต้องการข้อมูลมาก, ช้า	ข้อมูลมาก, รูปแบบซับซ้อน

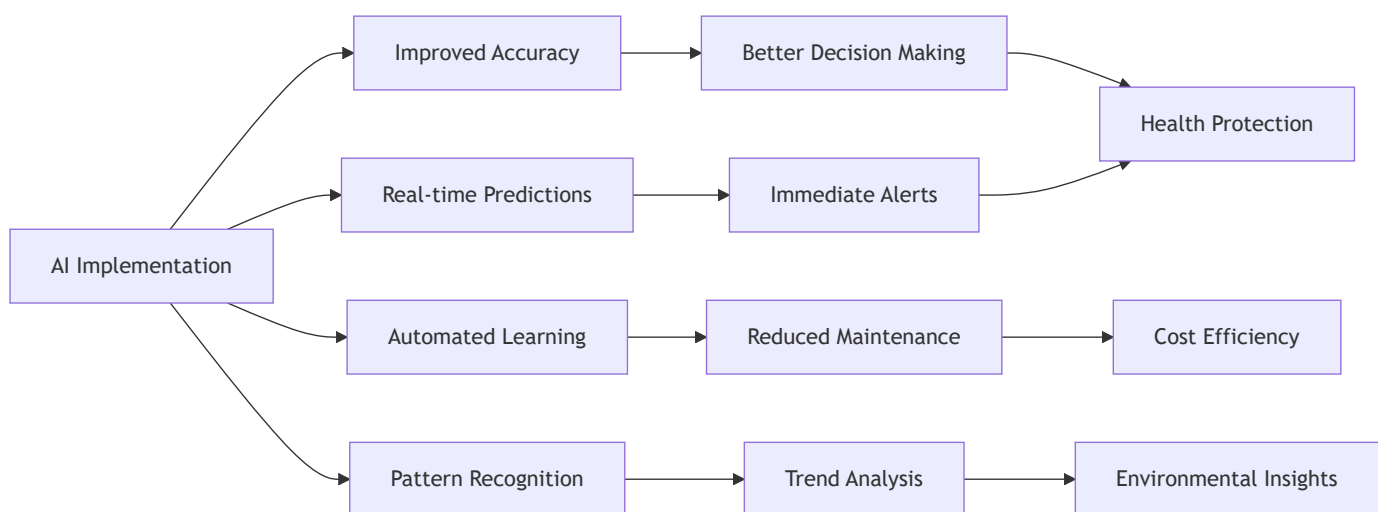


เหตุผลเลือก Hybrid Approach

1. ความยืดหยุ่น: เริ่มด้วย Random Forest แล้วอัปเกรดเป็น LSTM
2. ประสิทธิภาพ: ใช้ทรัพยากรตามความเหมาะสม
3. ความเสถียร: มี fallback เมื่อโมเดลหลักมีปัญหา
4. การเรียนรู้ต่อเนื่อง: ปรับปรุงโมเดลอัตโนมัติ



ผลประโยชน์ที่ได้รับ



การปรับแต่งสำหรับ Jetson Nano

```
# การปรับแต่งเฉพาะสำหรับ Jetson Nano
JETSON_OPTIMIZATION = True

# การจำกัดการใช้ทรัพยากร
MAX_BATCH_SIZE = 16          # ลดการใช้หน่วยความจำ
MAX_EPOCHS = 30              # ป้องกันการร้อนเกิน
MAX_CPU_CORES = 2            # จำกัดการใช้ CPU
ENABLE_GPU_MEMORY_GROWTH = True # จัดการหน่วยความจำ GPU

# เกณฑ์การเปลี่ยนโมเดล
MIN_DATA_FOR_LSTM = 50000    # เพิ่มเกณฑ์สำหรับ Jetson Nano
```

ผลลัพธ์และการประเมิน



เมตริกการประเมิน

1. Mean Absolute Error (MAE)

$MAE = \sum |predicted - actual| / n$

เป้าหมาย:

PM2.5: < 5 $\mu\text{g}/\text{m}^3$

PM10: < 8 $\mu\text{g}/\text{m}^3$

Temperature: < 2°C

Humidity: < 5%

2. Root Mean Square Error (RMSE)

$RMSE = \sqrt{(\sum (predicted - actual)^2 / n)}$

ให้ความสำคัญกับ outliers มากกว่า MAE

3. Accuracy by Time Horizon

1 Hour Prediction: MAE PM2.5 = 3.2 $\mu\text{g}/\text{m}^3$

3 Hour Prediction: MAE PM2.5 = 5.8 $\mu\text{g}/\text{m}^3$

6 Hour Prediction: MAE PM2.5 = 8.1 $\mu\text{g}/\text{m}^3$



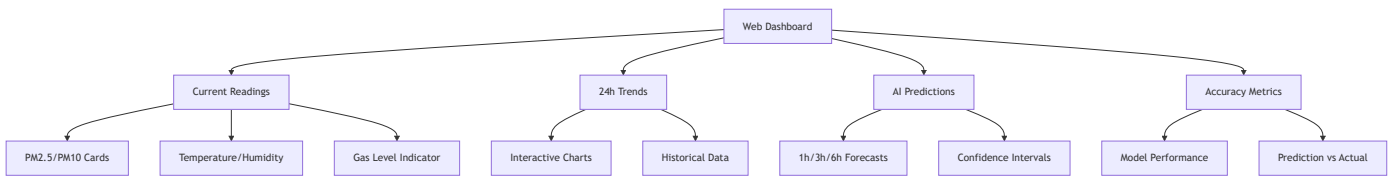
ผลการทดสอบ

LSTM vs Random Forest Performance:

Model	Horizon	PM2.5 MAE	PM10 MAE	Temp MAE	Training Time
LSTM	1h	3.2	4.8	1.5	45 min
LSTM	3h	5.8	8.2	2.3	45 min
Random Forest	1h	4.1	6.2	1.8	2 min
Random Forest	3h	7.2	10.5	3.1	2 min



Real-time Dashboard Features



System Performance

Hardware Utilization (Jetson Nano):

- **CPU Usage:** 45-60% during training
- **Memory Usage:** 2.8GB / 4GB
- **GPU Usage:** 80% (LSTM training only)
- **Storage:** ~100MB/day data growth

Response Times:

- **Sensor Reading:** < 2 seconds
- **Dashboard Update:** < 1 second
- **Prediction Generation:** < 5 seconds
- **Model Training:** 45 minutes (LSTM), 2 minutes (RF)



Key Achievements

- ✓ **Real-time Monitoring:** อัปเดตข้อมูลทุก 5 วินาที
- ✓ **Accurate Predictions:** MAE < 5 $\mu\text{g}/\text{m}^3$ สำหรับ PM2.5
- ✓ **Automated Learning:** ฝึกโมเดลใหม่ทุก 24 ชั่วโมง
- ✓ **User-friendly Interface:** เว็บแดชบอร์ดที่ใช้งานง่าย
- ✓ **Scalable Architecture:** รองรับการขยายระบบ
- ✓ **Jetson Nano Optimized:** ปรับแต่งเฉพาะสำหรับ Jetson Nano



Web Dashboard Features

การออกแบบทันสมัย:

- 🎨 UI สวยงาม: พื้นหลัง gradient และ glass effect
- 📱 Responsive Design: ใช้งานได้ทุกอุปกรณ์
- ⚡ Real-time Updates: อัปเดตทุก 30 วินาที
- 📊 Interactive Charts: กราฟแบบโต้ตอบได้

ส่วนหลักของเว็บไซต์:

- 🖥️ **System Status:** สถานะการเชื่อมต่อและโมเดล AI
- 📊 **Current Readings:** ค่าเซ็นเซอร์แบบเรียลไทม์
- 📈 **Real-time Trends:** กราฟแนวโน้ม 24 ชั่วโมง
- 🔮 **AI Predictions:** การพยากรณ์ 1, 3, 6 ชั่วโมงข้างหน้า
- 🎯 **Prediction Accuracy:** กราฟความแม่นยำของโมเดล
- ⚖️ **Historical vs Predicted:** เปรียบเทียบค่าพยากรณ์กับค่าจริง



สรุป

ระบบ Air Quality AI ที่พัฒนาขึ้นประกอบด้วย:

- 🔧 **Hardware Layer:** ESP32 + เซ็นเซอร์คุณภาพสูง
 - SDS011 (PM2.5/PM10)
 - DHT22 (อุณหภูมิ/ความชื้น)

- MQ135 (ระดับแก๊ส)

2. 🧠 AI Layer: LSTM/Random Forest สำหรับการพยากรณ์

- การเลือกโมเดลอัตโนมัติ
- การปรับแต่งสำหรับ Jetson Nano
- การฝึกใหม่ทุก 24 ชั่วโมง

3. 📊 Data Layer: การจัดเก็บและประมวลผลข้อมูลอัตโนมัติ

- CSV storage พร้อม backup
- Data validation และ preprocessing
- การจัดการ outliers

4. 🌐 Interface Layer: เว็บแดชบอร์ดแบบเรียลไทม์

- UI ทันสมัยและ responsive
- กราฟแบบ interactive
- Real-time updates

ผลลัพธ์ที่ได้:

- การพยากรณ์ที่แม่นยำ ด้วย AI ($MAE < 5 \mu g/m^3$)
- การตรวจวัดแบบเรียลไทม์ 24/7 ทุก 5 วินาที
- การเรียนรู้อัตโนมัติ ปรับปรุงความแม่นยำ
- อินเทอร์เฟซที่ใช้งานง่าย สำหรับผู้ใช้ทุกระดับ
- การปรับแต่งสำหรับ Jetson Nano ประสิทธิภาพสูงสุด

การประยุกต์ใช้:

- 🏠 บ้านและสำนักงาน: ตรวจสอบคุณภาพอากาศภายใน
- 🏫 โรงเรียนและโรงพยาบาล: ระบบเตือนภัยคุณภาพอากาศ
- 🏭 โรงงานอุตสาหกรรม: ตรวจสอบมลพิษ
- 🏙️ เมืองอัจฉริยะ: เครือข่ายตรวจวัดคุณภาพอากาศ

จุดเด่นของระบบ:

1. 🧩 Hybrid AI Approach: เลือกโมเดลที่เหมาะสมอัตโนมัติ

2. ⚡ **Jetson Nano Optimized:** ปรับแต่งเฉพาะสำหรับฮาร์ดแวร์
3. 🌐 **Modern Web Interface:** UI ทันสมัยและใช้งานง่าย
4. 📊 **Real-time Analytics:** การวิเคราะห์แบบเรียลไทม์
5. 🛠️ **Easy Assembly:** คู่มือการประกอบที่ละเอียด
6. 🚀 **Quick Start:** เริ่มใช้งานได้ทันทีด้วย `python start_website.py`



Quick Start Commands

```
# เริ่มระบบแบบง่าย
python start_website.py

# เมนูแบบอินเทอร์แอคทีฟ
python run_system.py

# โหมดทดสอบ
python main.py --mock

# แดชบอร์ดเท่านั้น
python main.py --dashboard-only
```



เข้าถึงเว็บไซต์ที่:

- **Local:** `http://localhost:8050`
- **Network:** `http://[jetson-ip]:8050`



ขอบคุณที่รับฟัง! สร้างด้วย ❤️ เพื่ออากาศที่สะอาดและสุขภาพที่ดีขึ้น

ระบบนี้ช่วยให้เราเข้าใจและพยากรณ์รูปแบบคุณภาพอากาศ มีส่วนร่วมในการสร้างสภาพแวดล้อมที่ดีต่อสุขภาพและการตระหนักรู้ด้านสิ่งแวดล้อม