

# การพยากรณ์ Load Profile (Load Profile Forecasting)

The BESS Project

# การพยากรณ์ Load Profile รายชั่วโมง: เปรียบเทียบ SARIMA กับ LSTM

## วัตถุประสงค์

- พยากรณ์การใช้ไฟฟ้าในอนาคต (Load Profiles) โดยใช้การวิเคราะห์อนุกรมเวลา (Time Series Forecasting)
- เปรียบเทียบประสิทธิภาพ ของโมเดลทางสถิติแบบดั้งเดิม (SARIMA) กับโมเดลการเรียนรู้เชิงลึก (LSTM)

# ขั้นตอนสำคัญ

## 1. การสร้างข้อมูล:

- สร้างข้อมูลโหลดไฟฟ้าจำลองรายชั่วโมง เป็นเวลา 3 เดือน (มีนาคม - พฤษภาคม 2025)
- ข้อมูลประกอบด้วยรูปแบบที่สมจริง: โหลดพื้นฐาน (base load), รูปแบบตามรายวันและรายสัปดาห์, แนวโน้มเล็กน้อย, และสัญญาณรบกวน (noise)

## 2. การเตรียมข้อมูล:

- ข้อมูลที่สร้างขึ้นจะถูกแบ่งออกเป็นชุดฝึก (**training set**) และชุดทดสอบ (**testing set**) (ใช้ 7 วันสุดท้ายสำหรับการทดสอบ)
- สำหรับ **LSTM**: ข้อมูลจะถูกปรับขนาด (ให้อยู่ในช่วง 0-1) และแปลงเป็นลำดับ (**sequences**) (เช่น การใช้ข้อมูล 24 ชั่วโมงที่ผ่านมาเพื่อพยากรณ์ชั่วโมงถัดไป)

### 3. โมเดล SARIMA:

- **Seasonal AutoRegressive Integrated Moving Average (SARIMA)** ถูกฝึกด้วยข้อมูลย้อนหลัง
- กำหนดค่าพารามิเตอร์ (เช่น  $(1,1,1)$  สำหรับส่วนที่ไม่ใช่ฤดูกาล,  $(1,1,0,24)$  สำหรับส่วนฤดูกาลรายวัน)
- สร้าง**การพยากรณ์**สำหรับช่วงเวลาทดสอบ

## 4. โมเดล LSTM:

- สร้างเครือข่ายประสาทเทียมแบบ **Long Short-Term Memory (LSTM)** โดยใช้ TensorFlow/Keras
- ฝึกด้วยข้อมูลลำดับที่ปรับขนาดแล้ว
- ทำการพยากรณ์แบบ **Rolling One-step-ahead** สำหรับช่วงเวลาที่ทดสอบ

## 5. การประเมินและการแสดงผล:

- **เมตริกประสิทธิภาพ:** โมเดลทั้งสองจะถูกประเมินโดยใช้ **Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE)**, และ **Mean Absolute Error (MAE)**
- **การเปรียบเทียบด้วยภาพ:** แผนภูมิจะแสดงโหลดจริง, การพยากรณ์ของ **SARIMA**, และการพยากรณ์ของ **LSTM** บนกราฟเดียวกัน เพื่อให้เห็นภาพการเปรียบเทียบความแม่นยำได้อย่างชัดเจน

# การพยากรณ์ Load Profile ล่วงหน้า 6 ชั่วโมง (ครั้งเดียว)

## วัตถุประสงค์

- เพื่อแสดงความสามารถของโมเดล SARIMA และ LSTM ในการ **พยากรณ์ Load Profile สำหรับ 6 ชั่วโมงข้างหน้าเพียงครั้งเดียว** ต่อจากข้อมูลการฝึกอบรมล่าสุด
- เน้นการพยากรณ์แบบ **Multi-step** จากจุดเริ่มต้นเดียว



# โมเดล SARIMA (พยากรณ์ 6 ชั่วโมง)

- **การฝึกโมเดล:** ฝึกโมเดล SARIMA ด้วยข้อมูลฝึกอบรมทั้งหมด
- **การพยากรณ์:**
  - ใช้เมธอด `predict()` โดยระบุจุดเริ่มต้นและสิ้นสุดของ 6 ชั่วโมงที่ต้องการพยากรณ์อย่างชัดเจน
  - ตั้งค่า `dynamic=True` เพื่อให้โมเดลใช้ ผลการพยากรณ์ของตัวเอง ในการพยากรณ์ชั่วโมงถัดไปภายในช่วง 6 ชั่วโมง (เป็นการพยากรณ์แบบ Multi-step ที่แท้จริง)

# โมเดล LSTM (พยากรณ์ 6 ชั่วโมง)

- **การฝึกโมเดล:**

- ฝึกโมเดล LSTM ให้เรียนรู้ที่จะพยากรณ์ 6 ชั่วโมงข้างหน้าพร้อมกันในการทำนายแต่ละครั้ง ( $n\_steps\_out = 6$ )
- เลเยอร์เอาต์พุตของโมเดลถูกตั้งค่าให้ส่งออก 6 ค่า

- **การพยากรณ์:**

- ป้อนข้อมูล 24 ชั่วโมงสุดท้าย จากชุดฝึกอบรม ( $n\_steps\_in = 24$ ) เข้าสู่โมเดล เพียงครั้งเดียว
- โมเดลจะส่งคืนผลลัพธ์ 6 ค่า ซึ่งเป็นการพยากรณ์โหลดสำหรับ 6 ชั่วโมงถัดไปในอนาคต

# การประเมินผล

- คำนวณ RMSE (Root Mean Squared Error) และ MAE (Mean Absolute Error) โดยเปรียบเทียบผลการพยากรณ์ทั้ง 6 ชั่วโมงของแต่ละโมเดล กับค่าโหลดจริงในช่วง 6 ชั่วโมงนั้น
- แสดงผลด้วยกราฟที่เน้นเฉพาะช่วงข้อมูลล่าสุดและผลการพยากรณ์ 6 ชั่วโมง เพื่อให้เห็นภาพประสิทธิภาพของโมเดลอย่างชัดเจน

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.metrics import mean_squared_error, mean_absolute_error
from math import sqrt
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
import warnings

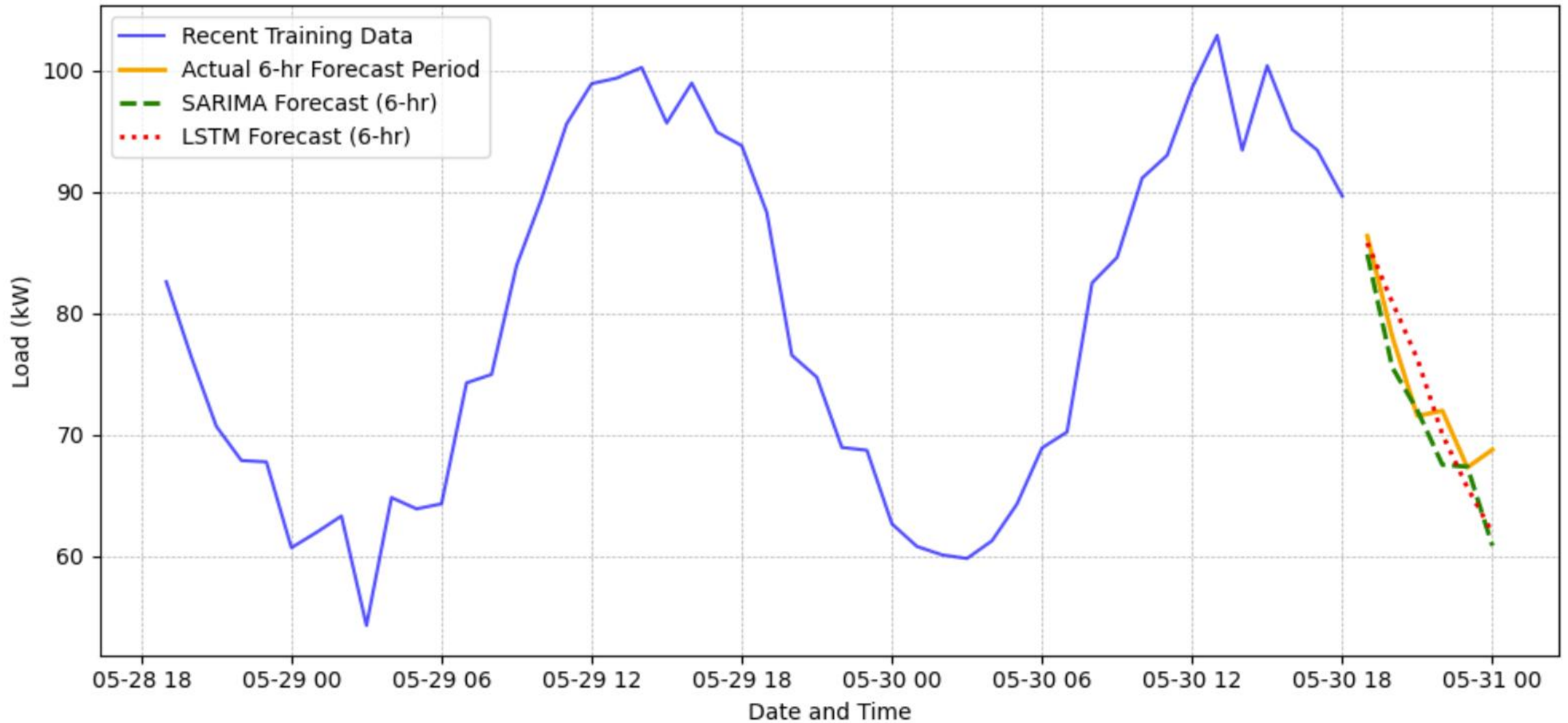
# Suppress harmless warnings
warnings.filterwarnings("ignore")

# --- 1. Generate Synthetic Hourly Load Profile Data ---
start_date = '2025-03-01'
end_date = '2025-05-31' # 3 months of hourly data
freq = 'H' # Hourly frequency

dates = pd.date_range(start=start_date, end=end_date, freq=freq)
n_points = len(dates)
```

[https://github.com/DrHammerhead/advanced\\_forecast/blob/d588a0b10e47a09e78bbd4b7ae1f8e32d23615e4/Load\\_Profile\\_LSTM\\_6h\\_onetime\\_3june25.ipynb](https://github.com/DrHammerhead/advanced_forecast/blob/d588a0b10e47a09e78bbd4b7ae1f8e32d23615e4/Load_Profile_LSTM_6h_onetime_3june25.ipynb)

# Single 6-hour Load Profile Forecast: SARIMA vs. LSTM



--- Summary of Performance for Single 6-hour Forecast ---

SARIMA: RMSE = 3.90, MAE = 2.82

LSTM: RMSE = 3.74, MAE = 3.10

- การกำหนดช่วงเวลาทดสอบ (Test Period):
- `test_period_hours = 7 * 24` กำหนดให้ช่วงข้อมูลทดสอบทั้งหมดคือ 7 วัน (168 ชั่วโมง)
- ทั้งโมเดล SARIMA และ LSTM จะถูกประเมินผลบนข้อมูลจริงใน 7 วันนี้
- `FORECAST_STEP_HORIZON = 6`:

# โมเดล SARIMA (พยากรณ์ต่อเนื่อง 7 วัน):

- การพยากรณ์: `sarima_model_fit.predict(start=len(train_data), end=len(load_profile) - 1, dynamic=True)`
- `start` และ `end` ครอบคลุมทั้ง 7 วันของข้อมูลทดสอบ
- `dynamic=True` มีความสำคัญอย่างยิ่ง: มันบอกให้ SARIMA ใช้ "ค่าที่โมเดลพยากรณ์ได้ในชั่วโมงก่อนหน้านี้" (ไม่ใช่ค่าจริง) เพื่อใช้ในการพยากรณ์ชั่วโมงถัดไป เป็นการจำลองสถานการณ์จริงที่โมเดลต้องพยากรณ์ยาวไปข้างหน้าโดยไม่มีข้อมูลจริงเข้ามา

# โมเดล LSTM (พยากรณ์ทีละ 6 ชม. วนไป 7 วัน):

- การเตรียมข้อมูลฝึกอบรม:
  - `n_steps_out = FORECAST_STEP_HORIZON` (คือ 6) หมายความว่าโมเดล LSTM จะถูกฝึกให้เรียนรู้ที่จะพยากรณ์ 6 ชั่วโมงข้างหน้าพร้อมกัน จากข้อมูลย้อนหลัง 24 ชั่วโมง (`n_steps_in = 24`)
- การพยากรณ์ในลูป:
  - โค้ดจะใช้ลูป `for i in range(0, len(test_data), FORECAST_STEP_HORIZON)`: ซึ่งจะวนลูปทีละ 6 ชั่วโมง ครอบคลุมตลอด 7 วันของข้อมูลทดสอบ



- ในแต่ละรอบของลูป:
  - โมเดลจะรับอินพุต (เช่น 24 ชั่วโมงล่าสุด) และทำการพยากรณ์ออกมา 6 ชั่วโมงทันที (`predicted_steps = lstm_model.predict(current_batch, verbose=0)[0]`)
  - ผลลัพธ์ 6 ชั่วโมงนี้จะถูกเพิ่มเข้าไปในรายการ `lstm_predictions_scaled`
  - จากนั้น `current_batch` (ชุดข้อมูลอินพุตสำหรับรอบถัดไป) จะถูกปรับปรุงให้ใช้ข้อมูลจริงจากชุดทดสอบ 6 ชั่วโมงถัดไป (`scaled_test_data[next_input_start_index : next_input_start_index + n_steps_in]`) ซึ่งเป็นการจำลองการที่โมเดลได้รับการ "อัปเดต" ด้วยข้อมูลจริงเมื่อมีข้อมูลใหม่เข้ามาในแต่ละช่วง 6 ชั่วโมง ก่อนที่จะพยากรณ์ 6 ชั่วโมงถัดไปอีกครั้ง

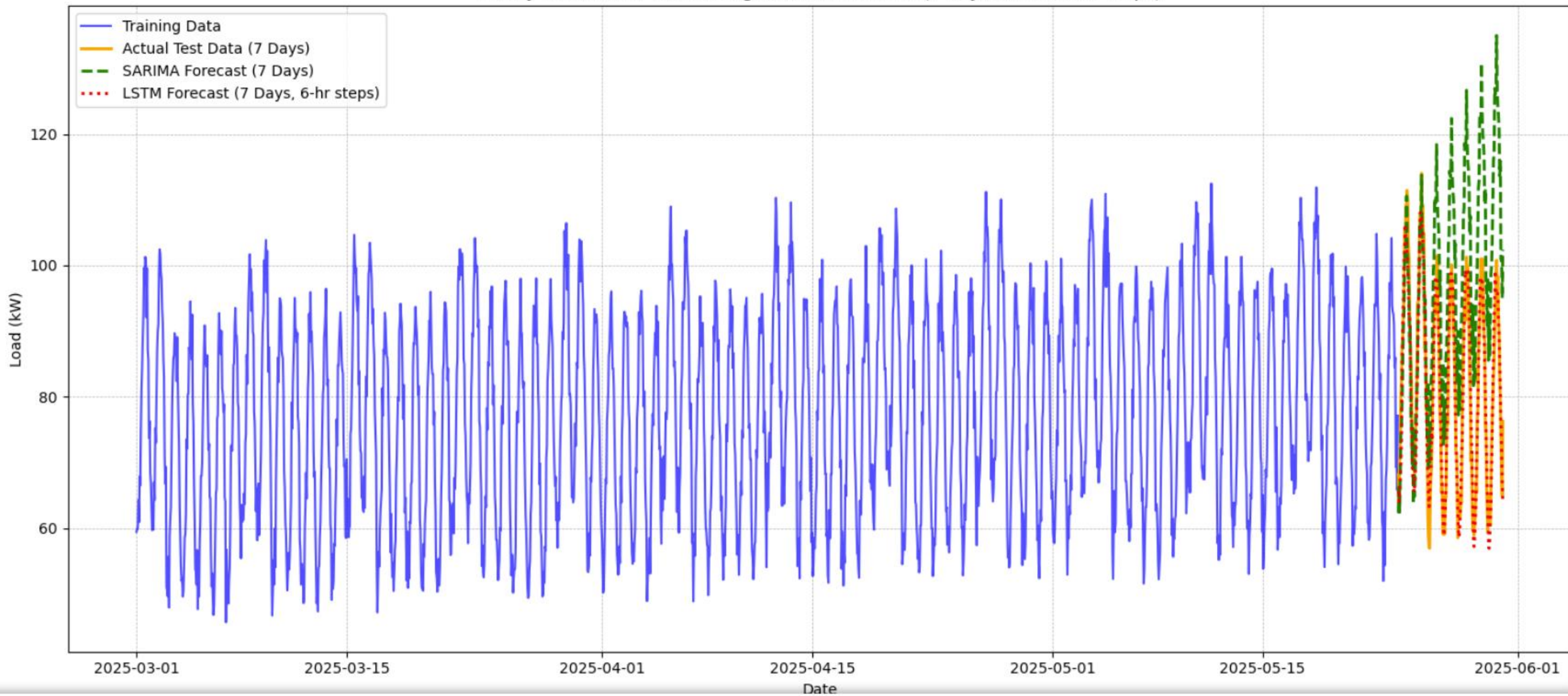
```
•[9]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.metrics import mean_squared_error, mean_absolute_error
from math import sqrt
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
import warnings

# Suppress harmless warnings
warnings.filterwarnings("ignore")

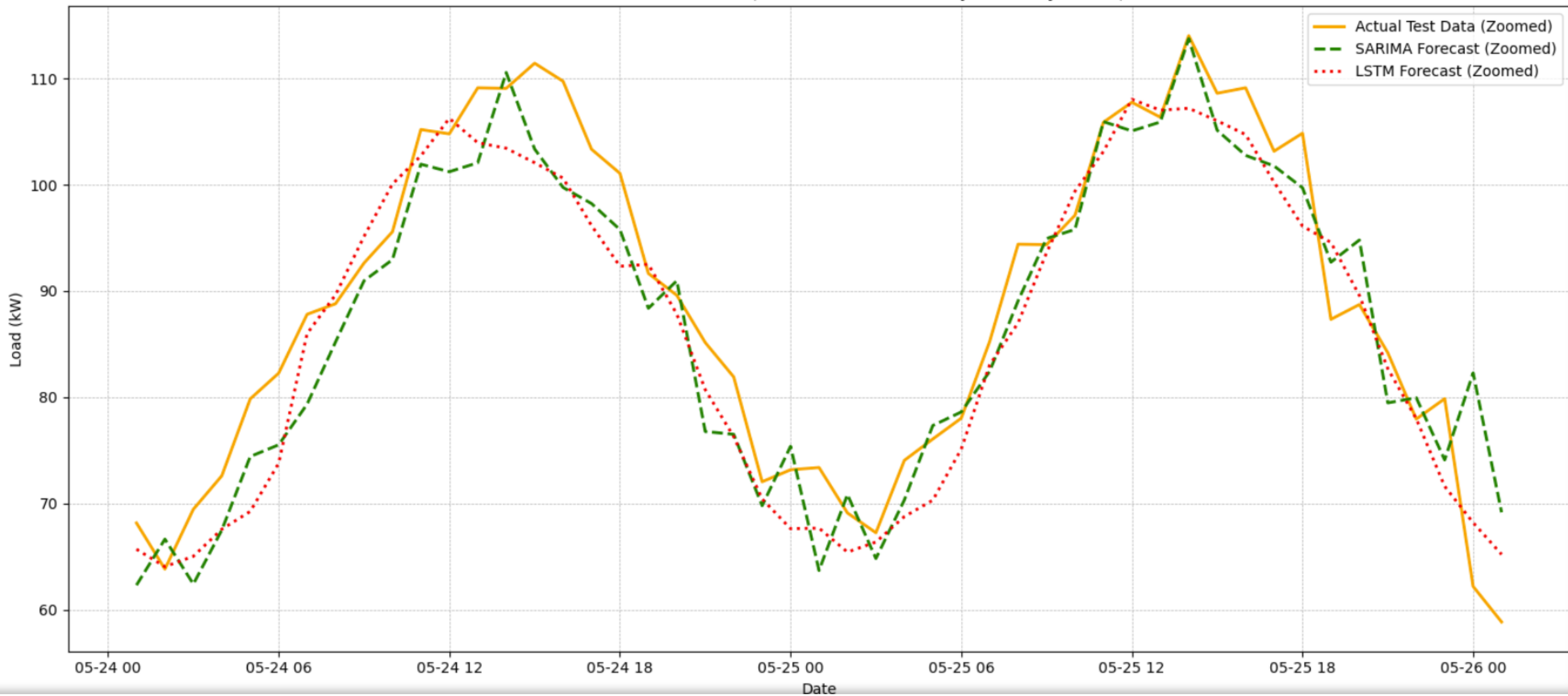
# --- 1. Generate Synthetic Hourly Load Profile Data ---
start_date = '2025-03-01'
end_date = '2025-05-31' # 3 months of hourly data
```

[https://github.com/DrHammerhead/advanced\\_forecast/blob/dba42ced026cf54c2cedf811a33e6c36a2f2e40a/Load\\_Profile\\_LSTM\\_6h\\_3june25.ipynb](https://github.com/DrHammerhead/advanced_forecast/blob/dba42ced026cf54c2cedf811a33e6c36a2f2e40a/Load_Profile_LSTM_6h_3june25.ipynb)

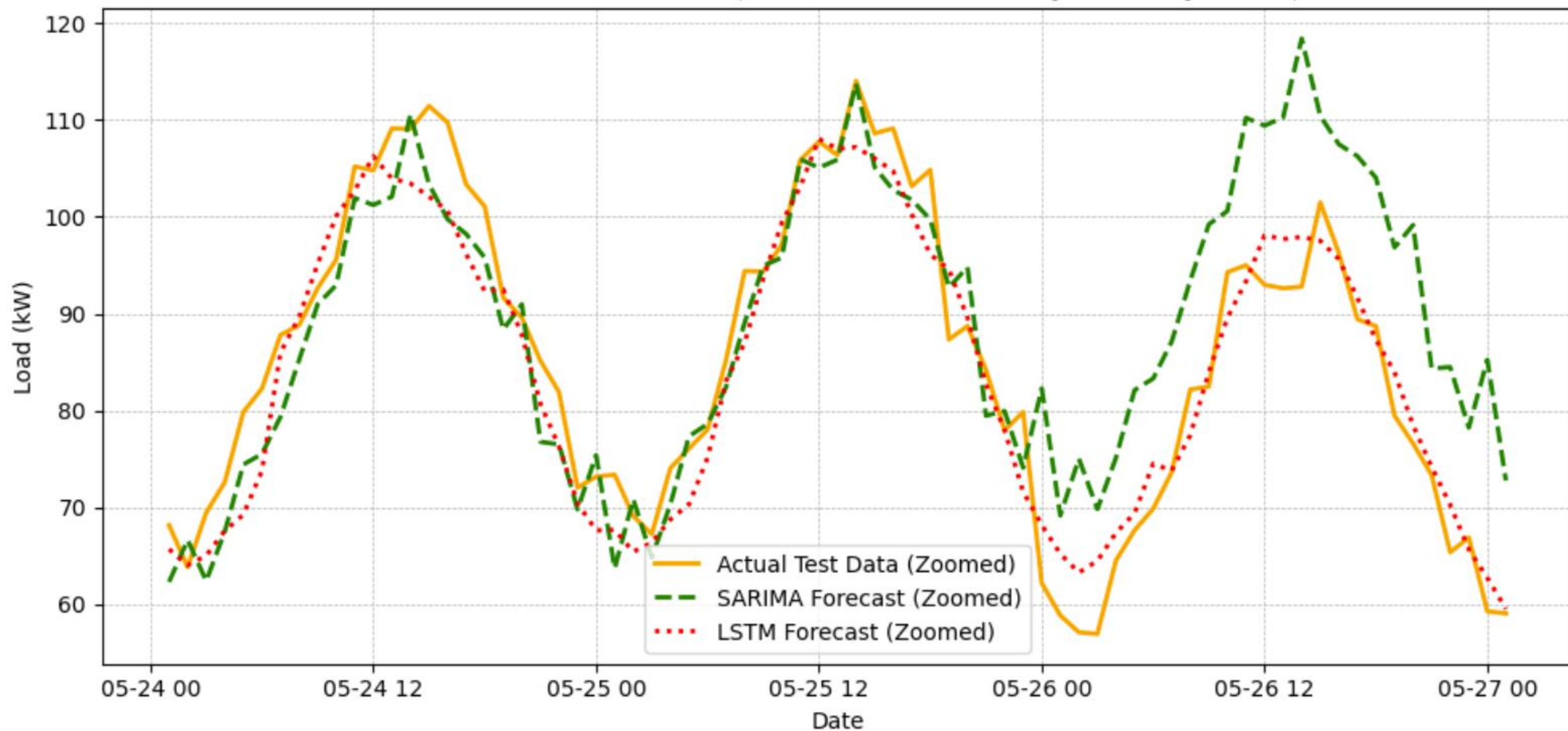
Hourly Load Profile Forecasting: SARIMA vs. LSTM (7-day Horizon, 6-hr steps)



SARIMA vs. LSTM Forecast (Zoomed into First 2 Days of 7-day Period)

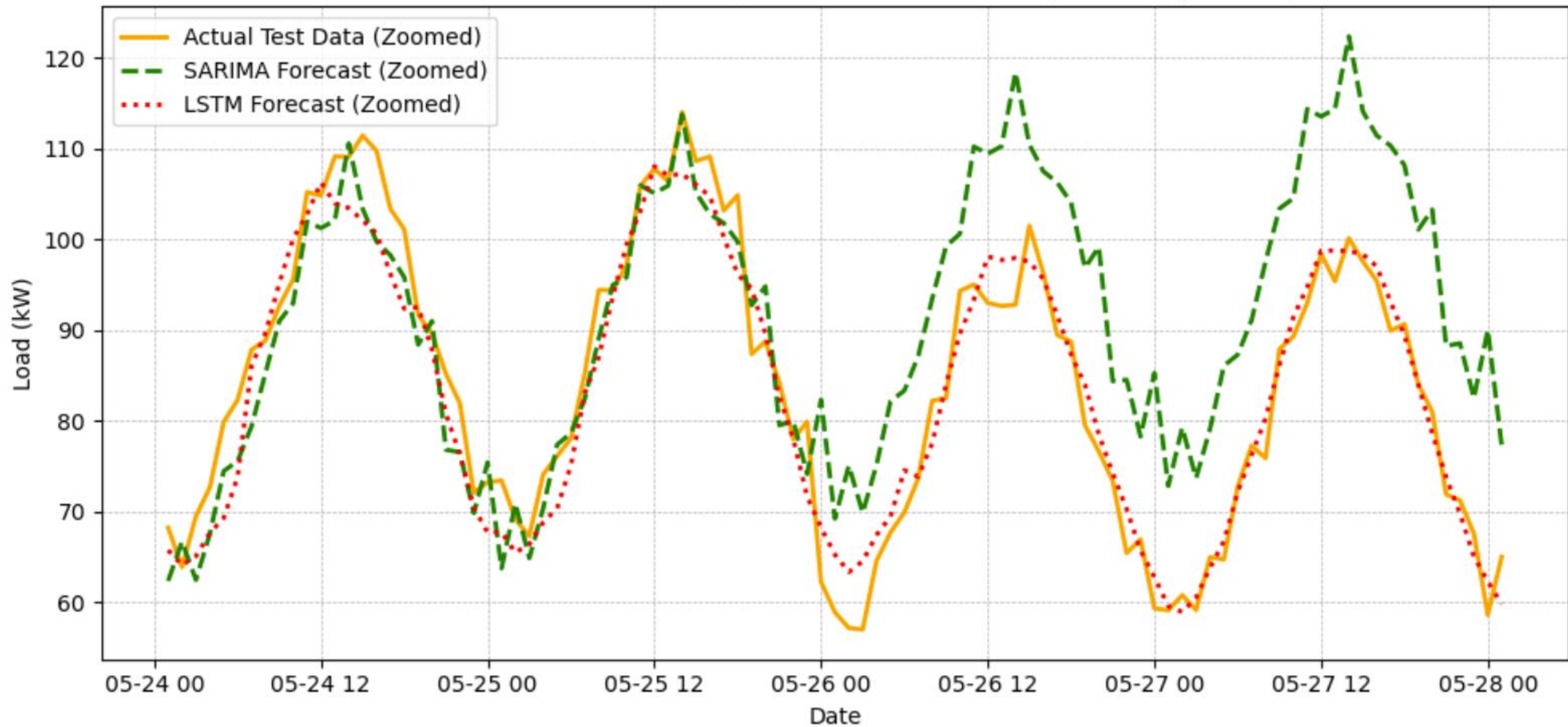


SARIMA vs. LSTM Forecast (Zoomed into First 3 Days of 7-day Period)





SARIMA vs. LSTM Forecast (Zoomed into First 4 Days of 7-day Period)



SARIMA vs. LSTM Forecast (Zoomed into First 5 Days of 7-day Period)

