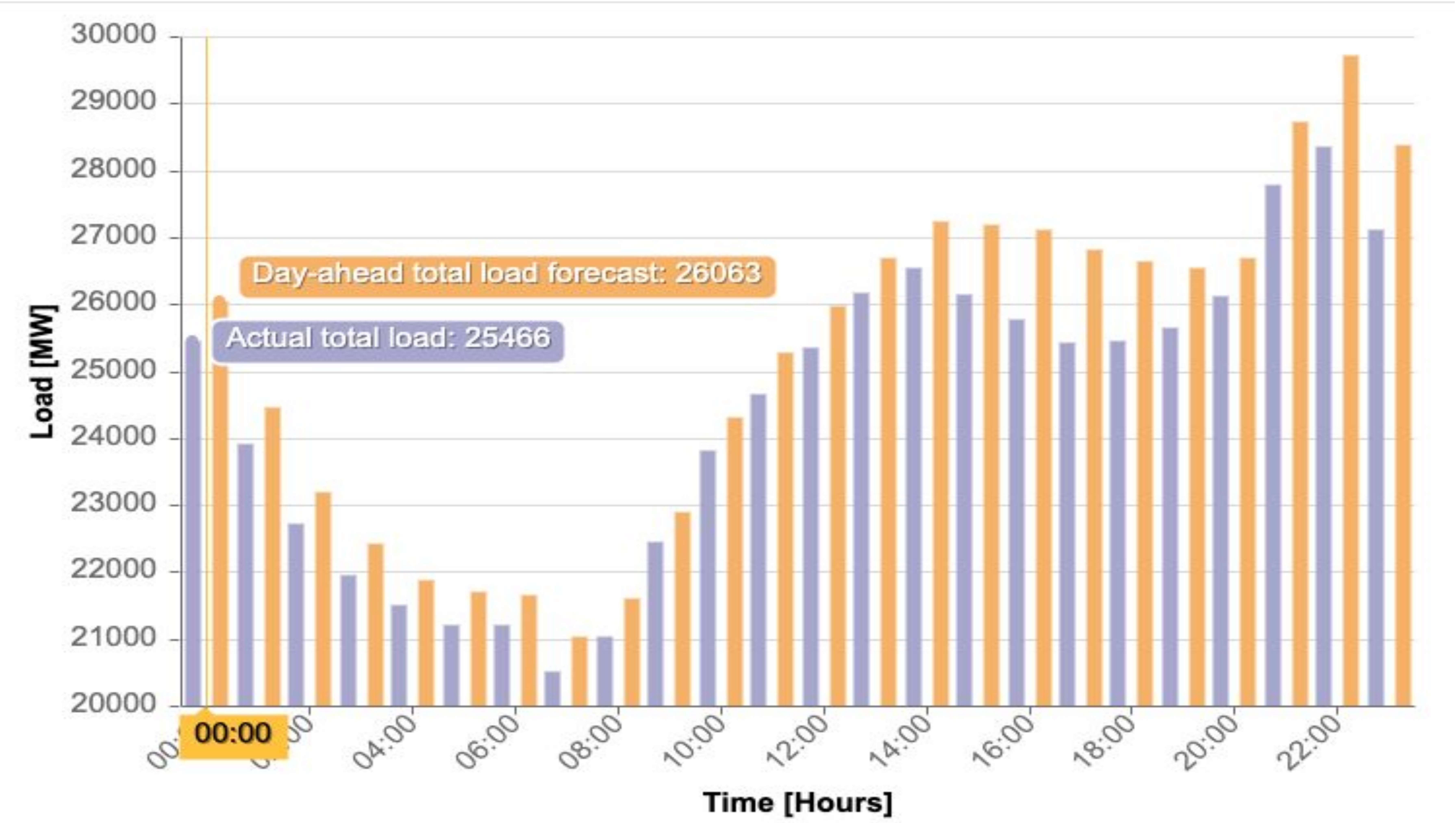
Short-term Energy Demand Forecasting

SARIMA, Prophet, and LSTMs

About me

- McMaster Engineering Physics
- Utrecht University Energy Science
- Spain National Curling Teams

What is short-term demand forecasting?



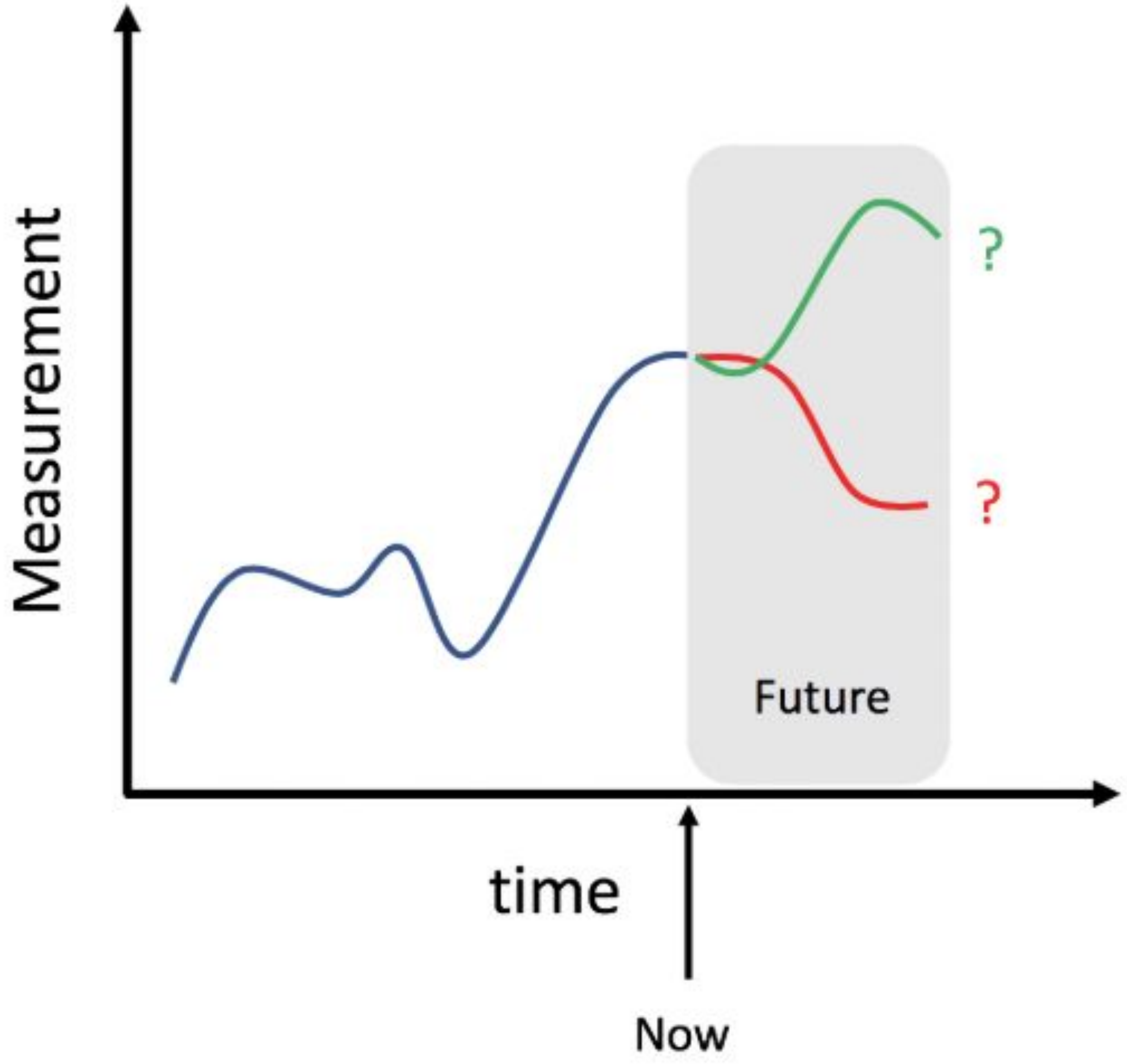
Each day predict the demand for the next 24 hours

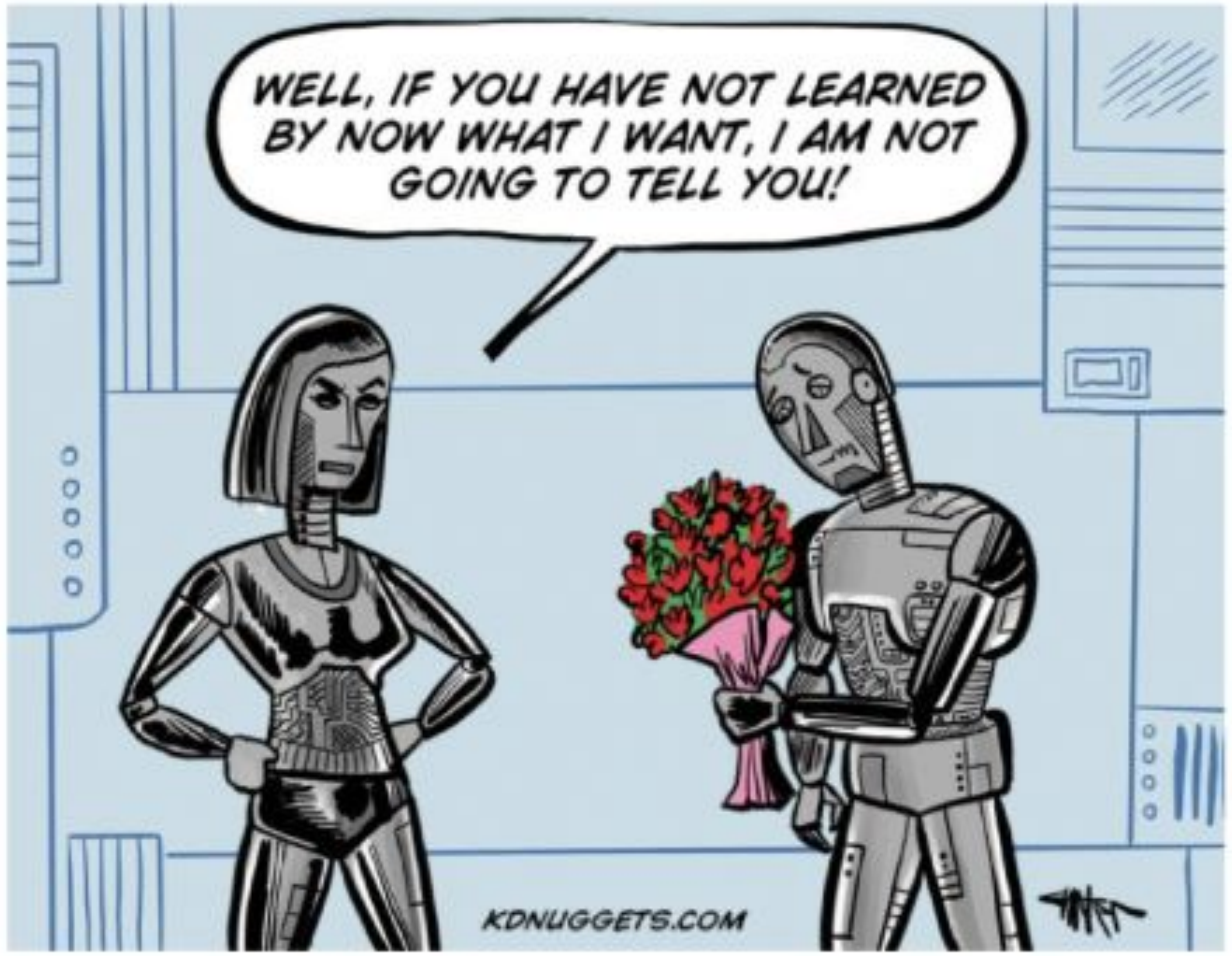
Why care about short-term demand forecasting?

BETTER 24 HOUR AHEAD FORECAST → REDUCED NEED FOR STANDBY RESERVE → MORE EFFICIENT ELECTRICAL GRID

- Plan to use renewable generation
- Plan for high load days
- Plan for low load days

Why do I care about energy forecasting





A difficult learning problem...

Why do I care about energy forecasting

... and highly relevant

State of the art approaches are using Neural Networks

energies

MDPI

Article

Short-Term Load Forecasting in Smart Grids: An Intelligent Modular Approach

Ashfaq Ahmad [1,*], Nadeem Javaid [2], Abdul Mateen [2], Muhammad Awais [2] and Zahoor Ali Khan [3]

1 School of Electrical Engineering and Computing, The University of Newcastle, Callaghan 2308, Australia
2 Department of Computer Science, COMSATS University Islamabad, Islamabad 44000, Pakistan; nadeemjavaid@comsats.edu.pk (N.J.); ammateen49@gmail.com (A.M.); amawais@hotmail.com (M.A.)
3 Computer Information Science, Higher Colleges of Technology, Fujairah 4114, UAE; zkhan1@hct.ac.ae
* Correspondence: ashfaqahmad@ieee.org; Tel.: +61-416-618-613

Received: 11 November 2018; Accepted: 1 January 2019; Published: 4 January 2019

check for updates

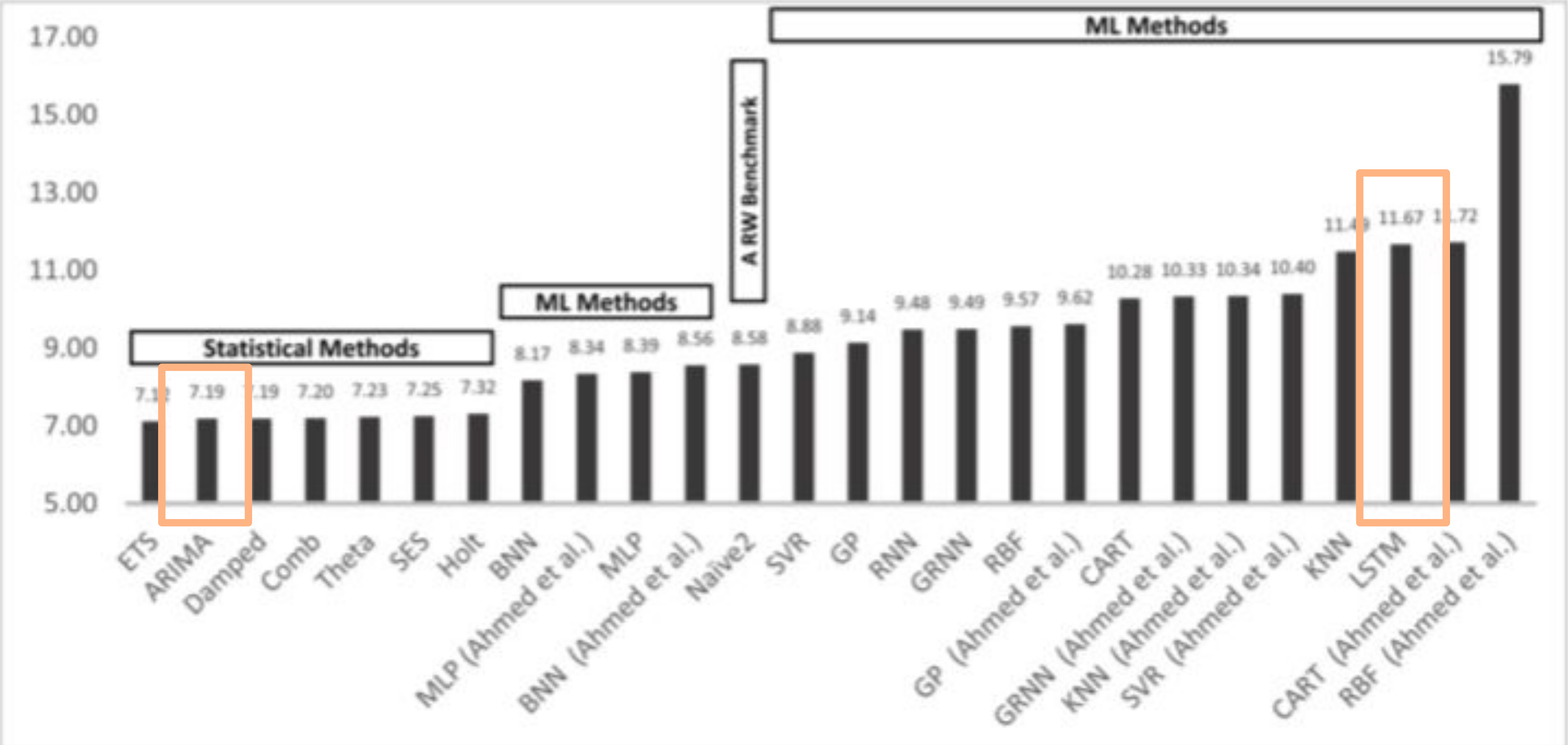
Abstract: Daily operations and planning in a smart grid require a day-ahead load forecasting of its customers. The accuracy of day-ahead load-forecasting models has a significant impact on many decisions such as scheduling of fuel purchases, system security assessment, economic scheduling of generating capacity, and planning for energy transactions. However, day-ahead load forecasting is a challenging task due to its dependence on external factors such as meteorological and exogenous variables. Furthermore, the existing day-ahead load-forecasting models enhance forecast accuracy by paying the cost of increased execution time. Aiming at improving the forecast accuracy while not paying the increased executions time cost, a hybrid artificial neural network-based day-ahead load-forecasting model for smart grids is proposed in this paper. The proposed forecasting model comprises three modules: (i) a pre-processing module; (ii) a forecast module; and (iii) an optimization

Average Percent Error 1-3%

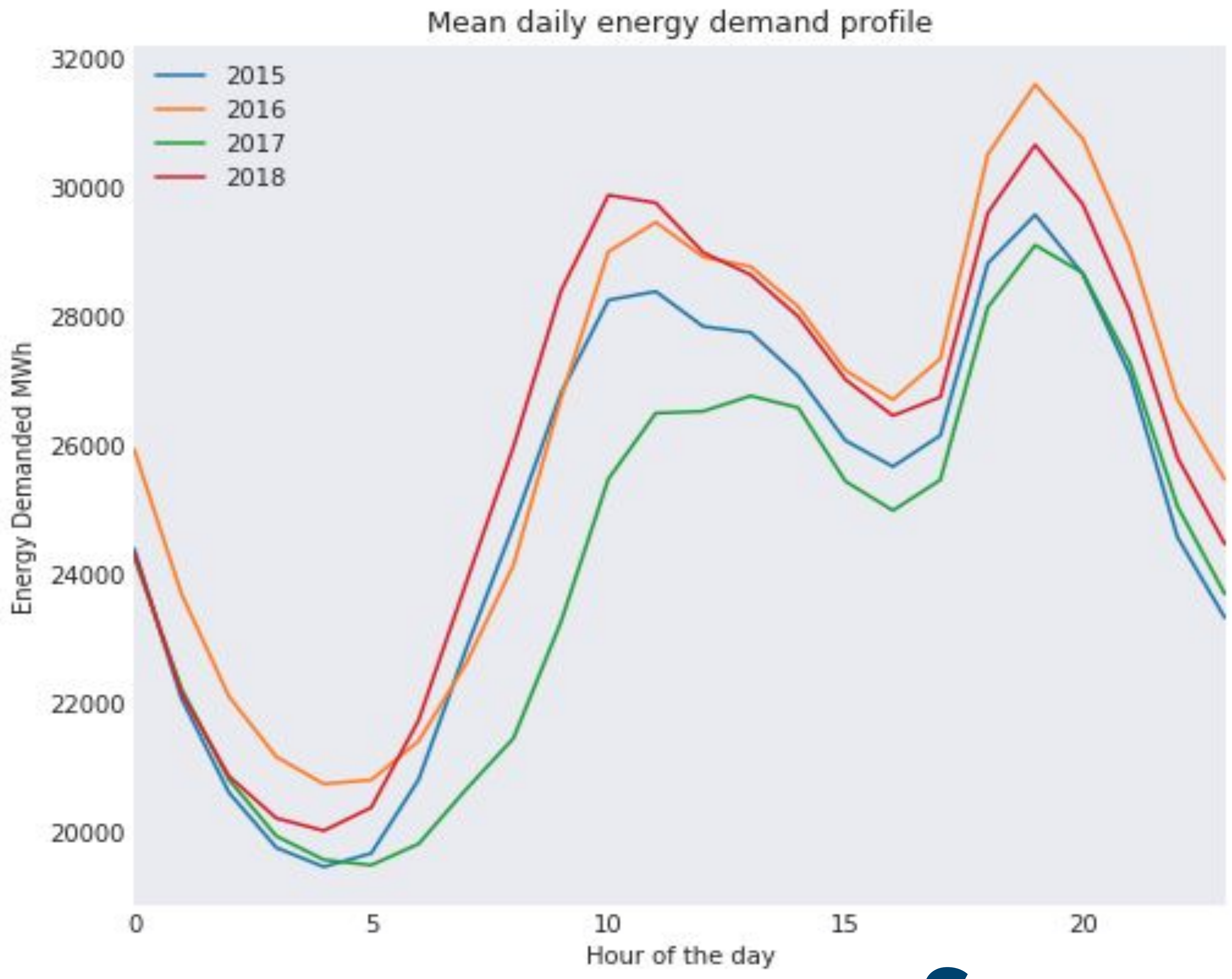
State of the art and classic forecasting tools

Statistical and Machine Learning forecasting methods: Concerns and ways forward

Spyros Makridakis[1], Evangelos Spiliotis[2]*, Vassilios Assimakopoulos[2]

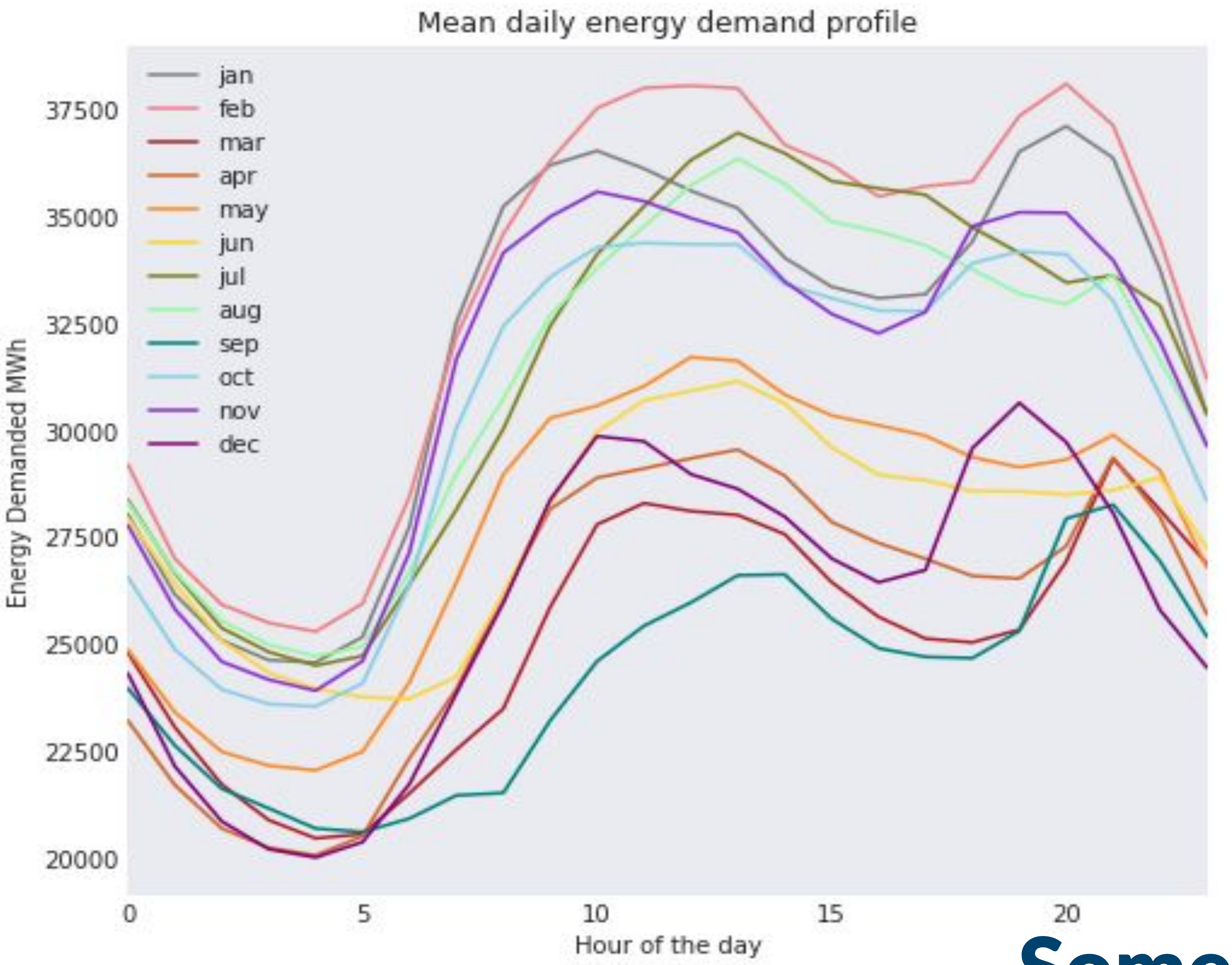


Energy demand changes through the day



Seems reasonable...

Also changes through the year...



Sometimes by a lot!

What do we use make forecasts?

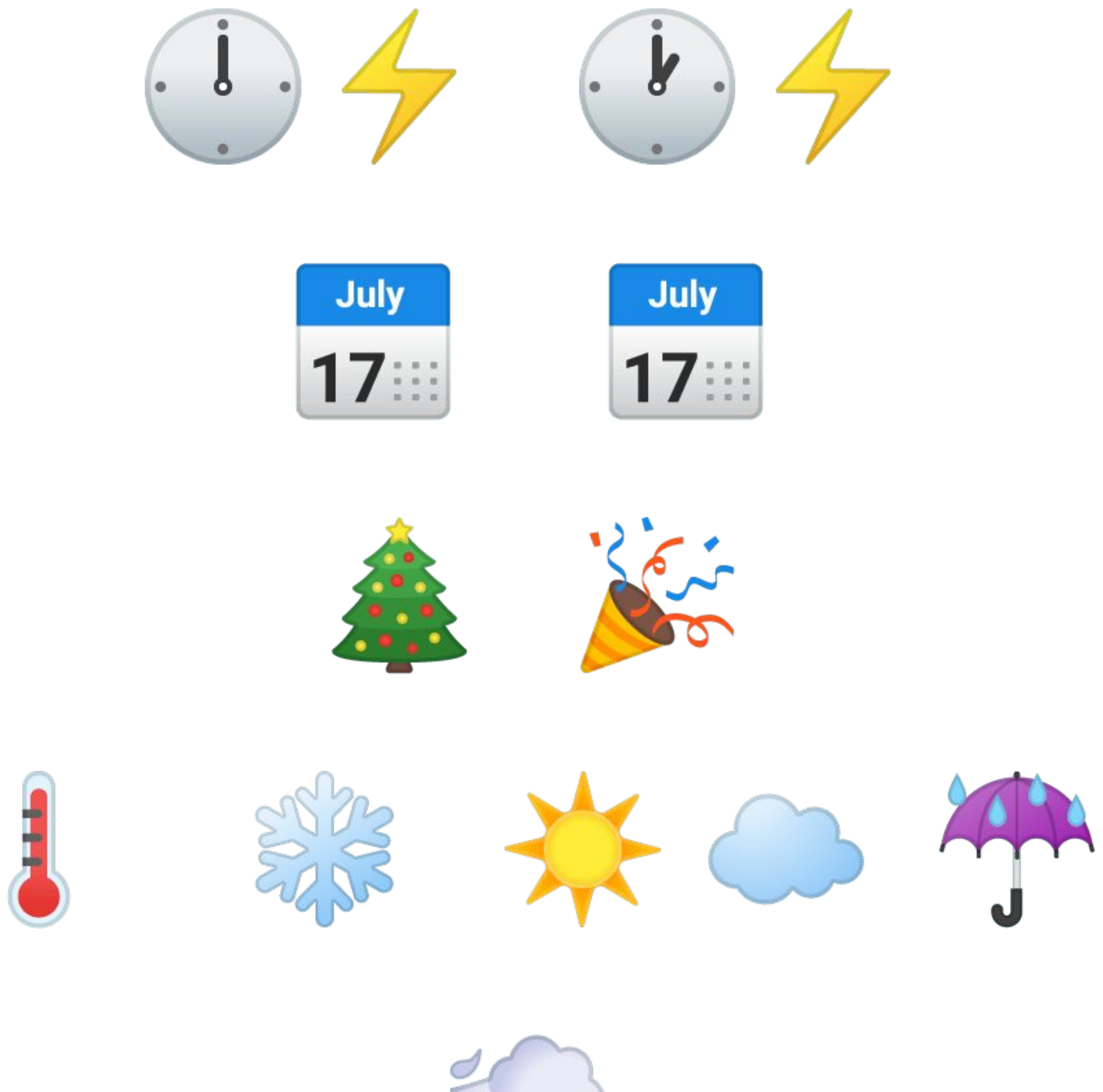
Models

SARIMA

Prophet

LSTM

Input Variables (Features)



SARIMA: Complex regression formula...

$$\underset{\substack{\uparrow \\ \left(\substack{\text{Non-seasonal} \\ \text{AR(1)}}\right)}}{(1-\phi_1 B)}\ \underset{\substack{\uparrow \\ \left(\substack{\text{Seasonal} \\ \text{AR(1)}}\right)}}{(1-\Phi_1 B^4)}\ \underset{\substack{\uparrow \\ \left(\substack{\text{Non-seasonal} \\ \text{difference}}\right)}}{(1-B)}\ \underset{\substack{\uparrow \\ \left(\substack{\text{Seasonal} \\ \text{difference}}\right)}}{(1-B^4)}y_t \;=\; \underset{\substack{\uparrow \\ \left(\substack{\text{Non-seasonal} \\ \text{MA(1)}}\right)}}{(1+\theta_1 B)}\ \underset{\substack{\uparrow \\ \left(\substack{\text{Seasonal} \\ \text{MA(1)}}\right)}}{(1+\Theta_1 B^4)}e_t.$$

Models trend and a seasonal repetitions

$$\text{ARIMA} \quad \underbrace{(p, d, q)}_{\substack{\uparrow \\ \left(\begin{array}{l}\text{Non-seasonal part} \\ \text{of the model}\end{array}\right)}} \quad \underbrace{(P, D, Q)_m}_{\substack{\uparrow \\ \left(\begin{array}{l}\text{Seasonal part} \\ \text{of the model}\end{array}\right)}}$$

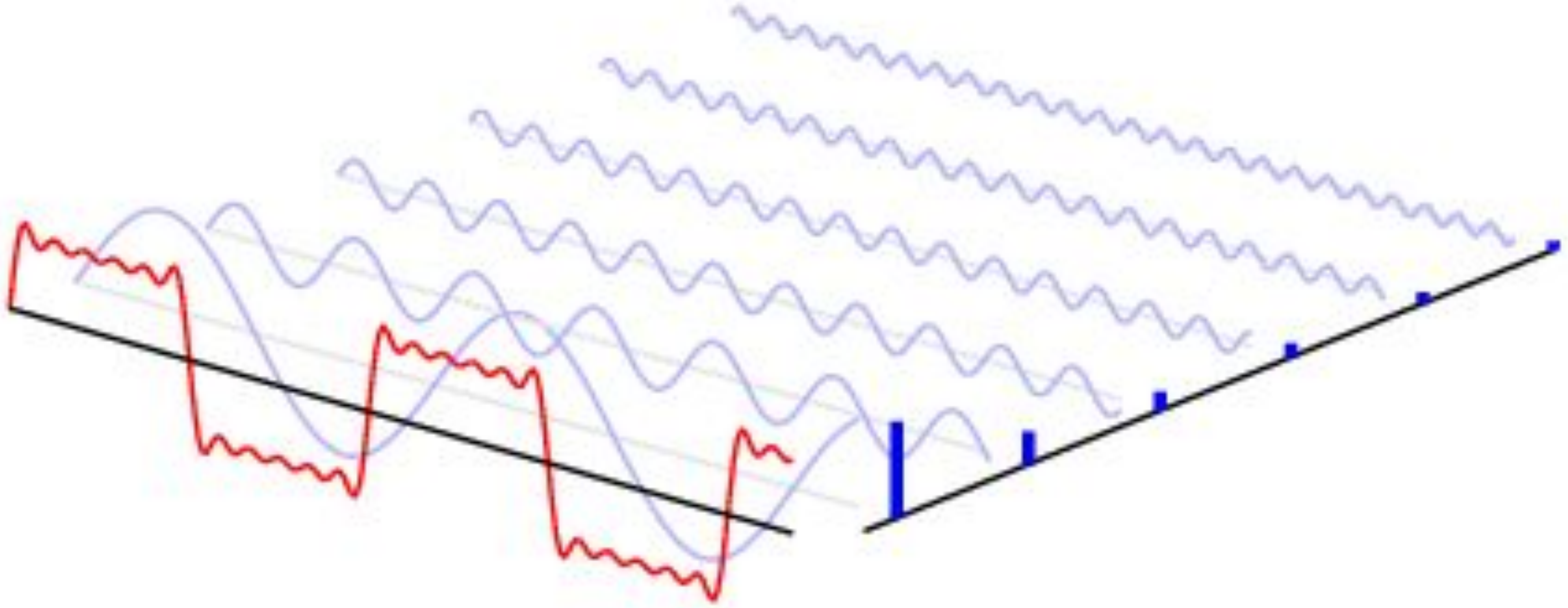
Fast general additive model

PROPHET

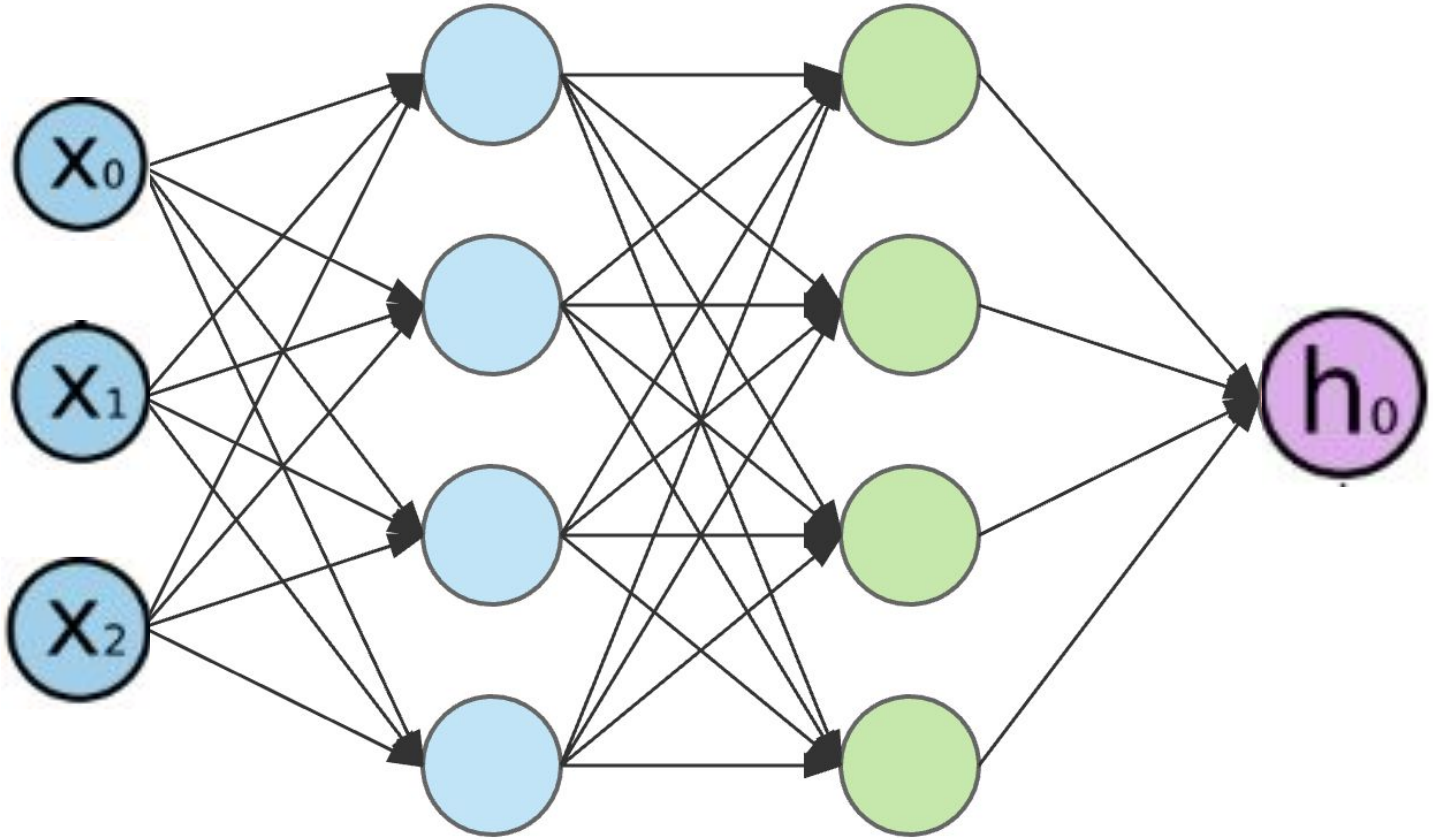
Forecasting at scale.

In use for capacity planning, goal setting, anomaly detection...

Yearly + monthly + daily patterns = Forecast



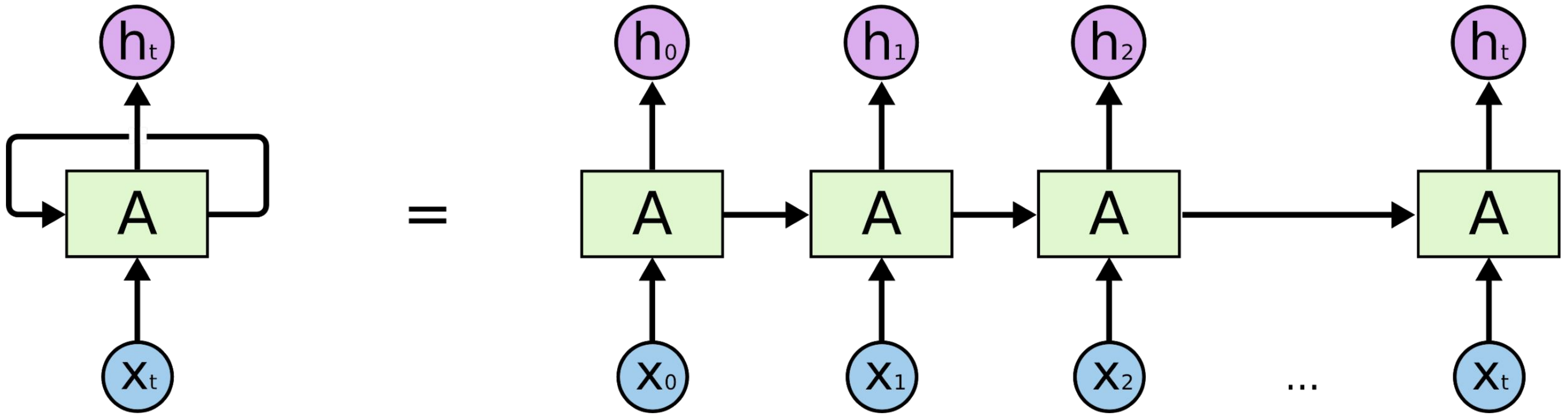
Classic neural networks process a sequence directly



Good at memorizing patterns...

Recurrent Neural Network

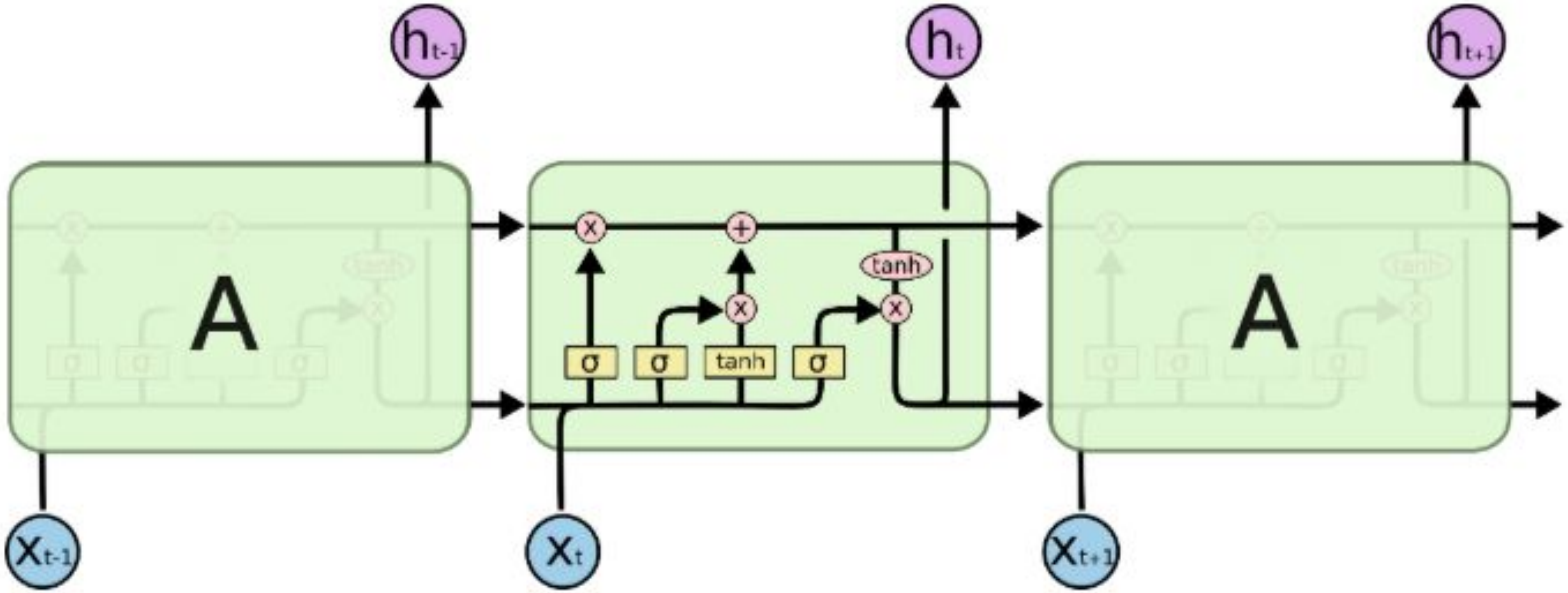
Good at finding relationships between inputs...



But bad at finding long term dependencies

Long-Short Term Memory (LSTM)

Able to 'remember' important parts of a sequence



Validating predictions with walk forward (backtesting)

Validating predictions with walk forward (backtesting)

Validating predictions with walk forward (backtesting)

Validating predictions with walk forward (backtesting)

Validating predictions with walk forward (backtesting)

Average how wrong we are at each cycle

SARIMA found the general shape of the day

Configuring walk forward was a little slow...

```
[Parallel(n_jobs=8)]: Using backend MultiprocessingBackend with 8 concurrent workers.

 Model [24, 0, 0, 1, 1, 1, 24]: 4241.718928386647
 Model [24, 0, 0, 2, 1, 1, 24]: 4003.5593081976613
 Model [24, 0, 0, 1, 2, 1, 24]: 4480.561686881436

[Parallel(n_jobs=8)]: Done   3 out of  16 | elapsed: 38.8min remaining: 168.3min

 Model [24, 0, 0, 1, 1, 1, 48]: 3604.430050456027
 Model [48, 0, 0, 1, 1, 1, 24]: 4177.370619587565

[Parallel(n_jobs=8)]: Done   5 out of  16 | elapsed: 82.1min remaining: 180.7min

 Model [24, 0, 0, 2, 2, 1, 24]: 5026.634648452434
 Model [48, 0, 0, 1, 2, 1, 24]: 4715.07641950746

[Parallel(n_jobs=8)]: Done   7 out of  16 | elapsed: 182.4min remaining: 234.5min

 Model [48, 0, 0, 2, 1, 1, 24]: 3844.476546036894
 Model [24, 0, 0, 2, 1, 1, 48]: 3569.2822786280876

[Parallel(n_jobs=8)]: Done   9 out of  16 | elapsed: 202.8min remaining: 157.8min

 Model [24, 0, 0, 1, 2, 1, 48]: 4073.1733620231066
 Model [48, 0, 0, 1, 1, 1, 48]: 3866.7211928488273

[Parallel(n_jobs=8)]: Done  11 out of  16 | elapsed: 249.7min remaining: 113.5min

Bad config[48, 0, 0, 2, 2, 1, 48]
 Model [48, 0, 0, 2, 2, 1, 24]: 4523.992357046215

[Parallel(n_jobs=8)]: Done  13 out of  16 | elapsed: 341.3min remaining: 78.8min

 Model [24, 0, 0, 2, 2, 1, 48]: 3632.000559581665
 Model [48, 0, 0, 1, 2, 1, 48]: 4259.056781541469
```

```
[Parallel(n_jobs=8)]: Done  16 out of  16 | elapsed: 456.9min finished

CPU times: user 15.3 s, sys: 5.83 s, total: 21.1 s
Wall time: 7h 36min 56s
```

No problem Google has fast computers...

```
[Parallel(n_jobs=8)]: Using backend MultiprocessingBackend with 8 concurrent workers.
 Model [24, 0, 0, 1, 1, 1, 24]: 4241.718928386647
 Model [24, 0, 0, 2, 1, 1, 24]: 4003.5593081976613
 Model [24, 0, 0, 1, 2, 1, 24]: 4480.561686881436
[Parallel(n_jobs=8)]: Done   3 out of  16 | elapsed: 38.8min remaining: 168.3min
 Model [24, 0, 0, 1, 1, 1, 48]: 3604.430050456027
 Model [48, 0, 0, 1, 1, 1, 24]: 4177.370619587565
[Parallel(n_jobs=8)]: Done   5 out of  16 | elapsed: 82.1min remaining: 180.7min
 Model [24, 0, 0, 2, 2, 1, 24]: 5026.634648452434
 Model [48, 0, 0, 1, 2, 1, 24]: 4715.07641950746
[Parallel(n_jobs=8)]: Done   7 out of  16 | elapsed: 182.4min remaining: 234.5min
 Model [48, 0, 0, 2, 1, 1, 24]: 3844.476546036894
 Model [24, 0, 0, 2, 1, 1, 48]: 3569.2822786280876
[Parallel(n_jobs=8)]: Done   9 out of  16 | elapsed: 202.8min remaining: 157.8min
 Model [24, 0, 0, 1, 2, 1, 48]: 4073.1733620231066
 Model [48, 0, 0, 1, 1, 1, 48]: 3866.7211928488273
[Parallel(n_jobs=8)]: Done  11 out of  16 | elapsed: 249.7min remaining: 113.5min
Bad config[48, 0, 0, 2, 2, 1, 48]
 Model [48, 0, 0, 2, 2, 1, 24]: 4523.992357046215
[Parallel(n_jobs=8)]: Done  13 out of  16 | elapsed: 341.3min remaining: 78.8min
 Model [24, 0, 0, 2, 2, 1, 48]: 3632.000559581665
 Model [48, 0, 0, 1, 2, 1, 48]: 4259.056781541469
```

Machine configuration

Machine family

General-purpose | Memory-optimised

Machine types for common workloads, optimised for cost and flexibility

Generation

Second

Powered by Cascade Lake CPU platform

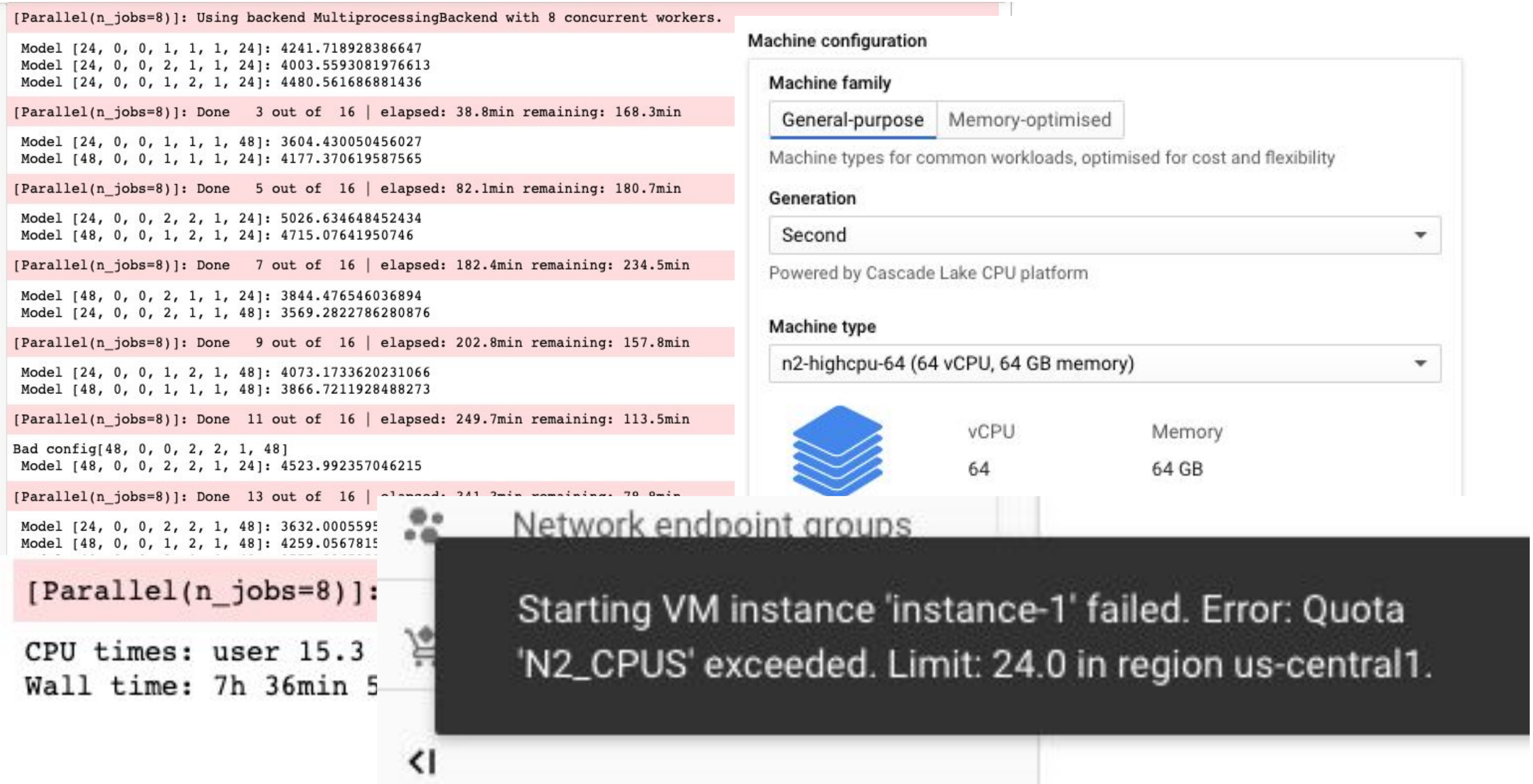
Machine type

n2-highcpu-64 (64 vCPU, 64 GB memory)

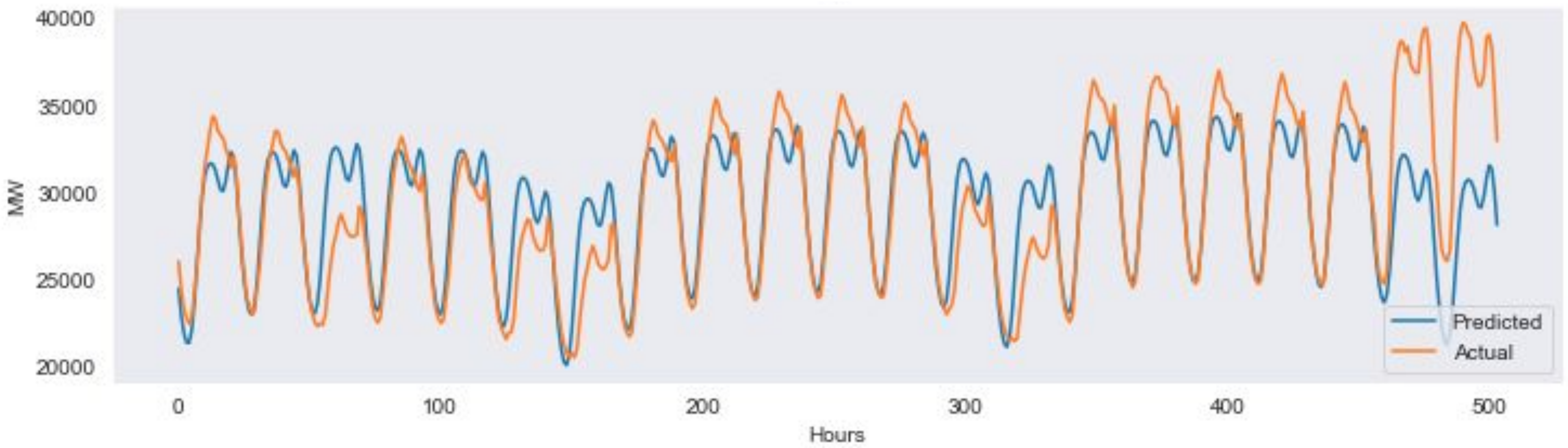
vCPU	Memory
64	64 GB

```
[Parallel(n_jobs=8)]: Done  16 out of  16 | elapsed: 456.9min finished
CPU times: user 15.3 s, sys: 5.83 s, total: 21.1 s
Wall time: 7h 36min 56s
```

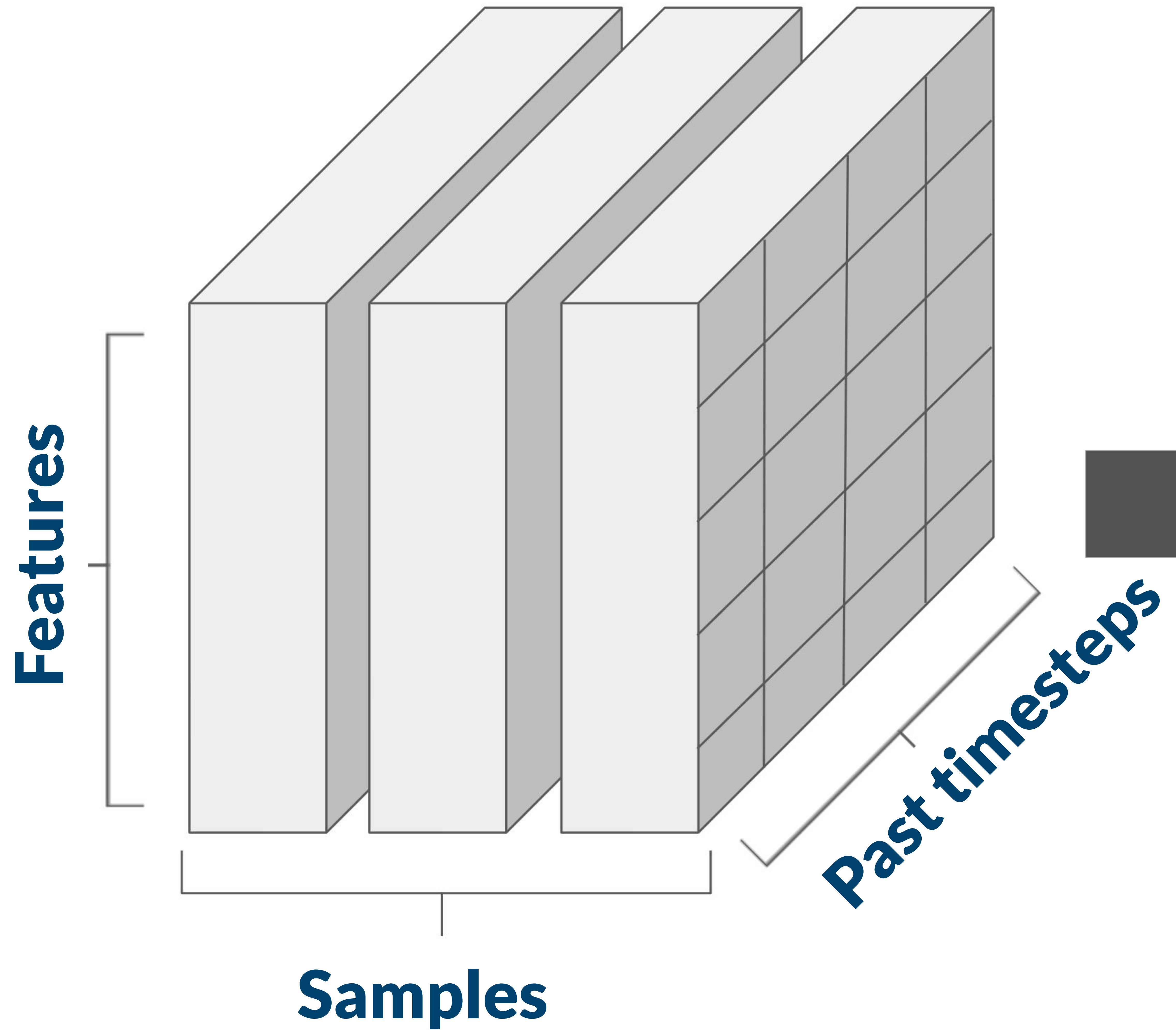
Just don’t try to use them with your free credits



Prophet captured weekly changes well



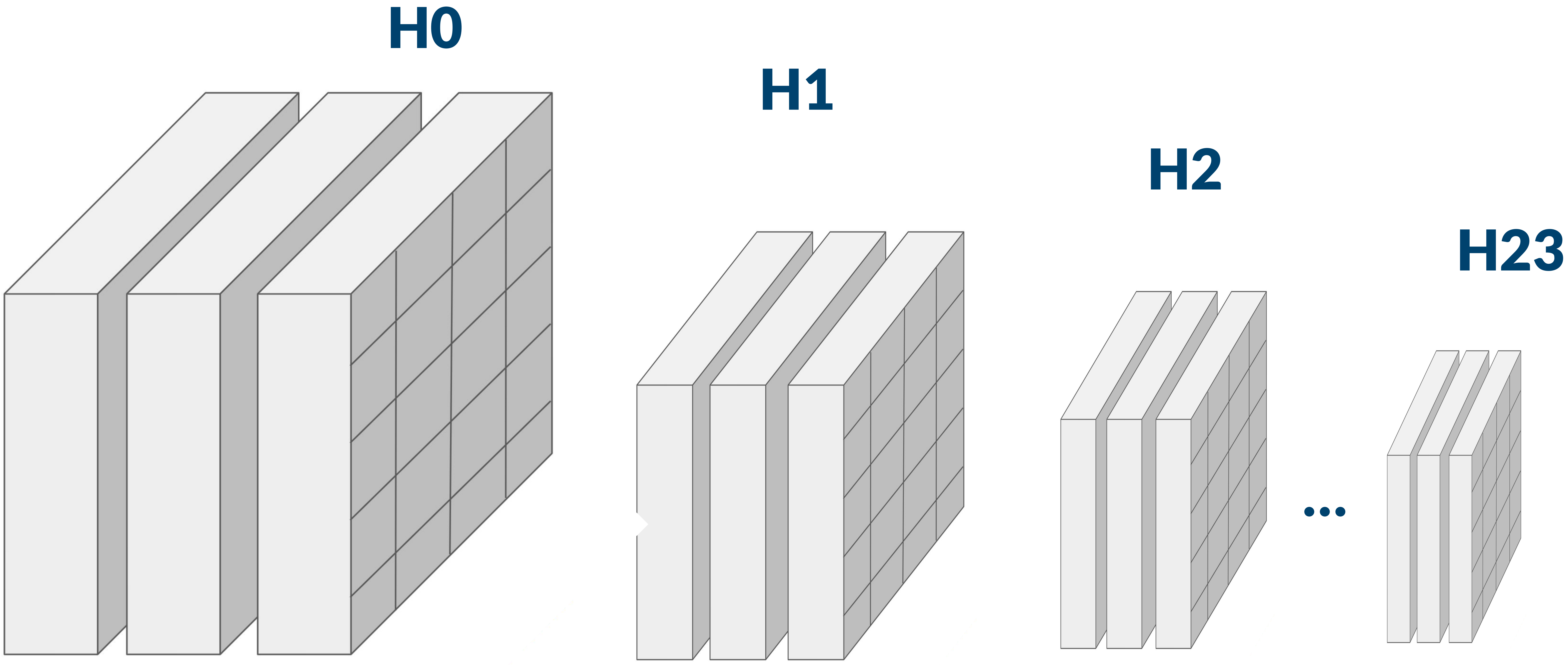
Typical data structure for an LSTM



Output is sequence of next 24 hour energy demand

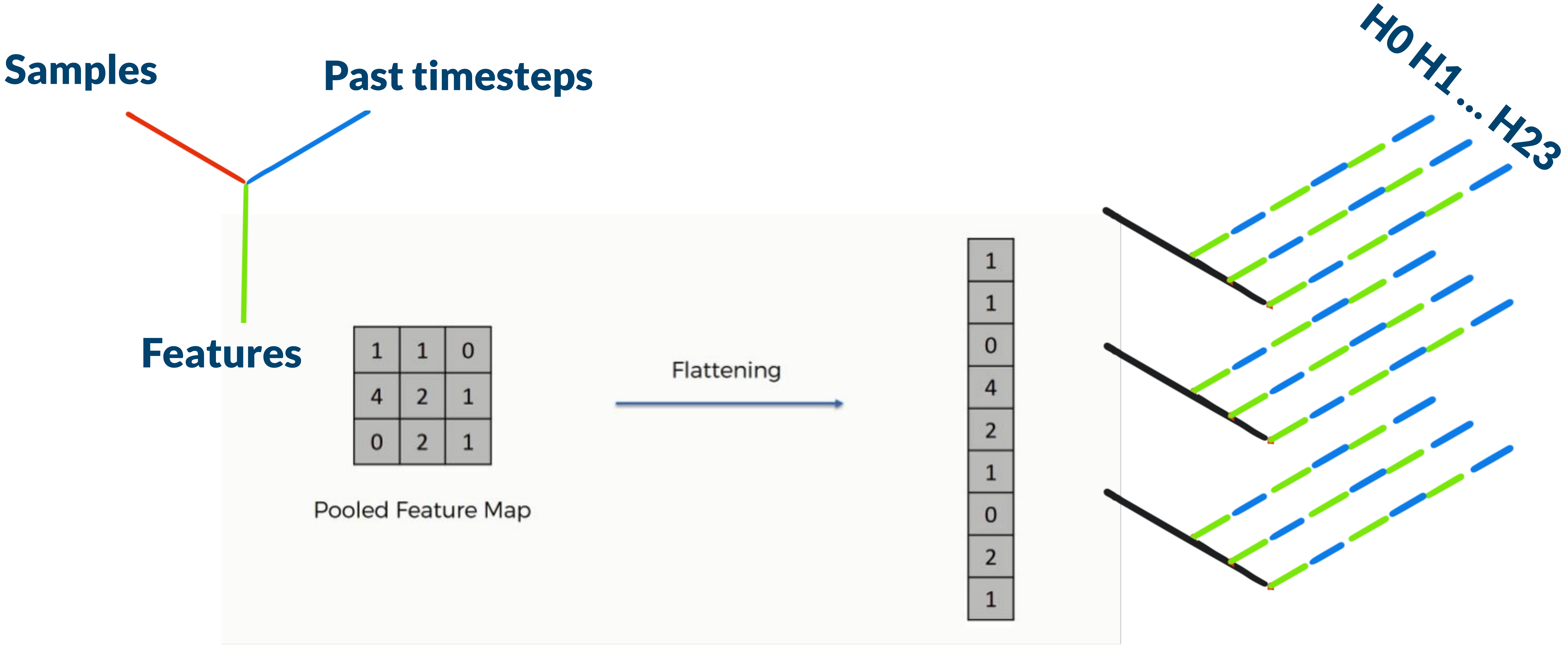
3	141	140	6	7	137	13
H0	H1	H2	H3	H4 ...		

Repeat for every hour in the day 3D → 4D



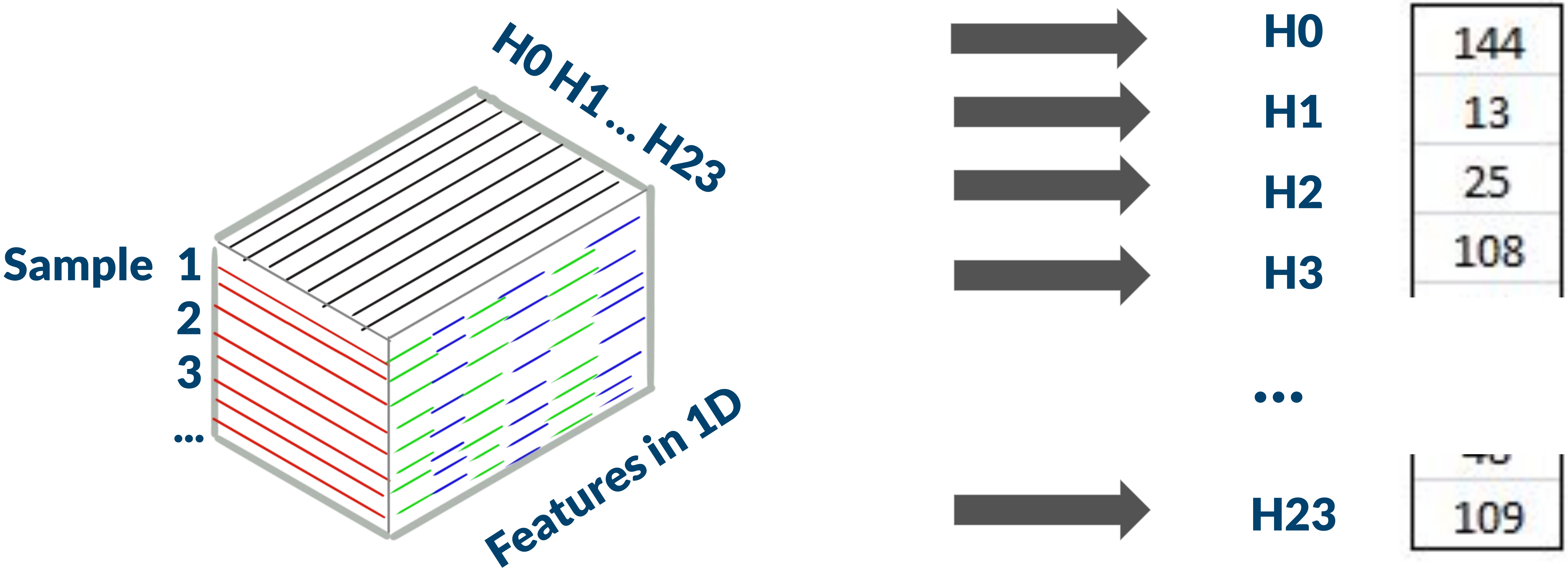
Restructuring from 4D back to 3D

Flatten Time Stepped Features into row vectors

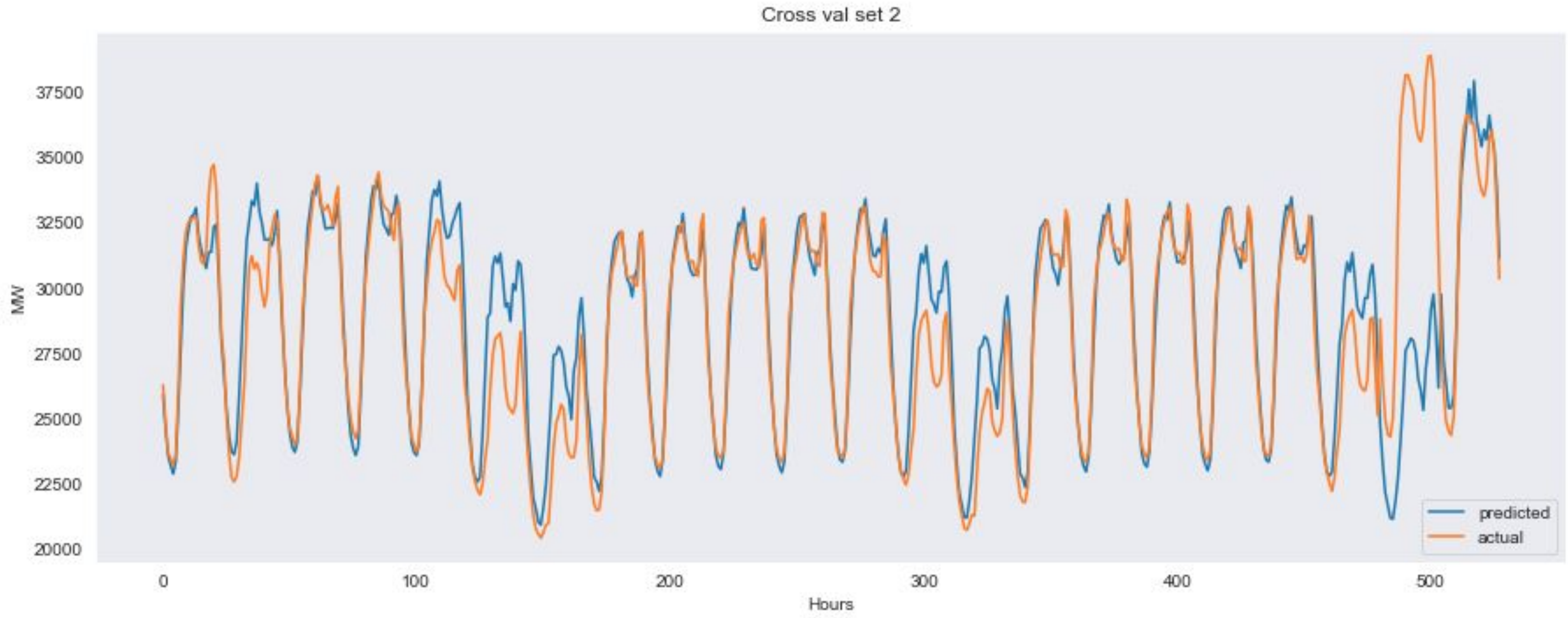


Exploiting the recurrent capacity of LSTM

Combine features into one long input for each hour of the day

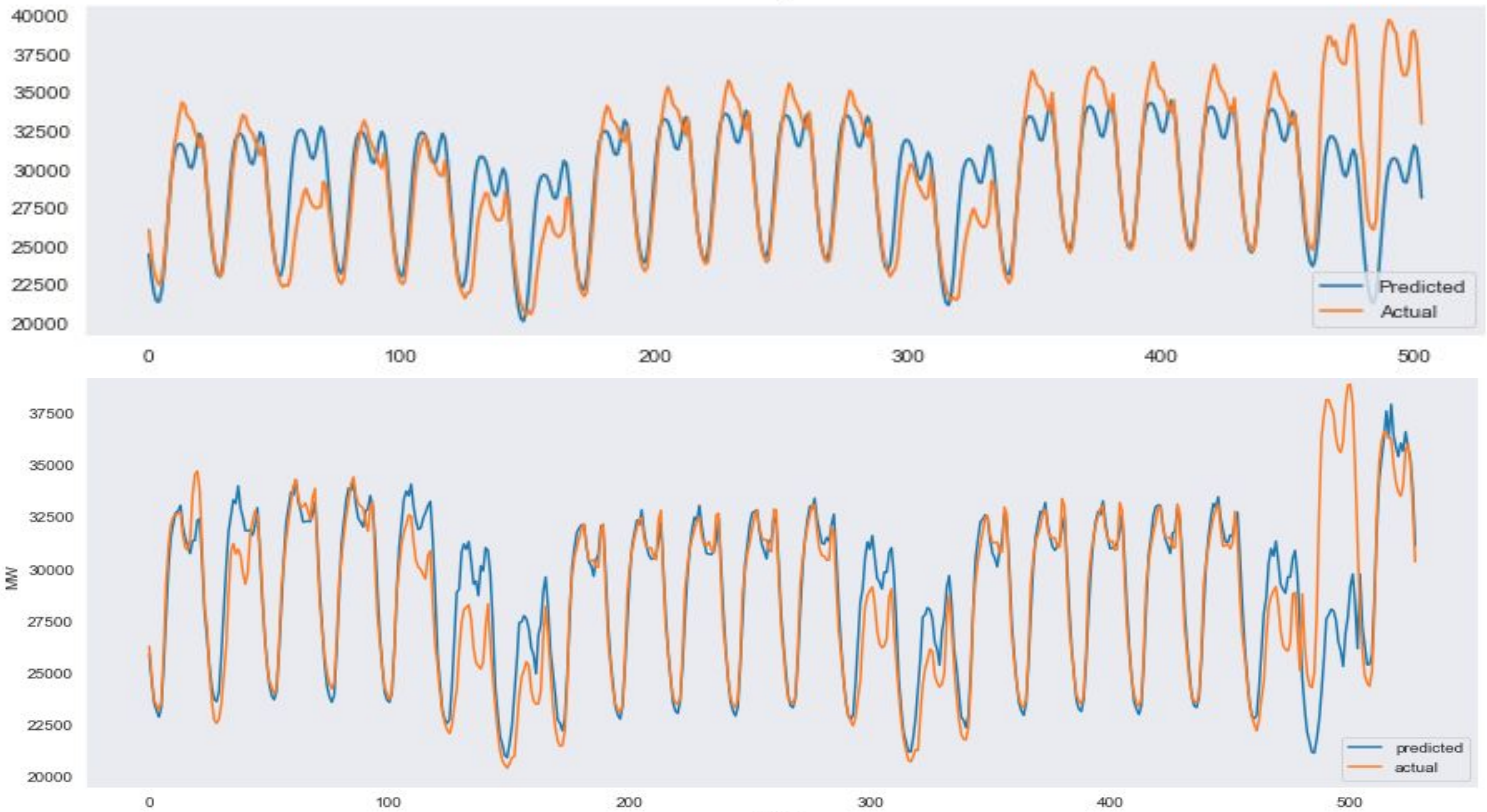


LSTM



Cross val set 2
37500
35000
32500
30000
27500
25000
22500
20000
MW
0
100
200
300
400
500
Hours
predicted
actual

Comparing the models



Prophet

LSTM

Comparing model errors

Model	Mean Absolute Error (MW)	Mean Absolute Percent Error (%)
SARIMA	2821	9.81%
Prophet	2116	7.54%
LSTM	1901	6.61%
Spain’s TSO	350	1.5%

Spain’s current forecast error

350 MW

1-2%

Two week project vs. team of experts

3.2X

Thank you!