Energy Consumption Forecasting

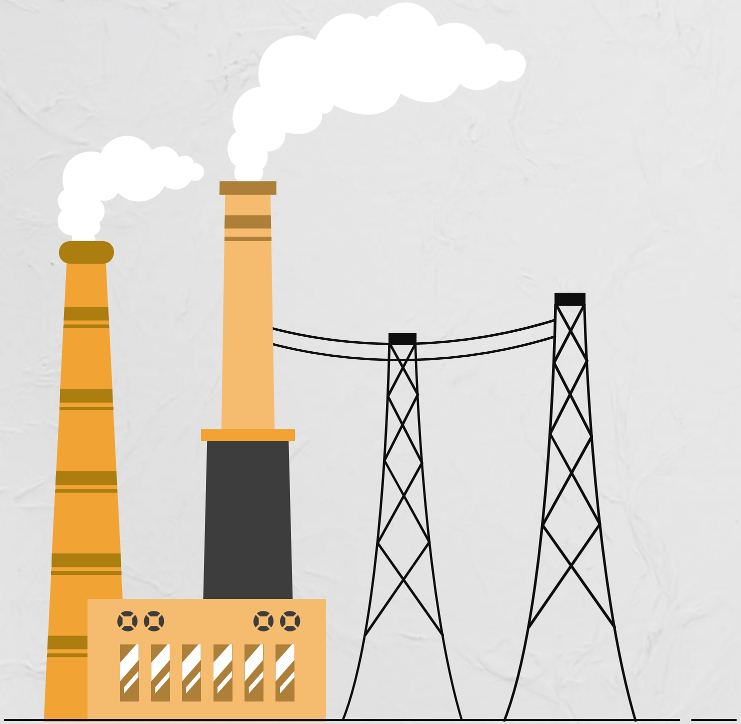
LSTM Model

Group L
Adrià Soria
Judit Viladecans
Paula Ceprián

Universitat Pompeu Fabra Barcelona

Deep Learning 24-25
Final Project

Table of contents



01

Problem Description

Why we chose this project?

"Un apagón histórico: sin precedentes en España y entre los más masivos de Europa"

El País, 4th of May 2025

April 28th Blackout

When the Grid Fails to See What’s Coming

On April 28th, 2025, Spain experienced a widespread power outage affecting homes, businesses, and essential services.

Triggered by an unexpected surge in electricity demand.

Grid operators were reactive, not proactive and they responded after the system was overloaded.

What Else Can This Model Do?

Beyond Forecasting

Energy Efficiency

Shift consumption to cheaper time slots

Preventive Maintenance

Detect faulty appliances via unusual patterns

Renewable Energy Optimization

Decide when to store or export energy

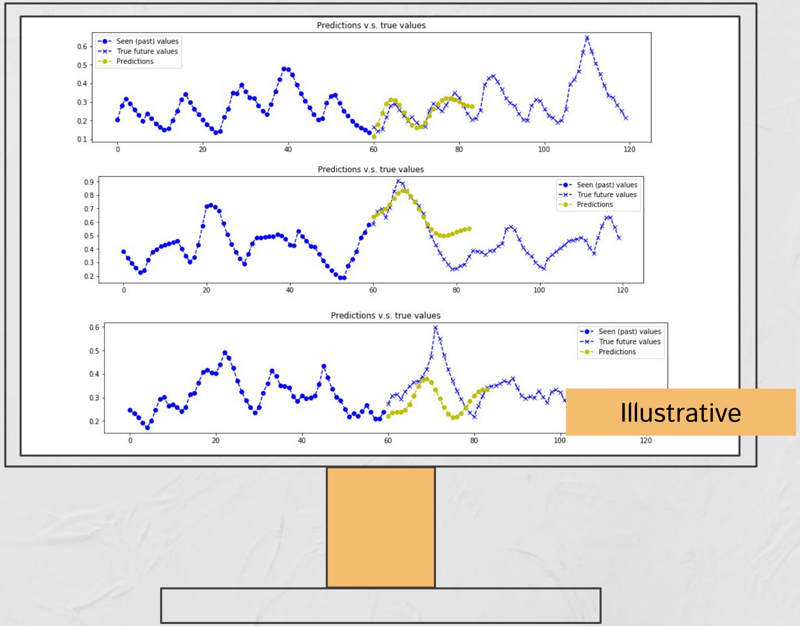
Energy Auditing

Understand and improve household energy habits

Our objective

Predict to Prevent

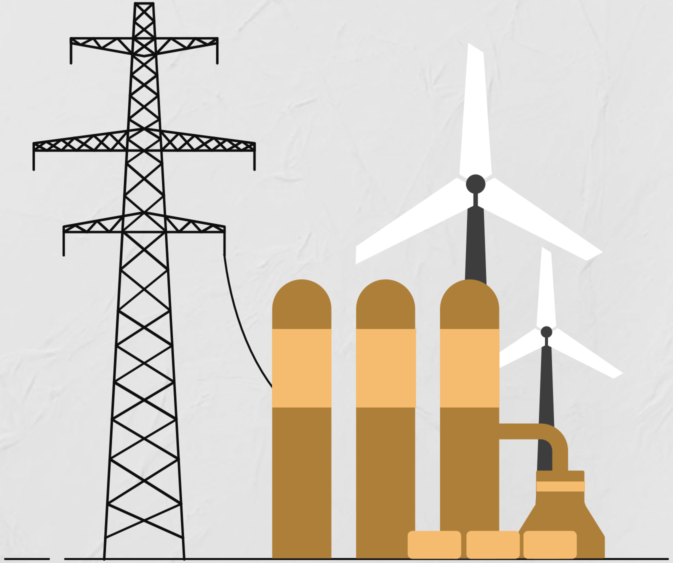
To forecast a household’s real-time energy consumption using last hours of reactive power, voltage, intensity, and time-based patterns



02

Data Analysis

Data Gathering, Exploration and Insights



Our Dataset

This dataset contains **2075259 measurements** gathered in a house located in Sceaux (7 km of Paris, France) between December 2006 and November 2010 (47 months)

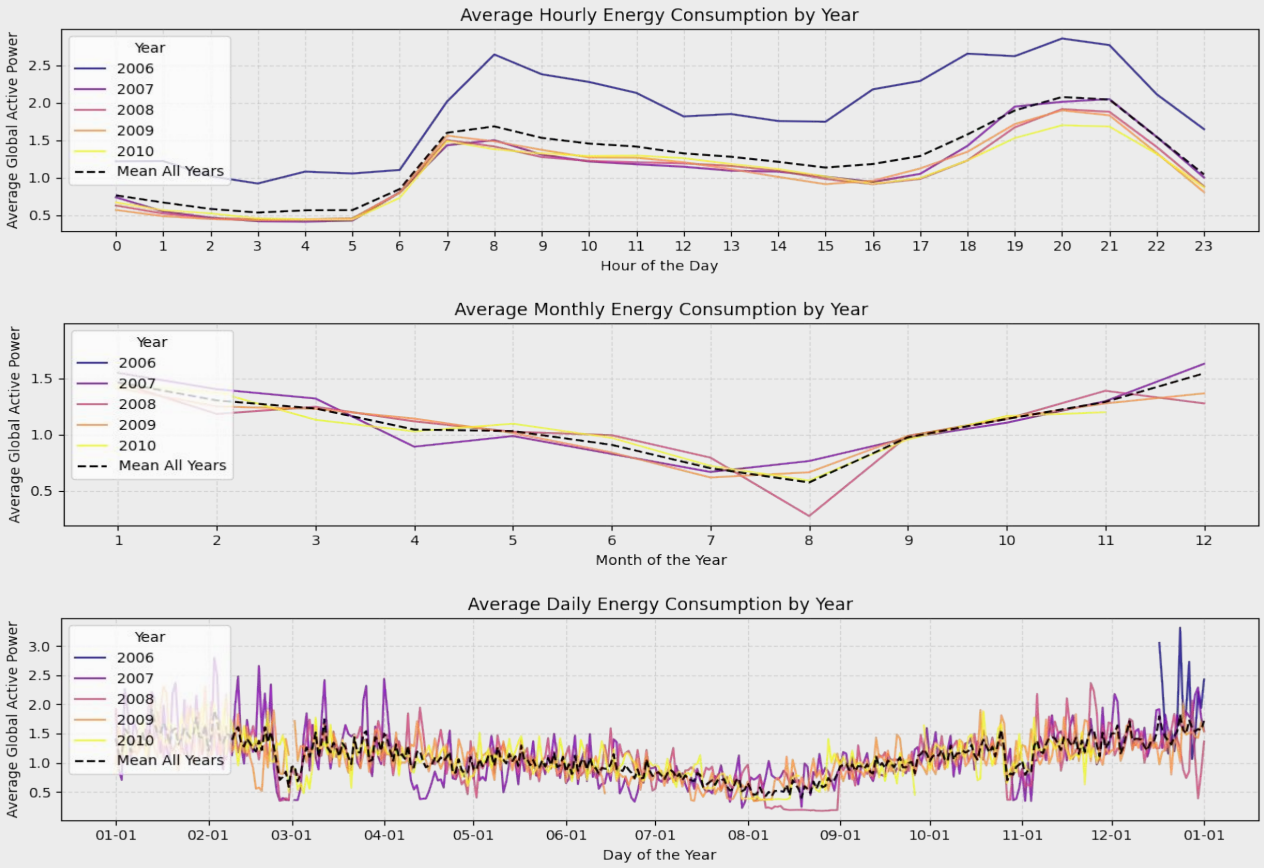
Date	Date in format dd/mm/yyyy
Time	Time in format hh:mm:ss
Global Active Power	Household global minute-averaged active power (in kilowatt)
Global Reactive Power	Household global minute-averaged reactive power (in kilowatt)
Voltage	Minute-averaged voltage (in volt)
Global Intensity	Household global minute-averaged current intensity (in ampere)
Sub-metering 1	Energy sub-metering No. 1 (in watt-hour of active energy) of the kitchen (dishwasher, oven, microwave).
Sub-metering 2	Energy sub-metering No. 2 (in watt-hour of active energy) of the laundry room (washing-machine, tumble-drier, refrigerator).
Sub-metering 3	Energy sub-metering No. 3 (in watt-hour of active energy) of an electric water-heater and an air-conditioner.

Our Dataset

	First Row	Second Row	Third Row	Fourth Row	Fifth Row
Date	2006-12-16	2006-12-16	2006-12-16	2006-12-16	2006-12-16
Time	17:25:00	17:26:00	17:27:00	17:28:00	17:29:00
Global Active Power	5.36	5.374	5.388	3.666	3.52
Global Reactive Power	0.436	0.498	0.502	0.528	0.522
Voltage	233.63	223.29	233.74	235.68	235.02
Global Intensity	23.0	23.0	23.0	15.8	15.0
Sub-metering 1	0.0	0.0	0.0	0.0	0.0
Sub-metering 2	1.0	2.0	1.0	1.0	2.0
Sub-metering 3	16.0	17.00	17.00	17.00	17.00

Date	Time	Global Active Power	Global Reactive Power	Voltage	Global Intensity	Sub Metering 1	Sub Metering 2	Sub Metering 3
2006-12-16 00:00:00	18:30:00	2.93	0.0	236.15	12.4	0.0	1.0	17.0
2006-12-16 00:00:00	18:31:00	2.912	0.05	235.81	12.4	0.0	1.0	17.0
2006-12-16 00:00:00	18:32:00	2.608	0.052	235.41	11.0	0.0	1.0	17.0
2006-12-16 00:00:00	18:33:00	2.714	0.162	234.82	11.6	0.0	0.0	17.0
2006-12-16 00:00:00	18:34:00	3.538	0.086	233.76	15.6	0.0	1.0	16.0
2006-12-16 00:00:00	18:35:00	6.072	0.0	232.48	26.4	0.0	27.0	17.0
2006-12-16 00:00:00	18:36:00	4.536	0.0	233.54	19.4	0.0	1.0	17.0
2006-12-16 00:00:00	18:37:00	4.408	0.0	232.32	18.8	0.0	1.0	16.0
2006-12-16 00:00:00	18:38:00	2.912	0.048	234.02	13.0	0.0	1.0	17.0
2006-12-16 00:00:00	18:39:00	2.326	0.054	234.76	9.8	0.0	1.0	17.0
2006-12-16 00:00:00	18:40:00	2.264	0.054	234.67	9.6	0.0	1.0	17.0
2006-12-16 00:00:00	18:41:00	2.27	0.054	235.27	9.6	0.0	1.0	17.0
2006-12-16 00:00:00	18:42:00	2.258	0.054	235.12	9.6	0.0	1.0	17.0
2006-12-16 00:00:00	18:43:00	2.188	0.068	235.8	9.2	0.0	1.0	17.0
2006-12-16 00:00:00	18:44:00	2.978	0.166	234.81	13.2	0.0	1.0	17.0
2006-12-16 00:00:00	18:45:00	4.2	0.174	234.38	17.8	0.0	1.0	17.0
2006-12-16 00:00:00	18:46:00	4.204	0.186	234.2	17.8	0.0	1.0	16.0
2006-12-16 00:00:00	18:47:00	4.218	0.178	233.98	18.0	0.0	1.0	17.0
2006-12-16 00:00:00	18:48:00	2.786	0.188	234.99	12.0	0.0	2.0	17.0
2006-12-16 00:00:00	18:49:00	2.54	0.088	234.67	10.8	0.0	4.0	17.0
2006-12-16 00:00:00	18:50:00	2.496	0.08	233.92	10.6	0.0	3.0	17.0
2006-12-16 00:00:00	18:51:00	2.336	0.07	233.51	10.0	0.0	1.0	16.0
2006-12-16 00:00:00	18:52:00	2.322	0.0	233.44	9.8	0.0	0.0	17.0
2006-12-16 00:00:00	18:53:00	2.448	0.0	233.64	10.6	0.0	1.0	17.0
2006-12-16 00:00:00	18:54:00	4.298	0.0	232.39	18.4	0.0	1.0	16.0
2006-12-16 00:00:00	18:55:00	4.23	0.09	232.25	18.2	0.0	1.0	17.0

Consumption Trends through the years



Data Processing and Cleaning

01

Fill missing values with a smoothed version of the consumption pattern from periods with a similar temporal behavior.

Fill missing values: Nulls

	Count(Nulls)
Date	0
Time	0
Global Active Power	0
Global Reactive Power	25979
Voltage	25979
Global Intensity	25979
Sub-metering 1	25979
Sub-metering 2	25979
Sub-metering 3	25979

25.979 nulls in all these variables… Why?

08-08-2008 < no info < 01-09-2008, which are 23 days and around 30.000 minutes.

No info → **Fill** with values from 08-2007

Data Processing and Cleaning

01

Fill missing values with a smoothed version of the consumption pattern from periods with a similar temporal behavior

02

Remove unnecessary columns (like sub-metering values) that do not affect our target variable and correct the format of the remaining features

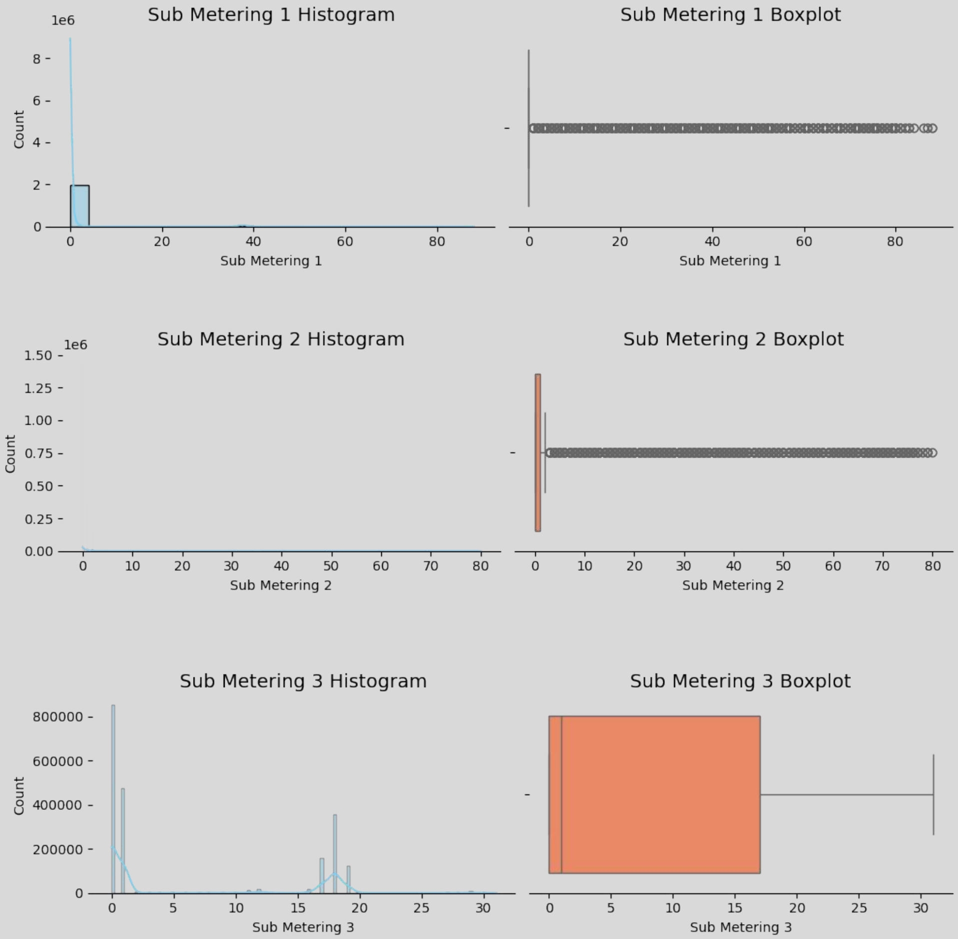
Why predicting Global Active Power and not Sub-Metering variables?

<u>Sub Metering 1</u> → NO INFO

<u>Sub Metering 2</u> → Lots of 0's, small values (1.0-5.0), and big outliers (28.0-37.0)

<u>Sub Metering 3</u> → Values are rather 0.0 (almost all) or between 17.00-19.00

Conclusion → **DROP** these columns.



Data Processing and Cleaning

01

Fill missing values with a smoothed version of the consumption pattern from periods with a similar temporal behavior.

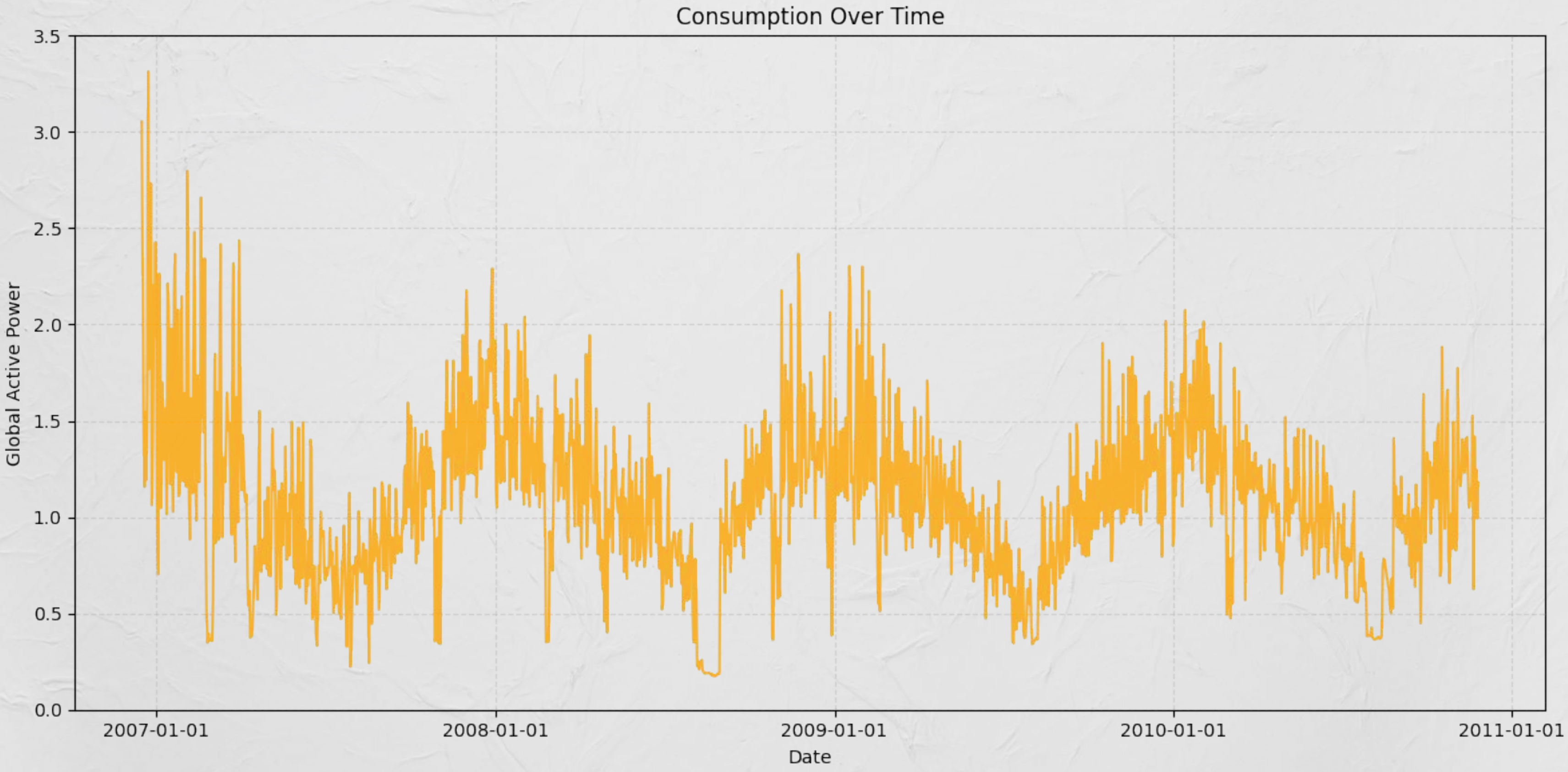
02

Remove unnecessary columns (like sub-metering values) that do not affect our target variable and correct the format of the remaining features

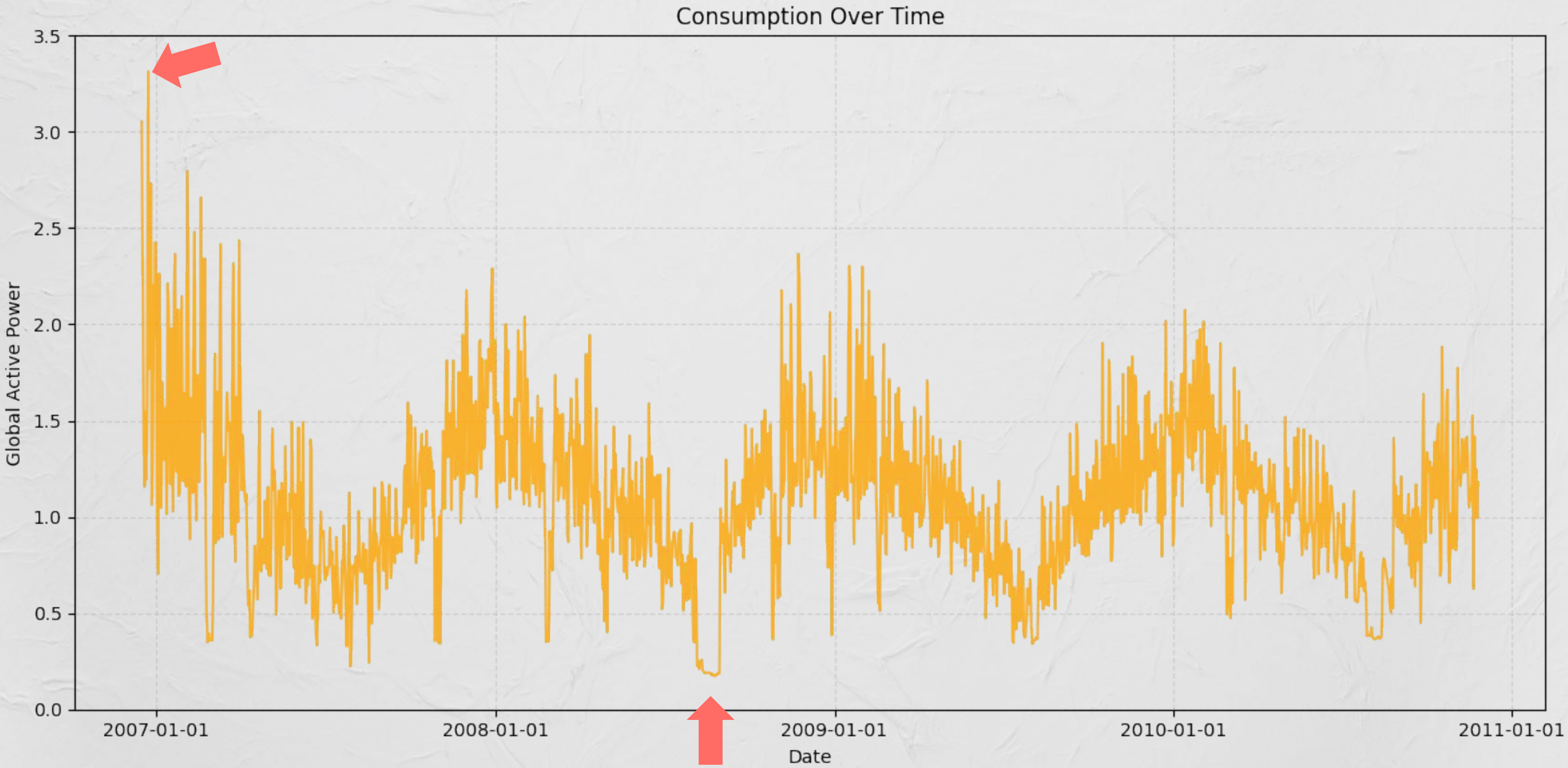
03

Smooth our data and delete rows with outliers to find general trends and seasonality, removing consumption picks that do not represent our data well

Raw Dataset

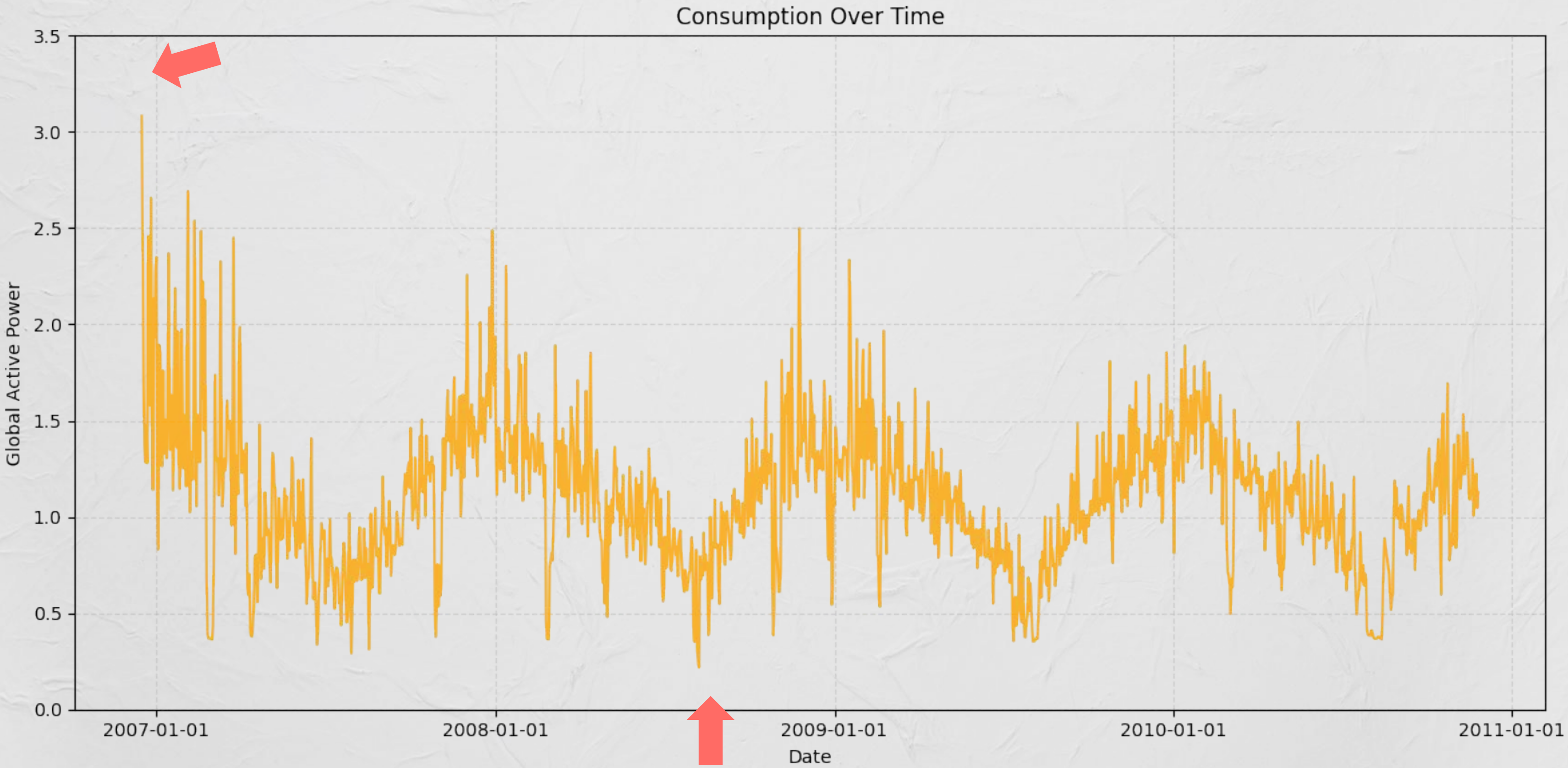

Consumption Over Time
Global Active Power
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
2007-01-01
2008-01-01
2009-01-01
2010-01-01
2011-01-01
Date

Raw Dataset



Consumption Over Time
Global Active Power
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
2007-01-01
2008-01-01
2009-01-01
2010-01-01
2011-01-01
Date

Cleaned Dataset



03

Model Architecture

Different Implementations of our LSTM

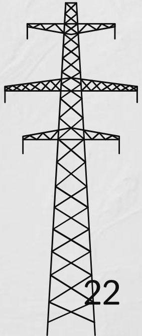
Metrics to measure performance

RMSE (Root Mean Squared Error):
How far our predictions are (on average) from the true values.
Lower is better.

MAPE (Mean Absolute Percentage Error):
Error as a percentage.
Lower is better.

R^2 (Coefficient of Determination):
How well the model captures the variability in the data.
Higher is better.

Accuracy with relative Tolerance:
How many predictions are “good enough” within a defined margin.
Higher is better.



Baseline Model

The Model:

LSTM+ReLU+LSTM+ReLU

Its HyperParameters:

SEQ. LEN = 100

FUTURE_PERIOD_PREDICT = 1

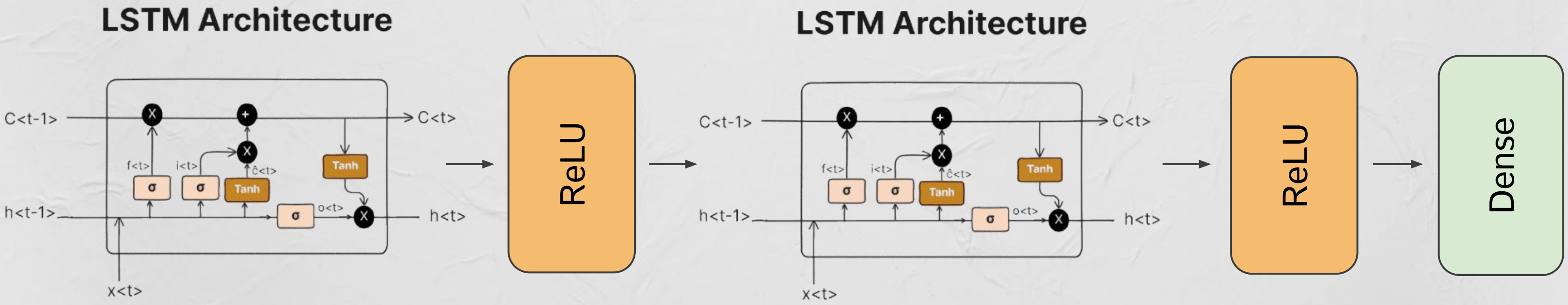
EPOCHS = 30

BATCH SIZE = 64

Other Parameters:

Smoothing df = 0.2

Factor for outliers = 3



Baseline Model

Performance:

Seq. len	Future predict	Epochs	Batch	RMSE	R^2	MAPE	Accuracy
100	6	30	64	0.3349	0.4748	23.75%	29.28%

- Sequence Length: how long of a preceding sequence to collect (in 1 hour minutes)
- Future Period Predict: how far into the future are we trying to predict (in hours)
- Number of Epochs
- Batch Size

How can we improve Accuracy?

Adapt hyperparameters and other parameters to values with more "sense" for our topic.

First Approach: Hyperparameter Tuning

Seq. len	Future predict	Epochs	Batch	RMSE	R^2	MAPE	Accuracy
50	**1**	**20**	**64**	**0.1068**	**0.9463**	**7.04%**	**75.18%**
50	**1**	**50**	**64**	**0.1048**	**0.9483**	**6.66%**	**76.28%**
50	3	20	64	0.2306	0.7500	16.93%	40.27%
50	3	50	64	0.2366	0.7367	15.81%	42.96%
100	**1**	**20**	**64**	**0.1080**	**0.9454**	**6.88%**	**76.14%**
100	**1**	**50**	**64**	**0.1111**	**0.9422**	**7.39%**	**75.52%**
100	3	20	64	0.2363	0.7387	16.03%	42.03%
100	3	50	64	0.2339	0.7438	15.99%	42.67%

alpha=0.1 and factor=5

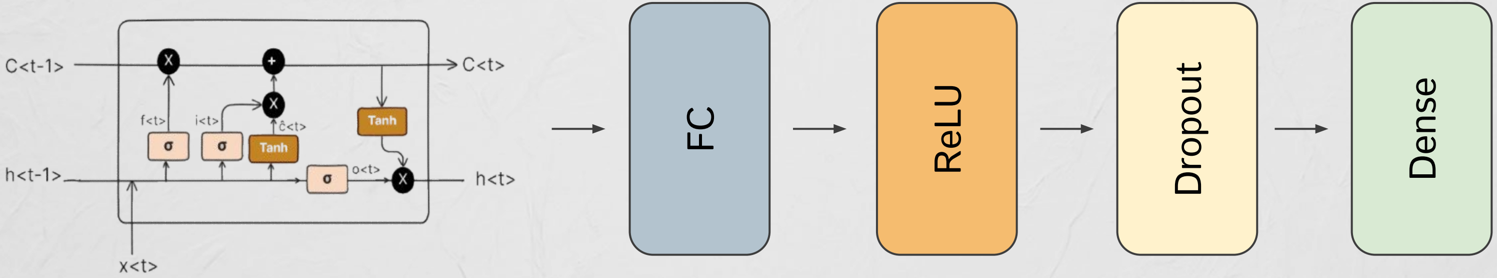
First Approach: Hyperparameter Tuning

- Shorter Sequence Length: does not affect the performance of the model
- Smaller Future Period Predict: to evaluate accuracy, it is better to start predicting more than 1-2 hours, as consumption varies depending on the time of the day → 1
- Number of epochs: tried with 75, 100, 200 epochs… got overfitting as it converges quite fast → leave 20-30 to avoid it
- Batch Size: good batch size for our architecture → 64

Second Approach: One LSTM with FC

The Model: LSTM+FC+ReLU+Dropout

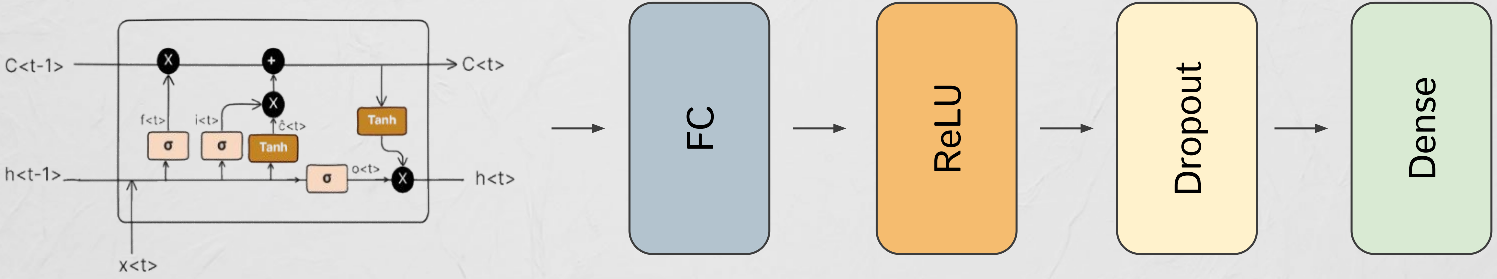
LSTM Architecture



Second Approach: One LSTM with FC

<u>The Model:</u> LSTM+FC+ReLU+Dropout

LSTM Architecture



1. **Dense layer after LSTM:** turn temporal patterns into useful features for better predictions.
2. **Dropout after Dense and ReLU:** reduces overfitting.
3. **LSTM + Dense:** more stable and easier to train.

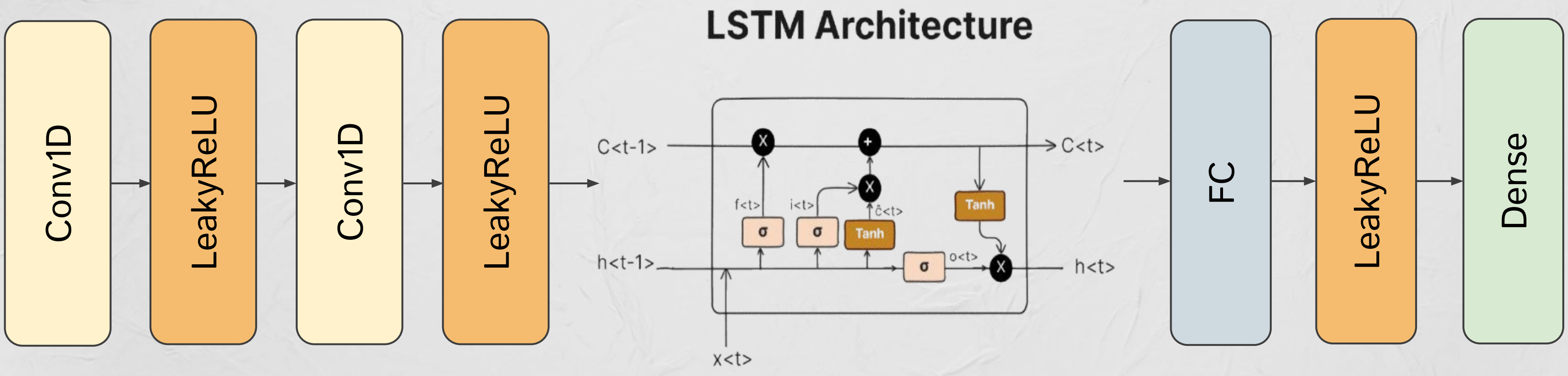
Second Approach: One LSTM with FC

The Model: LSTM+FC+ReLU+Dropout

Seq. len	Future predict	Epochs	Batch	RMSE	R^2	MAPE	Accuracy
50	**1**	**30**	**64**	**0.0551**	**0.9784**	**3.94%**	**91.86%**
50	3	30	64	0.1266	0.8858	8.69%	65.35%
50	6	30	64	0.1982	0.7211	14.35%	48.06%
100	**1**	**30**	**64**	**0.0547**	**0.9789**	**3.91%**	**93.48%**
100	3	30	64	0.1328	0.8754	9.37%	64.47%
100	6	30	64	0.1911	0.7420	13.83%	49.21%

Third Approach: LSTM with Conv1D

<u>The Model:</u> Conv1D+LeakyReLU+Conv1D+LeakyReLU+LSTM+FC+LeakyReLU



Third Approach: LSTM with Conv1D

1. **Conv1D Layers: extract short-term patterns** and reduce input noise by focusing on **relevant temporal features** before passing them to LSTM.

2. **LeakyReLU Activations:** help maintain **gradient flow** during training and improve performance when the input has **small variations**.

3. **Second Conv1D** + **LeakyReLU: stack feature extraction** and combine local patterns into **higher-level abstractions**.

4. **LSTM Layer:** capture **long-term dependencies** and good for time series forecasting since it maintains **temporal memory** of past inputs.

5. **Fully Connected Layer** + **LeakyReLU:** transforms time-dependent features into a **final prediction space**.

Third Approach: LSTM with Conv1D

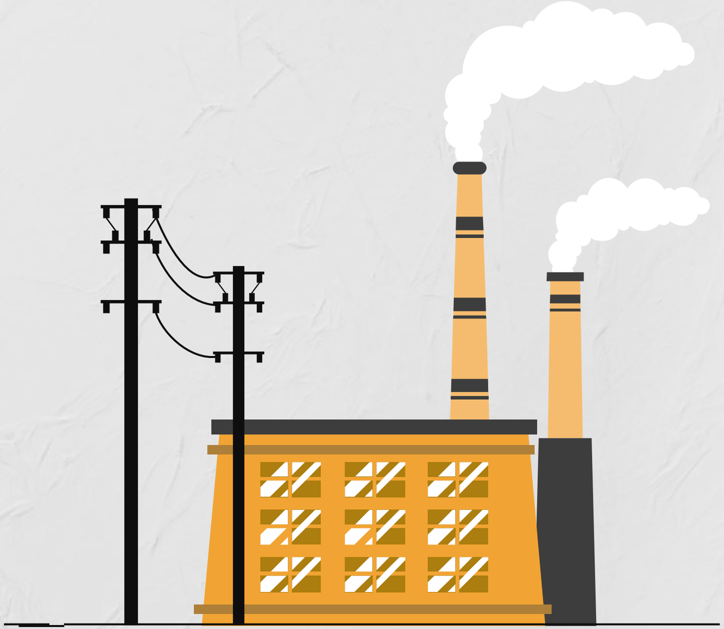
The Model: Conv1D+LeakyReLU+Conv1D+LeakyReLU+LSTM+FC+LeakyReLU

Seq. len	Future predict	Epochs	Batch	RMSE	R^2	MAPE	Accuracy
50	**1**	**30**	**64**	**0.0525**	**0.9804**	**3.61%**	**95.00%**
50	3	30	64	0.1341	0.8722	8.91%	65.44%
50	6	30	64	0.2180	0.6626	14.45%	46.79%
100	**1**	**30**	**64**	**0.0533**	**0.9799**	**3.71%**	**94.69%**
100	3	30	64	0.1278	0.8846	8.88%	66.35%
100	6	30	64	0.2193	0.6603	14.54%	47.61%

04

Conclusions

What did we learn?



Conclusions

Data preprocessing was as important as the model itself: without smoothing, null handling, and outlier removal, even the best architectures underperformed.

Smoothing and outlier removal are crucial for effective model training, especially in the presence of noise.

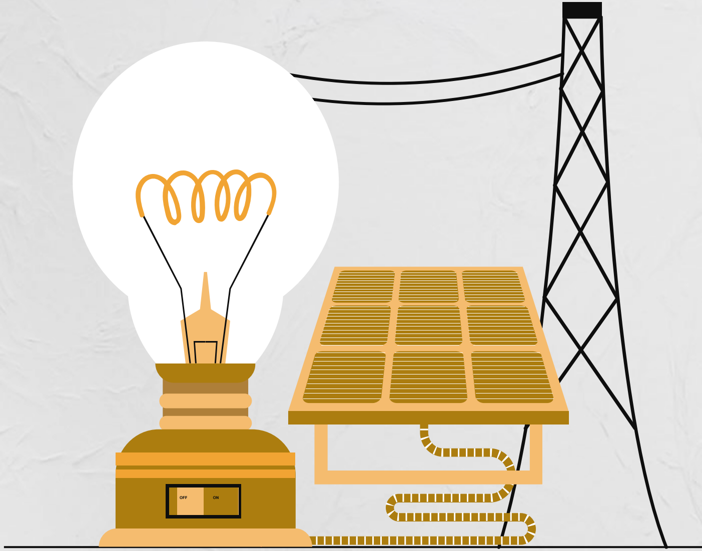
The Conv1D+LSTM architecture is well-suited for this type of data, particularly for short-term time series forecasting, as the power values exhibit significant variability over time.

The high accuracy of the final model (95%) demonstrates the feasibility of deploying such solutions in real-world applications.

Looking Ahead

Impact Beyond the Model

This project has prompted us to contemplate **the potential of real-time energy forecasting systems**. In fact, incorporating external data such as weather or occupancy can significantly enhance the performance of these models, **leading to solutions that benefit various segments of society**, including **energy providers** and also **end-users** like all of us.



Thank you

Do you have any questions?

Group L

Adrià Soria
Judit Viladecans
Paula Ceprián