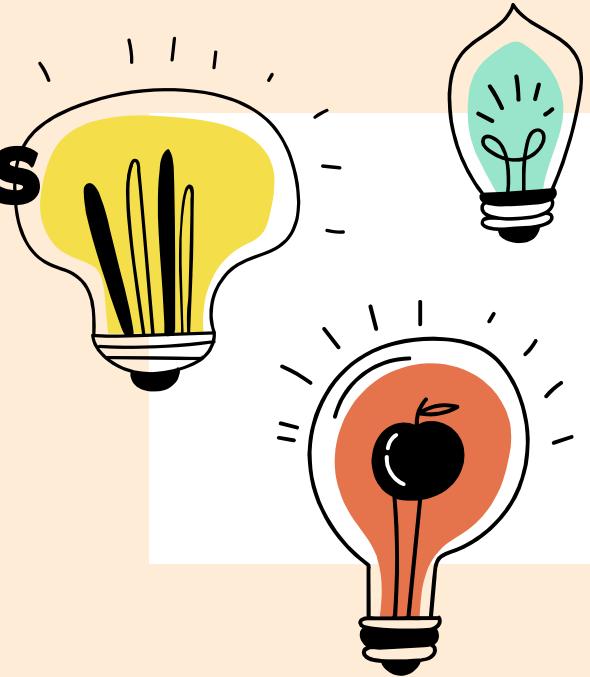


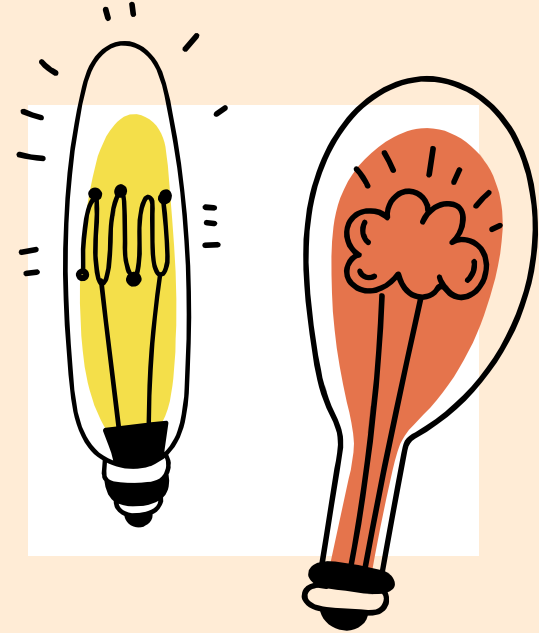
Time Series Analysis and Forecasting of Electricity Usage



Group: YYYY

Introduction

This project focuses on analyzing electricity consumption patterns and **predicting daily electricity usage for each client** using the Electricity Load Diagrams (2011-2014) dataset from the UCI Machine Learning Repository. The dataset contains electricity consumption data for Portuguese **370 clients** over a period of four years, recorded at **15-minute intervals**. This dataset provides a solid foundation for identifying consumption trends and developing robust forecasting models.



01	Problem and Business value
02	Process and Change Management
03	Project MileStone and TimeLine
04	Data Sources
05	Pre-processing
06	Pre-Modeling
07	Model
08	Results and Conclusion
09	Challenges and Future Steps



Table of contents

I. Problem and Business Value

Electricity demand changes constantly, causing waste, high costs, and power grid instability. Without accurate forecasting, energy providers may overproduce or struggle to meet demand, leading to higher expenses and potential blackouts.

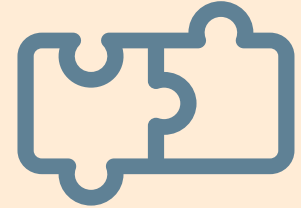
By using energy forecasting, companies can better balance supply and demand, reduce operational costs, and improve efficiency. It also supports renewable energy integration, lowers carbon emissions, and helps prevent power outages during extreme conditions. Additionally, businesses can optimize energy usage, minimize peak-hour costs, and implement smarter pricing strategies to increase revenue.



2. Process and Change Management

Utility Companies

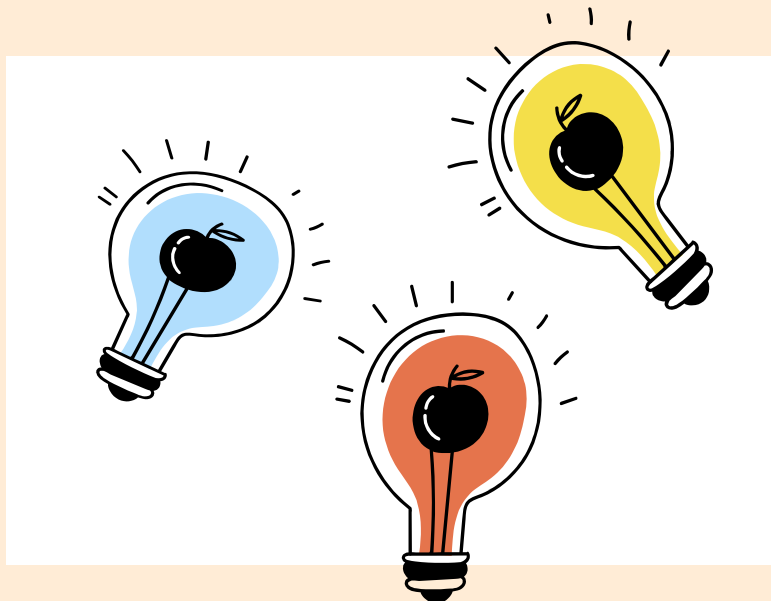
- Automate power supply scheduling using AI forecasting
- Replace manual adjustments with real-time predictive analytics
- Optimize power generation and distribution to reduce waste and costs
- Train grid operators to interpret forecasts and respond to anomalies



Policymakers & Regulators

- Use AI forecasts for long-term energy planning and sustainability.
- Plan infrastructure investments (e.g., power plants, EV charging stations)
- Collaborate on data-sharing and regulatory alignment
- Optimize renewable energy integration and grid resilience

3. Project Milestone



 **EDAV & Model Development**

 **Model Validation & Refinement**

Tune hyperparameters for better predictive accuracy

 **Results Interpretation**

Timeline

Preprocessing & EDA

Model Validation
& Refinement

2/20

2/25

3/15

3/23









Architecture:
Model
Development









Results
Interpretation

4.1 Data Source - Electric Load

-  **Dataset:** UCI Electricity Load Diagrams (2011-2014)
-  **Granularity:** 15-minute electricity consumption readings
-  **Columns:** Timestamp + multiple clients' energy usage (in kW)
-  **Time Zone:** Portuguese time with Daylight Saving Time (DST) shifts
-  **Missing Data Handling:** Some clients have zero consumption before activation
-  **Format:** Stored as a semicolon-separated (.txt) file, converted to CSV for analysis

4.2 External Data Source - Lisbon (Weather)

-  **Dataset:** Lisbon Weather Data (2008–2020), covering daily observations
-  **Granularity:** Daily weather conditions for each calendar date
-  **Columns:** date_time, maxtempC, mintempC, snowfall, sun hours, UV index, humidity, etc.
-  **Time Zone:** Local time in Lisbon, Portugal
-  **Missing Data Handling:** no missing data
-  **Format:** Originally downloaded as CSV from Kaggle

(Vivas, 2020)

5. Data Pre-Processing

- **Standardized Numerical Format:** Converted decimal separators from commas (,) to periods (.) for correct numerical processing.
- **Unit Conversion (kW → kWh):** Divided all values by 4 to convert power readings from kilowatts to kilowatt-hours for consistency.
- **Removed 2011 Data:** Excluded 2011 records due to many zero values from inactive customers, ensuring cleaner data.
- **Daylight Saving Time (DST) Adjustments:**
 - **March (Spring Forward):** Missing 1:00 - 2:00 AM values were filled using interpolation.
 - **October (Fall Back):** Duplicated 1:00 - 2:00 AM values were averaged to prevent overestimation.

6. Pre-Modeling

- **Data Aggregation:** sum up the readings taken at 15-minute intervals to calculate the total electricity consumption for each day, which help us observe the various patterns among different clients.
- **Train-Test Splitting:** split the data chronologically, using the last 365 days (1 year of 2014) as the test set and the remaining earlier data (2012-2013) for training.
- **Dictionary:** To build a multi-time series forecasting model that predicts the daily electricity consumption for each client, we use dictionary to treat each client as individual time series but process together under a shared model.
- **Dynamic Time Warping (DTW) Based Clustering:** assess shaped-based similarity and analyze clients with different behaviors separately (client segmentation).

6.1 Pre-Modeling: Data Aggregation

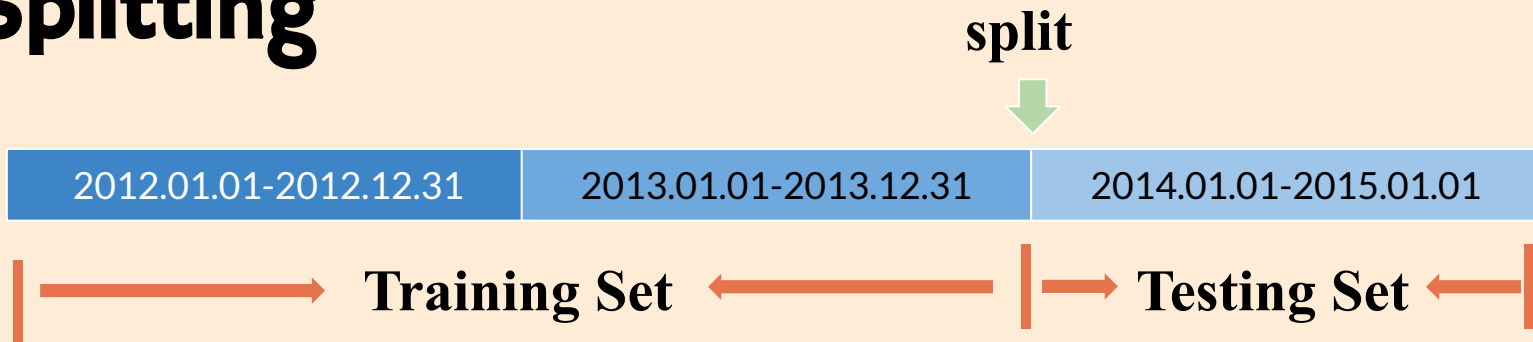
Aggregate 15 minutes interval into one day



timestamp	MT_001	MT_002	MT_003	MT_004	MT_005
2012-01-01 00:00:00	0.000000	0.000000	0.000000	0.000000	0.000000
2012-01-01 00:15:00	0.951777	5.689900	19.331017	34.044715	17.682927
2012-01-01 00:30:00	1.269036	5.689900	19.331017	34.044715	18.292683
2012-01-01 00:45:00	0.951777	5.689900	19.331017	35.060976	17.378049
2012-01-01 01:00:00	0.951777	5.689900	19.331017	35.060976	18.902439

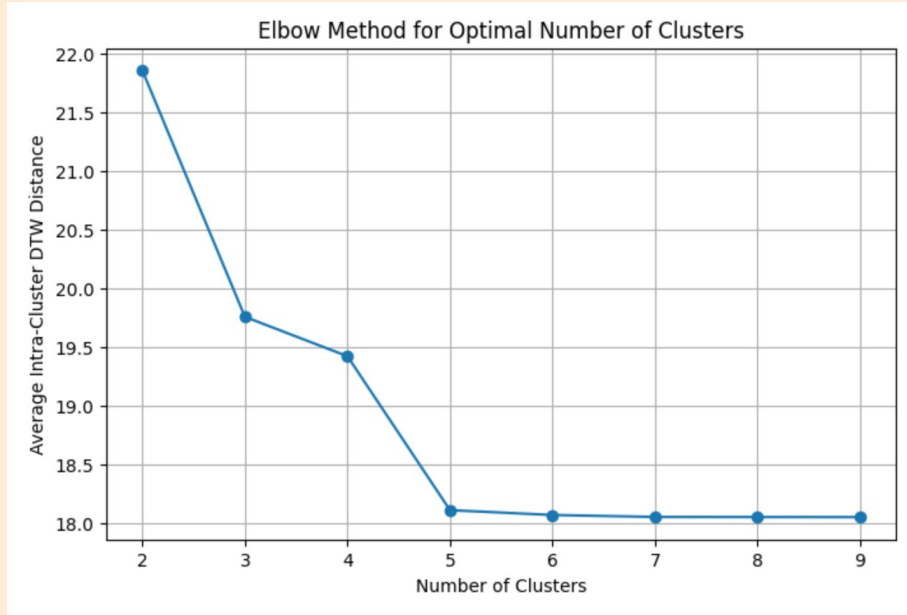
	timestamp	MT_001	MT_002	MT_003	MT_004	MT_005
0	2012-01-01	177.982234	624.466572	499.782798	2846.036585	1372.256098
1	2012-01-02	256.345178	644.025605	119.678540	3072.154472	1553.353659
2	2012-01-03	260.152284	719.061166	193.744570	2951.219512	1614.024390
3	2012-01-04	273.477157	627.489331	342.311034	2924.288618	1663.109756
4	2012-01-05	337.880711	660.206259	991.311903	2923.272358	1596.036585

6.2 Pre-Modeling: Train Test Splitting



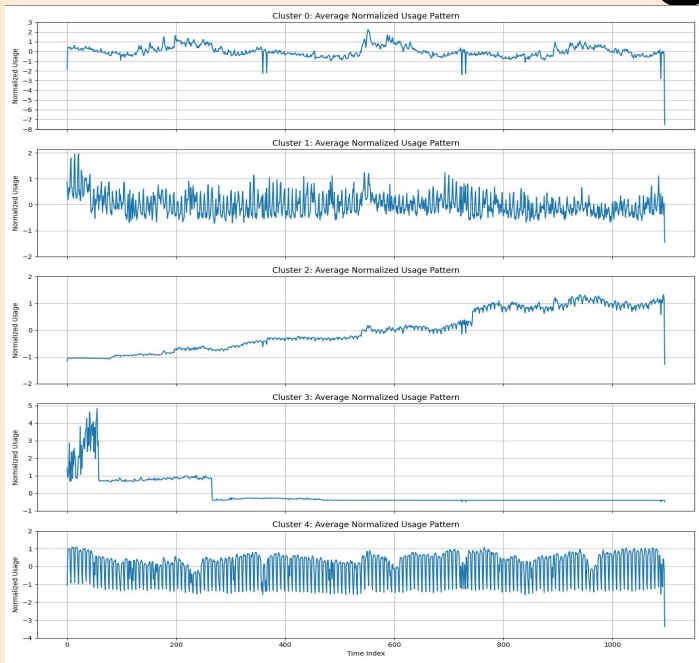
We split the data chronologically, using the 2012-2013 as training data set and 2014 as test data set. This setup preserves the natural time order and ensures that at least two full years of seasonal patterns are captured in the training set, which is important for reliable time series forecasting.

6.3 Pre-Modeling: DTW Clustering



- **DTW:** measure the similarity between two time series, even if they are not perfectly aligned in time
- **Why Clustering:** Group client with similar behaviors together instead of averaging all the clients (client segmentation)
- **Number of Clusters:** K=5 (Elbow Point)
- **What to do next?** Take cluster label as a feature to build model

6.3 Pre-Modeling: DTW Clustering



- **Cluster 0 (280 clients)**: Steady usage with minor variability, business with a constant level of electricity to support regular operations
- **Cluster 1 (7 clients)**: Highly fluctuating usage, businesses that don't follow a fixed schedule
- **Cluster 2 (53 clients)**: Gradually increasing trend, business expanding or using more energy as time goes on
- **Cluster 3 (2 clients)**: Initial spike followed by a long decline, businesses that initially operated at full capacity but then shutdown or scale-down.
- **Cluster 4 (28 clients)**: Periodic cyclical pattern, business with fixed and predictable schedule

6.4 Pre-Modeling: Dictionary

```
[45] print(series_dict_train['MT_001'])
```

```
timestamp
2012-01-01    177.982234
2012-01-02    256.345178
2012-01-03    260.152284
2012-01-04    273.477157
2012-01-05    337.880711
...
2013-12-27     57.106599
2013-12-28     53.934010
2013-12-29     54.251269
2013-12-30     42.195431
2013-12-31     42.195431
Freq: D, Name: MT_001, Length: 731, dtype: float64
```

```
[46] print(series_dict_test['MT_001'])
```

```
timestamp
2014-01-01     47.588832
2014-01-02     56.789340
2014-01-03     61.548223
2014-01-04     41.878173
2014-01-05     50.444162
...
2014-12-27     52.982234
2014-12-28     56.789340
2014-12-29     62.182741
2014-12-30     58.058376
2014-12-31     57.423858
Freq: D, Name: MT_001, Length: 365, dtype: float64
```

- Why Dictionary?

Help build a multi-time series forecasting model that predicts the daily electricity consumption for each client; allow us to treat each client as an individual time series, but processed together under a shared model

- Client Time Series Dictionary (series_dict):

- store the historical electricity consumption data for each client
- Key: client ID (e.g. “MT_001”)

6.4 Pre-Modeling: Dictionary

```
[15] print(exog_dict_train['MT_001'])
```

timestamp	sin_dow	cos_dow	sin_doy	cos_doy	avg_temp	Cluster
2012-01-01	-0.781831	0.623490	1.721336e-02	0.999852	12.0	0
2012-01-02	0.000000	1.000000	3.442161e-02	0.999407	13.0	0
2012-01-03	0.781831	0.623490	5.161967e-02	0.998667	10.5	0
2012-01-04	0.974928	-0.222521	6.880243e-02	0.997630	13.5	0
2012-01-05	0.433884	-0.900969	8.596480e-02	0.996298	12.5	0
...
2013-12-27	-0.433884	-0.900969	-6.880243e-02	0.997630	15.0	0
2013-12-28	-0.974928	-0.222521	-5.161967e-02	0.998667	12.0	0
2013-12-29	-0.781831	0.623490	-3.442161e-02	0.999407	11.0	0
2013-12-30	0.000000	1.000000	-1.721336e-02	0.999852	10.5	0
2013-12-31	0.781831	0.623490	6.432491e-16	1.000000	14.0	0

[731 rows x 6 columns]

```
print(exog_dict_test['MT_001'])
```

timestamp	sin_dow	cos_dow	sin_doy	cos_doy	avg_temp	Cluster
2014-01-01	0.974928	-0.222521	1.721336e-02	0.999852	14.0	0
2014-01-02	0.433884	-0.900969	3.442161e-02	0.999407	15.5	0
2014-01-03	-0.433884	-0.900969	5.161967e-02	0.998667	14.5	0
2014-01-04	-0.974928	-0.222521	6.880243e-02	0.997630	12.5	0
2014-01-05	-0.781831	0.623490	8.596480e-02	0.996298	14.0	0
...
2014-12-27	-0.974928	-0.222521	-6.880243e-02	0.997630	11.5	0
2014-12-28	-0.781831	0.623490	-5.161967e-02	0.998667	13.0	0
2014-12-29	0.000000	1.000000	-3.442161e-02	0.999407	9.0	0
2014-12-30	0.781831	0.623490	-1.721336e-02	0.999852	8.0	0
2014-12-31	0.974928	-0.222521	6.432491e-16	1.000000	8.5	0

[365 rows x 6 columns]

- **Exogenous Features Dictionary (exog_dict):**
 1. **Cyclical time features:** sine and cosine transformations of the day of the week and day of the year
 2. **Average daily temperature (avg_temp):** the average of the daily minimum and maximum temperatures in Lisbon
 3. **Cluster label (cluster):** Based on DTW-based time series clustering, each client was assigned a cluster representing its typical consumption pattern.

7. Model

- **SARIMAX:** Cluster-level models capture weekly seasonality and temperature-driven patterns in electricity demand using representative clients' data.
- **LSTM:** Sequence-based deep learning model that learns temporal patterns from past electricity usage to forecast future consumption.
- **Hist GradientBoostingRegressor:** The model enables efficient multi-time series forecasting across hundreds of users by modeling nonlinear patterns with rich feature inputs.

7.I SARIMAX

Why SARIMAX?

- Captures **weekly seasonality** in electricity usage.
- Incorporates **external factors** like temperature.
- Suitable for **non-stationary** time series data.
- Offers a **clear, interpretable** structure.

How We Used SARIMAX

- Applied **one SARIMAX model per cluster** (clusters built using DTW).
- Selected **up to 5 clients per cluster**, and **concatenated their daily usage**.
- Used **temperature**, **day-of-week/year cycles**, and **cluster ID** as exogenous variables.
- Split the data to **train on all but the last 365 days**, using the last year for testing.
- Configured model with **(1,1,1)** and **(1,1,1,7)** for **weekly seasonality**.
- Evaluated using **RMSE**, **MAE**, and **MAPE**.

7.2 LSTM - Long Short-term Memory

Why LSTM?

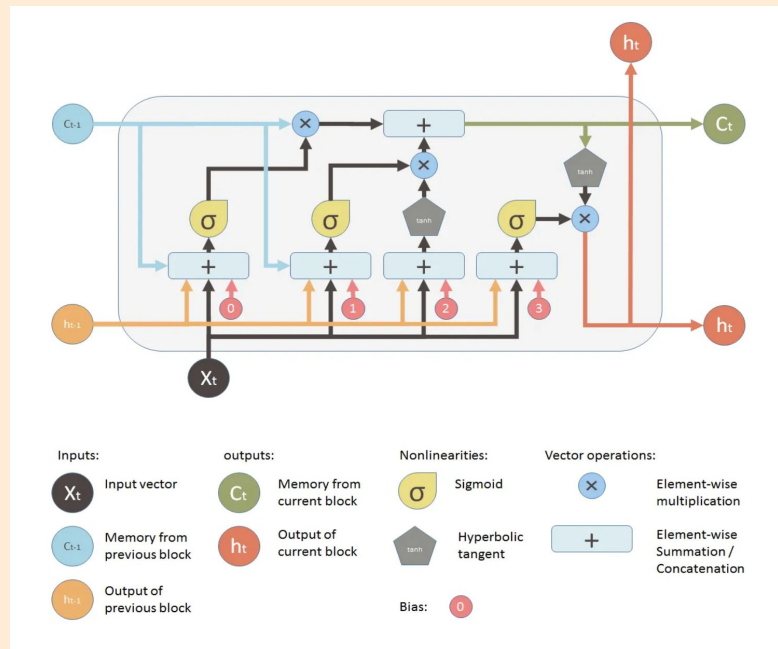
- Captures **long-term dependencies** in sequential data.
- Handles **nonlinear temporal dynamics** effectively.
- Well-suited for **seasonality, weekly cycles, and user behavior trends**.

Model Configuration:

- Hidden size: 16; Learning rate: 0.03; Dropout: 0.1
- Loss function: **SMAPE (Symmetric Mean Absolute Percentage Error)**

Training Strategy:

- Global model trained across all users.
- Trained for **10 epochs** using PyTorch Lightning Trainer.
- Applied **gradient clipping** to improve stability.
- Learns **shared patterns across users**, while capturing individual usage behavior.



(Yan, 2017)

7.3 Multi-series Recursive Forecaster

Why Multi-Series Forecasting?

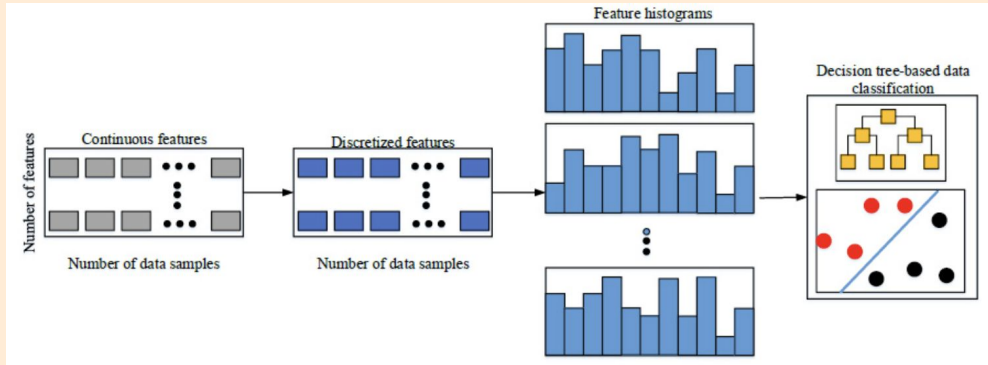
We have 370 clients, each with their own time series of daily electricity usage. Instead of training 370 separate models, we use one shared model that learns from all series. This allows us to capture global patterns across clients, improves generalization, and reduces computation and code complexity

Why Recursive Forecasting?

We predict multiple future steps (e.g., next 24 days) one at a time. After predicting day $t+1$, the prediction is used as input to forecast $t+2$, and so on.

This lead to our model choice: **HistGradientBoostingRegressor**

7.4 Histogram-based Gradient Boosting Model



(Hoang & Nguyen, 2023)

Applied a sliding window strategy

- input: past 30 days
- target: next 7 days

Used recursive multi-step prediction

- predict Day 1, feed it into Day 2, and so on
- mimics real-world conditions where future inputs are unknown

Employed gradient boosting (HGBR)

- Ability to capture nonlinear patterns
- Robustness to outliers
- Strong performance with tabular time series data

Recursive Structure

- introduces temporal dependency in predictions
- Forces model to generalize across varying input patterns

8.1 Result - SARIMAX

Cluster-Level Evaluation:

- Cluster 0 (users: 5): MAPE = 72.40%
- Cluster 1 (users: 5): MAPE = 75.18%
- Cluster 2 (users: 5): MAPE = 426.93%
- Cluster 4 (users: 5): MAPE = 67.70%
- Cluster 3 was removed as an outlier — actual usage was flat, while SARIMAX produced unrealistic fluctuations.

Average MAPE across clusters: 160.55%

SARIMAX Results Summary

- Clusters 0, 1, and 4 showed reasonable forecasting accuracy (MAPE ~67–75%).
- Cluster 2 had an extremely high MAPE (426.93%) due to outliers and unstable usage patterns, indicating poor model generalization.
- Overall average MAPE (160.55%) suggests SARIMAX struggles with volatile or irregular demand.
- Model performs best on stable, seasonal patterns but is sensitive to sudden shifts and noise.



8.2 Result - LSTM

The overall Performance

- MAPE: 135.94%
- The model has difficulty generalizing across all user behaviors.
- High Mape might due to the presence of near-zero values in the actual data (e.g., MT_003)

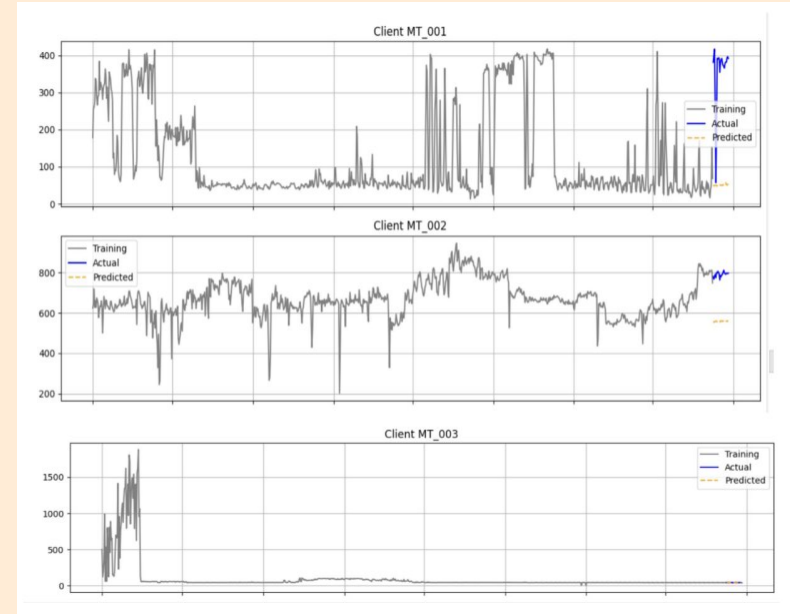
The plots show actual vs. predicted values

- The model performs reasonably well for users with stable and consistent consumption patterns, capturing general trends.
- It struggles with users showing irregular, flat, or noisy usage, leading to less accurate predictions.

While the LSTM captures some consumption dynamics, the results highlight several areas for improvement, including:

- Using longer historical input windows
- Enhancing feature engineering
- Exploring hybrid or alternative model architectures

These findings suggest that while LSTM provides some predictive value, it may not be the optimal approach for all users in this dataset.

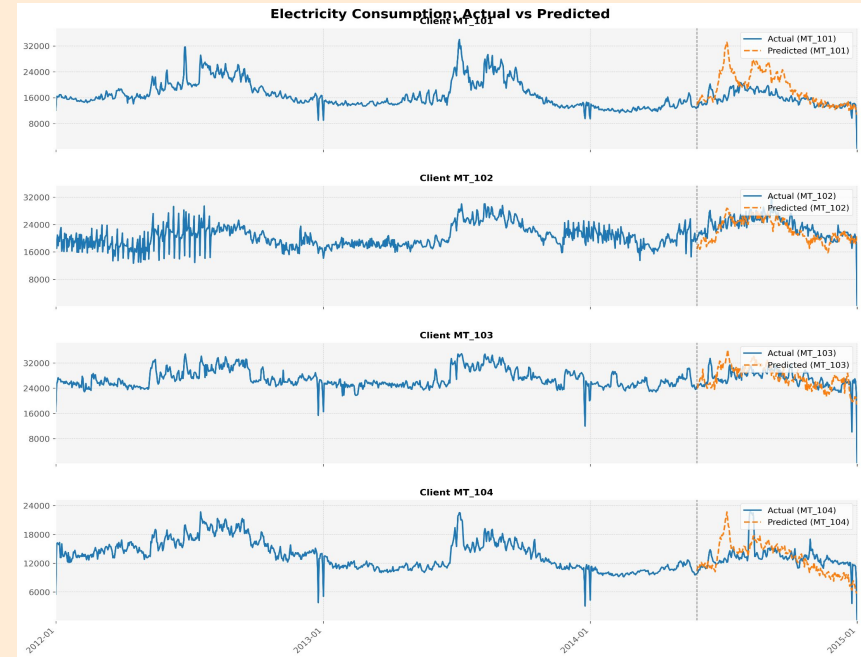


8.3 Result - *HistGradientBoostRegressor*

Predicted and actual electricity consumption comparison for selected clients

MAPE: 30.20%, which is consider to be a reasonable forecasting

Performance is **stable and reliable** across a diverse user base



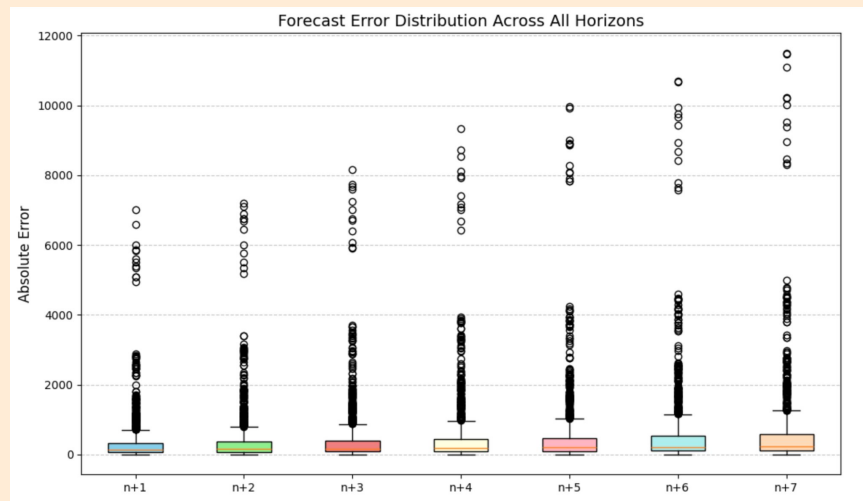
8.3 Result - *HistGradientBoostRegressor*

Observed trend:

- Errors **slightly increase** as forecast horizon extends (Day 1 → Day 7)
- Typical in **recursive forecasting** setups

Boxplot Analysis:

- Tighter error bounds in early predictions (e.g., Day 1)
- Gradual spread increase in longer horizons (e.g., Day 7)
- Majority of predictions remain **within a reasonable error range**



9. Challenges and Future Steps

1. Improving Cluster Balance:
 - 280/370 clients are in cluster 0 -> imbalanced cluster
 - Utilized more advanced clustering techniques in the future
2. Limited Feature Availability:
 - Original dataset only included client IDs and electricity usage over time
 - Need additional useful features for better prediction



Conclusion

- We tested SARIMAX and LSTM, but early models struggled to capture client-level variability and produced inconsistent results.
- Added features like temperature, cluster labels, and time-based signals (day-of-week/year) to support modeling.
- Used DTW clustering and trained one model per cluster, but LSTM and SARIMAX underperformed due to lack of personalization.
- Switched to **HistGradientBoostingRegressor**, which models all clients together using rich, client-specific features.
- Final results: **MAE** 655.66, **RMSE** 779.34, **MAPE** 30.20% — far better than SARIMAX and LSTM.
- Rolling validation confirmed accuracy decreases slightly over time but stays reliable.
- Most clients saw low error (median MAPE 4.19%), with a few outliers causing higher mean.
- The model scales well and supports better grid planning and personalized energy insights.

Reference

Vivas, L. (2020, April 23). *Spain Portugal weather*. Kaggle.
<https://www.kaggle.com/datasets/luisvivas/spain-portugal-weather?resource=download>

Rodrigo, J. A., & Ortiz, J. E. (2022, October). Global forecasting models: Modeling multiple time series with machine learning. Global forecasting models: multiple time series forecasting with skforecast.
<https://cienciadedatos.net/documentos/py44-multi-series-forecasting-skforecast>

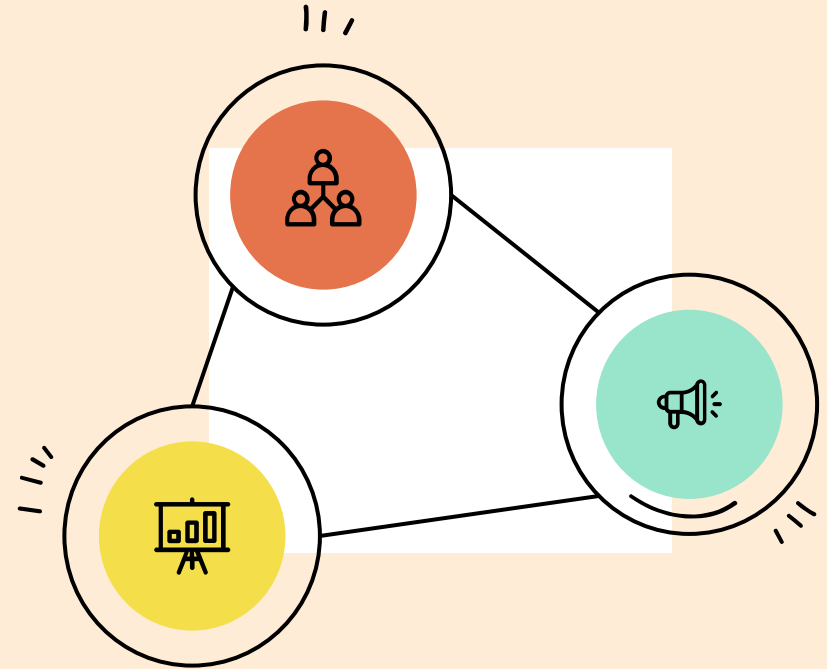
Hoang, N.-D., & Nguyen, Q.-L. (2023). *Computer Vision-based recognition of pavement crack patterns using light gradient boosting machine, deep neural network, and Convolutional Neural Network*. Soft Computing in Civil Engineering. https://www.jsoftcivil.com/article_168894.html

Yan, S. (2017, November 15). *Understanding LSTM and its diagrams*. Medium.
<https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714>

Mission statement

We aim to create accurate electricity consumption forecasts using time series models. By applying feature engineering, exploratory data analysis, and fine-tuning model parameters, we improve prediction accuracy.

Our goal is to build a scalable forecasting system that helps manage energy better, keeps the grid stable, and supports smarter decision-making. In the long run, this will cut costs, improve efficiency, and make power distribution more reliable.



Team



Yueer Liu



**Zimeng(Sandy)
Li**



Ziying Song



Yihan Yang

Thank You!

