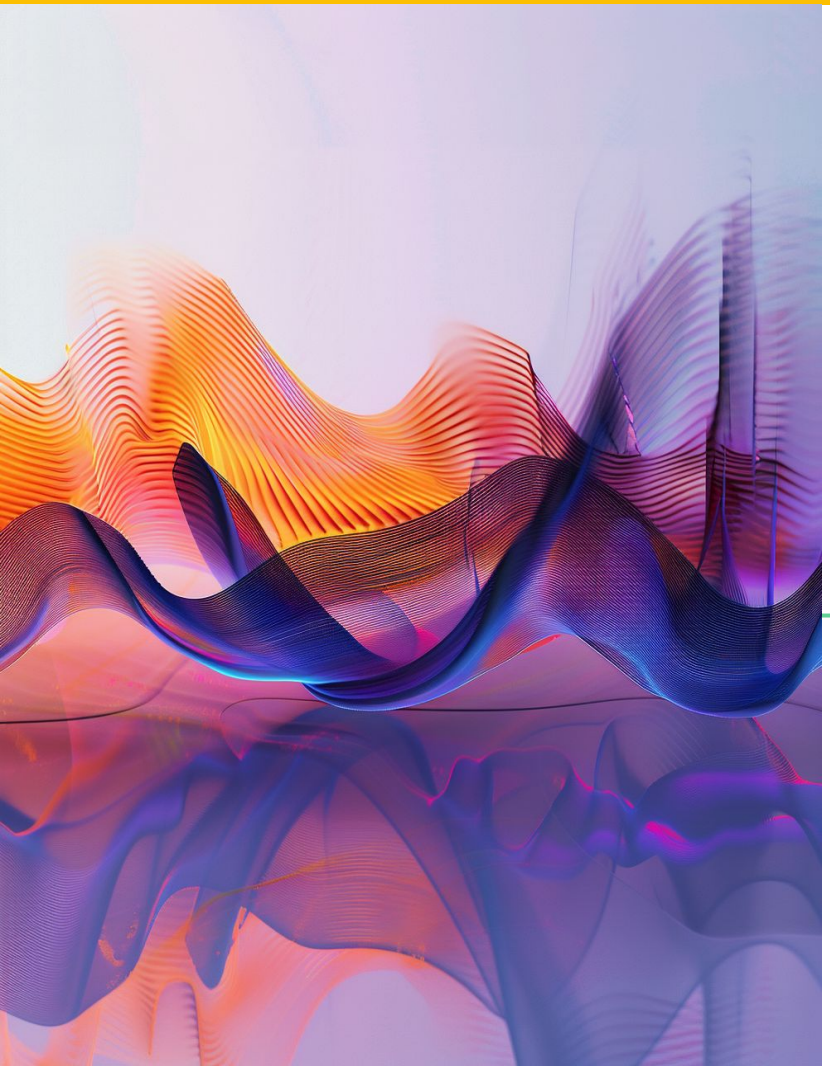


Techniques in Time Series Analysis for Machine Learning Enthusiasts

A Comprehensive Guide for Professionals Seeking to Enhance their Time-Series Skills

– Sankalp Gilda, Ph.D.

DATATUNE
NASHVILLE
2024

Thank you sponsors!

Gold Sponsor

DBADuck ben miller

DataForge

ddn

Lovelytics

BROOKLYN DATA CO. a velir company

CGI

AdvizeX FULCRUM IT PARTNERS

sigma

NASHVILLE SOFTWARE SCHOOL

ACREE CONSULTING

((CENTRIC))

onix

FLAGRANT

ROYAL BLUE ANALYTICS

Couchbase

Y42

GREATER NASHVILLE TECHNOLOGY COUNCIL

vaco

coneXess

Agenda

- The Uniqueness of Time Series (TS)
- What This Presentation Is Not
- Exploring the TSU™
  - **Predicting the Future: Forecasting**
  - Categorizing Patterns: Classification
  - Spotting the Outliers: Anomaly Detection
  - Grouping Similarities: Clustering
- **TSF meets ML**
  - Cross-Validation
  - Deterministic v/s Probabilistic + Bootstrapping
  - Metrics
  - Libraries
- Hot Off The Press: Innovations in TS Analysis

Agenda

- The Uniqueness of Time Series (TS)
- What This Presentation Is Not
- Exploring the TSU™
  - **Predicting the Future: Forecasting**
  - Categorizing Patterns: Classification
  - Spotting the Outliers: Anomaly Detection
  - Grouping Similarities: Clustering
- **TSF meets ML**
  - Cross-Validation
  - Deterministic v/s Probabilistic + Bootstrapping
  - Metrics
  - Libraries
- Hot Off The Press: Innovations in TS Analysis

Right Words To Google!

Time Series – An Introduction

- **Sequential Measurements:** Sequence of data points collected over time intervals, representing change of variable(s)
- **Ubiquitous Across Sectors:** Prevalent in almost every domain -- finance, meteorology, agriculture, etc.
- **Order Matters**: The sequence conveys crucial information; data points cannot be shuffled.

Time Series – In Practice

- **Predictive Analysis:** Aims to forecast future observations from historical trends.
- **Rich Field of Study:** Allows for forecasts, anomaly detection, and strategic insights.
- **Harnessing Potential:** Empowers data scientists for advanced predictive analytics and decision-making.
- **Autocorrelation and Non-Stationarity complicate analysis**

Time Series – They Are Everywhere!

Financial Markets

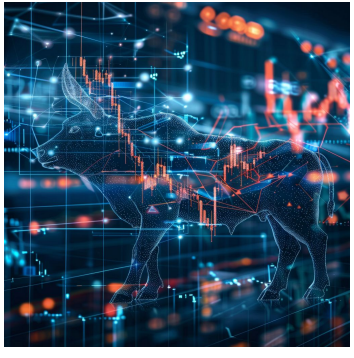
Climate Science

Energy

HealthCare

Time Series – They Are Everywhere!

Financial Markets



Climate Science

Energy

HealthCare

Manufacturing

Transportation

IoT

Astrophysics

What This Presentation Is Not

- **Algorithmic Trading 101**

What This Presentation Is Not

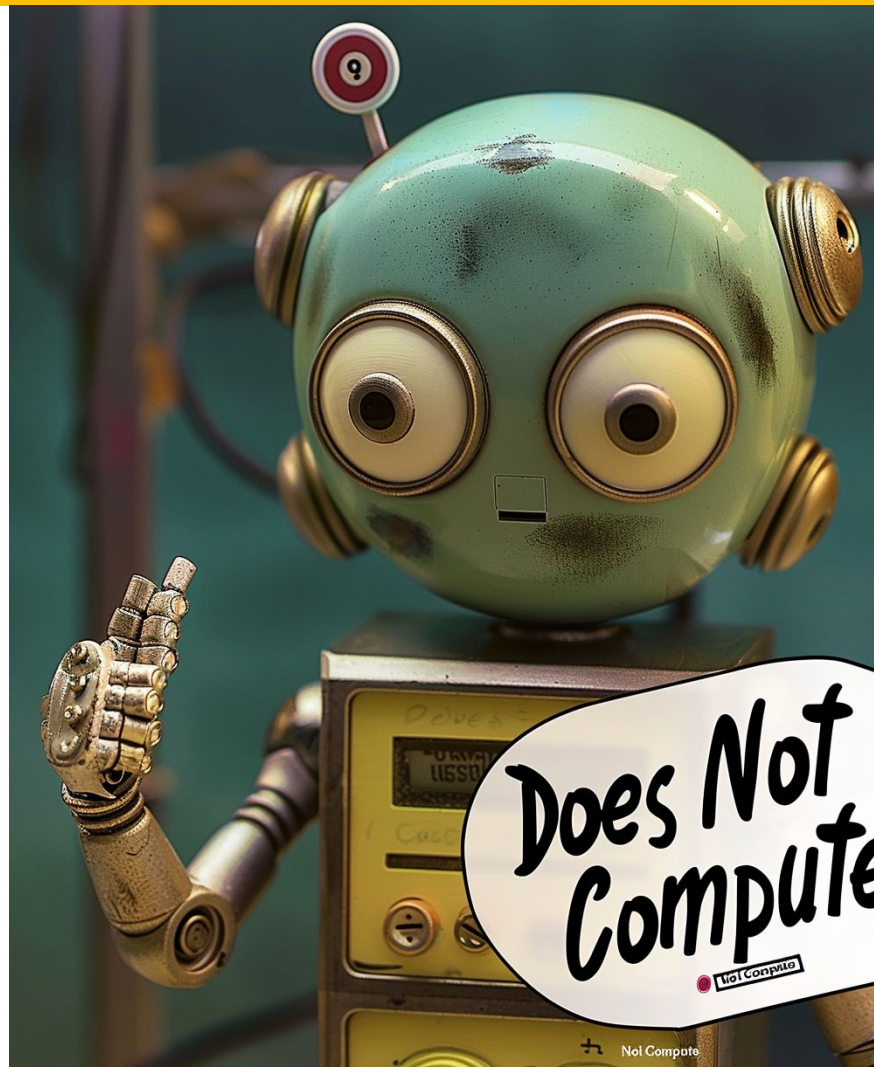
- Algorithmic Trading 101
- **Time Travel Tips**

What This Presentation Is Not

- Algorithmic Trading 101
- Time Travel Tips
- **Weather Forecasting For Dummies**

What This Presentation Is Not

- Algorithmic Trading 101
- Time Travel Tips
- Weather Forecasting For Dummies
- **A Sci-Fi Convention**



Anomaly Detection

- Identification of unusual patterns
- Real-time alerting systems
- Critical for fraud, faults, and spikes

Classification

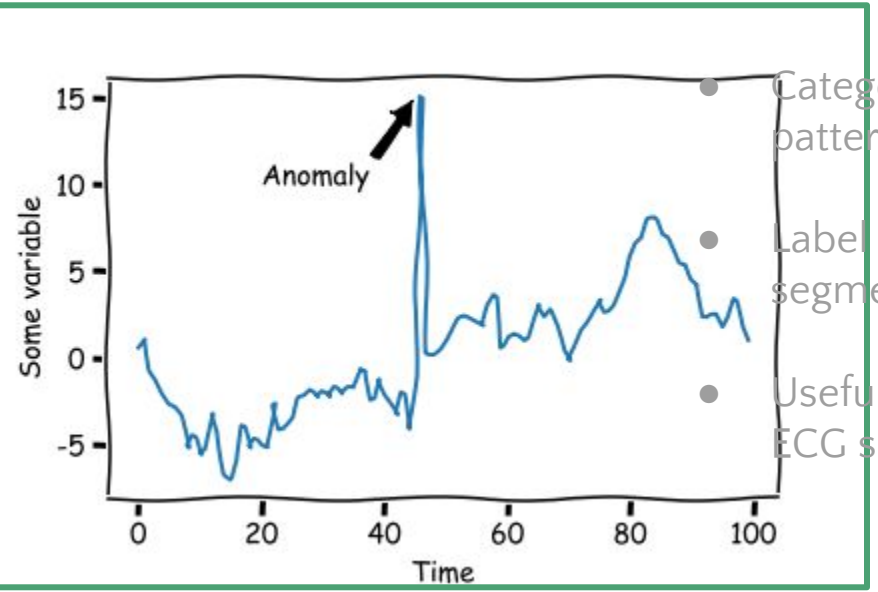
- Categorization based on patterns
- Label assignment to data segments
- Useful in gesture recognition, ECG signals

Clustering

- Grouping similar sequences
- Unlabeled data categorization
- Applied in market segmentation, activity patterns

Time Series Universe

Anomaly Detection



Classification

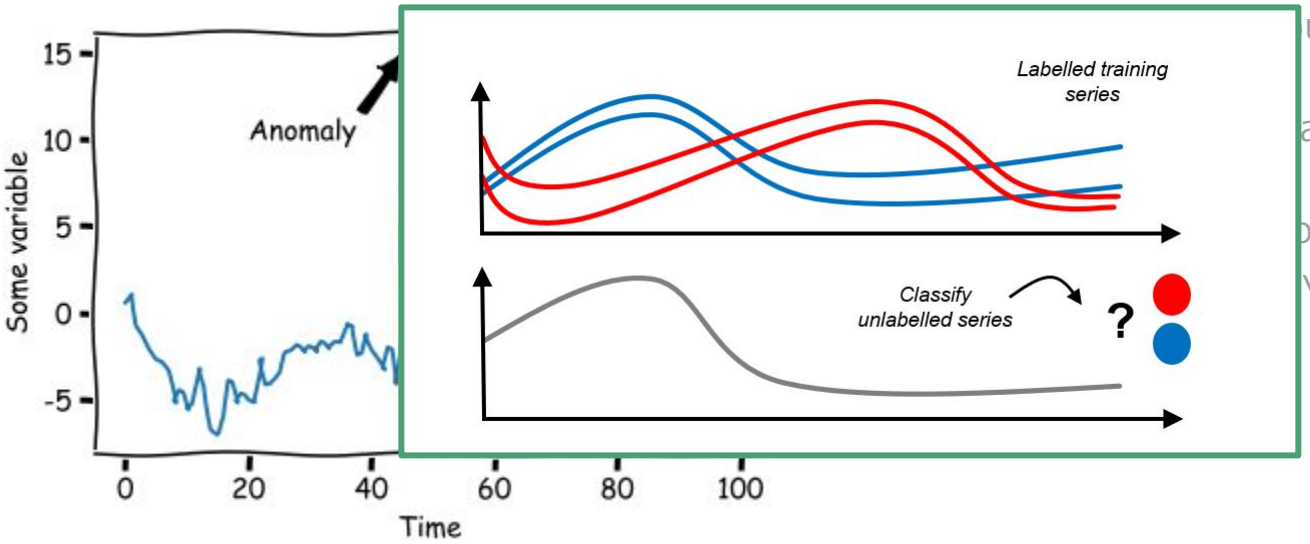
- Categorization based on patterns
- Label assignment to data segments
- Useful in gesture recognition, ECG signals

Clustering

- Grouping similar sequences
- Unlabeled data categorization
- Applied in market segmentation, activity patterns

https://github.com/DHI/tsod?tab=readme-ov-file

Anomaly Detection Classification Clustering



uping similar sequences

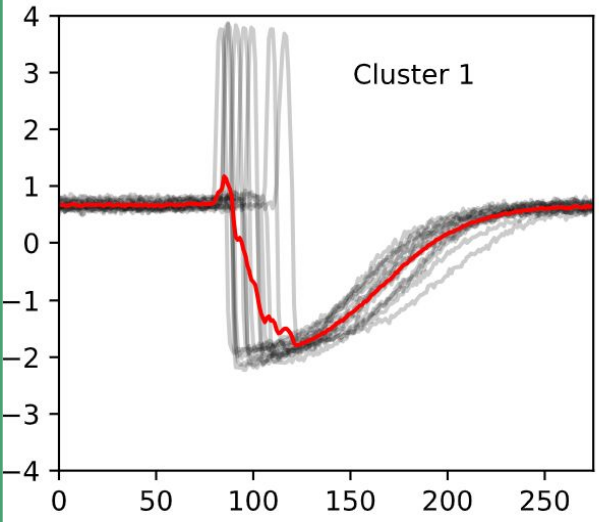
abeled data categorization

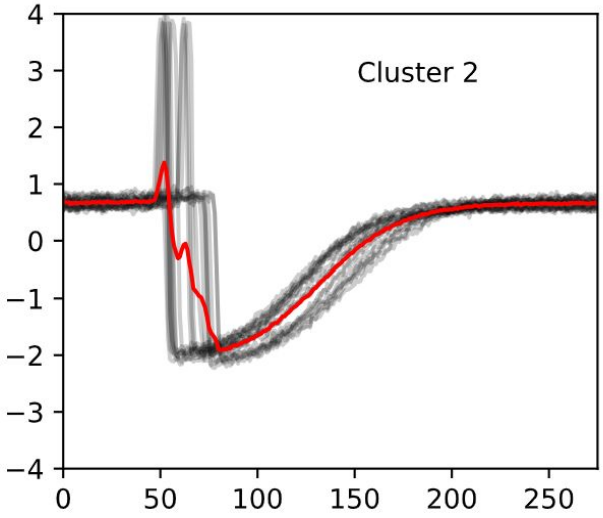
lied in market segmentation, vity patterns

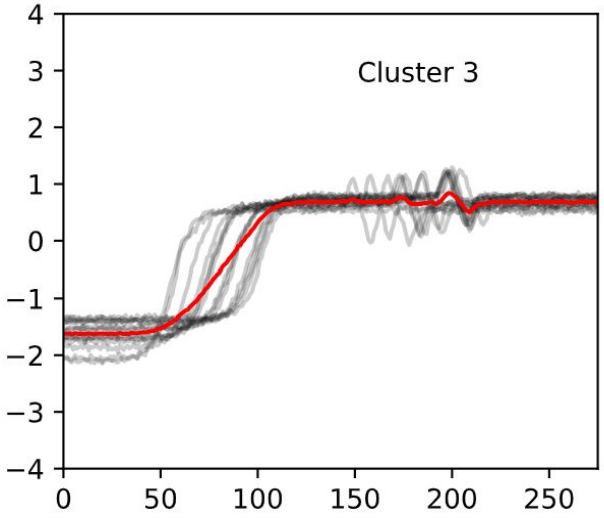
https://github.com/DHI/tsod?tab=readme-ov-file

https://www.sktime.net/en/stable/examples/02_classification.html

Anomaly Detection Classification **Clustering**







https://tslearn.readthedocs.io/en/stable/_images/kmeans.svg

Time Series Forecasting

- Time series forecasting (TSF) is an essential technique used to predict future values based on historical data.

- It plays a crucial role in various fields
  - Finance, where it's used for stock market prediction
  - Economics for macroeconomic indicators prediction
  - Environmental science for climate modeling
  - Energy sector for demand forecasting

Photo by Pexels

Basic Baselines: Let's Start At The Very Beginning

Y
1
4
9
16
25
36
49
64
81
?

- Naive → 81
- Seasonal Naive
- Mean → 31.66
- EWA → 41
- Moving Mean
- Median → 25
- Linear Fit → 81.67
- Polynomial Fit

Statistical Models

- **AR:** Predict future values from past observations.
- **MA:** Use past forecast errors for prediction.
- **ARIMA:** Combine AR and MA models with differencing.
- **SARIMA:** Include seasonality in ARIMA models.
- **ETS:** Model error, trend, and seasonality.
- **VAR**: Forecast interdependent time series.
- **STL:** Decompose series into trend, seasonality, and residuals.

Old Meets New

- Many ML models with numerous parameters can underperform in certain conditions.
- High autocorrelation in time series reduces the effective sample size, impacting ML model training.
- ML and deep learning (DL) models often require large datasets for optimal training.
- A study of 30 datasets shows classical models often outperform newer ML models, depending on the dataset specifics.

Dataset	SES	Theta	TBATS	ETS	(DHR-)ARIMA	PR	CatBoost	FFNN	DeepAR	N-BEATS	WaveNet	Transformer	Prophet	Informer
M1 Yearly	4.938	4.191	**3.499**	3.771	4.479	4.588	4.427	4.355	4.603	4.384	4.666	5.519	5.633	-
M1 Quarterly	1.929	1.702	1.694	**1.658**	1.787	1.892	2.031	1.862	1.833	1.788	1.700	2.772	2.136	-
M1 Monthly	1.379	1.091	1.118	**1.074**	1.164	1.123	1.209	1.205	1.192	1.168	1.200	2.191	1.712	-
M3 Yearly	3.167	**2.774**	3.127	2.860	3.417	3.223	3.788	3.399	3.508	2.961	3.014	3.003	4.152	-
M3 Quarterly	1.417	**1.117**	1.256	1.170	1.240	1.248	1.441	1.329	1.310	1.182	1.290	2.452	1.672	-
M3 Monthly	1.091	0.864	**0.861**	0.865	0.873	1.010	1.065	1.011	1.167	0.934	1.008	1.454	1.375	-
M3 Other	3.089	2.271	1.848	**1.814**	1.831	2.655	3.178	2.615	2.975	2.390	2.127	2.781	4.694	-
M4 Yearly	3.981	**3.375**	3.437	3.444	3.876	3.625	3.649	-	-	-	-	-	5.256	-
M4 Quarterly	1.417	1.231	1.186	**1.161**	1.228	1.316	1.338	1.420	1.274	1.239	1.242	1.520	1.758	-
M4 Monthly	1.150	0.970	1.053	**0.948**	0.962	1.080	1.093	1.151	1.163	1.026	1.160	2.125	1.367	-
M4 Weekly	0.587	0.546	0.504	0.575	0.550	0.481	0.615	0.545	0.586	**0.453**	0.587	0.695	1.049	-
M4 Daily	1.154	1.153	1.157	1.239	1.179	1.162	1.593	**1.141**	2.212	1.218	1.157	1.377	3.698	-
M4 Hourly	11.607	11.524	2.663	26.690	13.557	**1.662**	1.771	2.862	2.145	2.247	1.680	8.840	1.776	-
Tourism Yearly	3.253	3.015	3.685	3.395	3.775	3.516	3.553	3.401	3.205	2.977	3.624	3.552	**2.590**	-
Tourism Quarterly	3.210	1.661	1.835	1.592	1.782	1.643	1.793	1.678	1.597	**1.475**	1.714	1.859	2.153	-

Why ML?

- Superior recognition of non-linear relationships in data
- Capacity to integrate vast, diverse non-temporal exogenous variables
- Proficiency in detecting global patterns across multiple time series
- Proven track record in recent forecasting competitions

The M Competitions

- Open competitions to evaluate the accuracy of competing TS algos
- M5: 42,840 time series across 12 different aggregation levels (product, store, region, etc.)
- 4 of the top 5 winners used LGBM
- Heavy use of ensembles

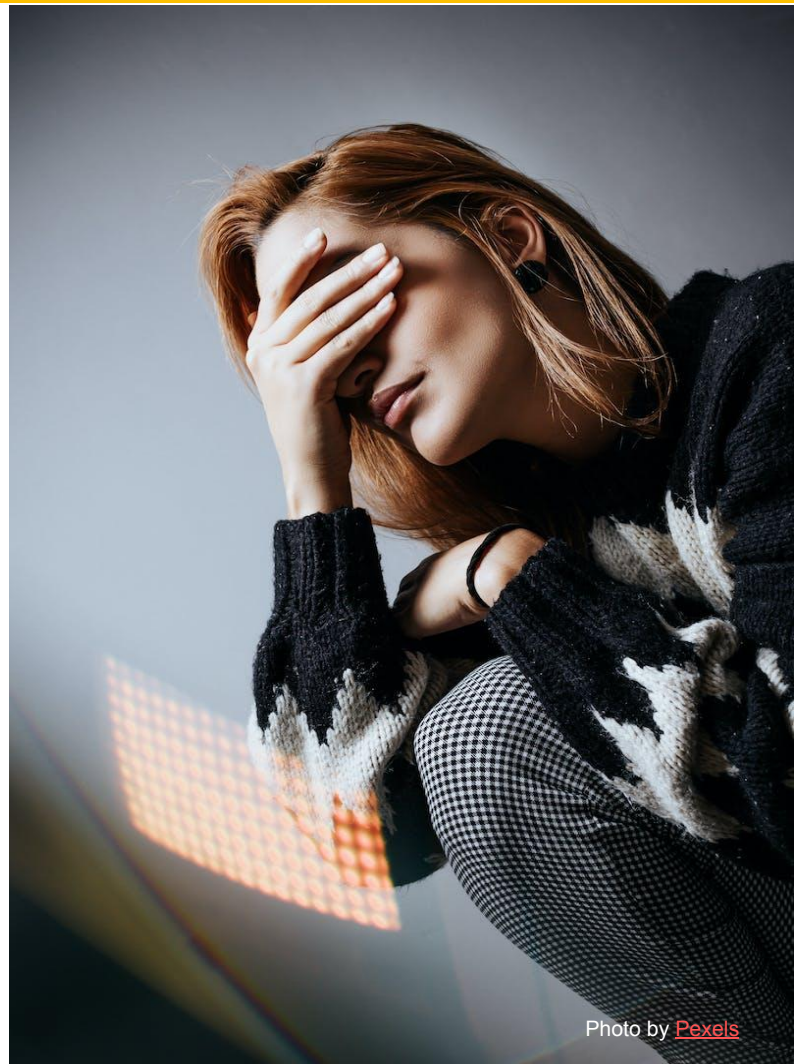
Y
1
2
3
14
15
16
170
180
190
2000

Original ------> Tabular

F1	F2	Y

Common Mistakes

- Optimizing the wrong metric
- Assuming stationary data
- Neglecting time-dependent structure, e.g., autocorrelation, or ONLY considering autocorrelation
- Ignoring feature or target imbalance
- Ignoring data drift
- Ignoring concept drift
- **Improper CV**
- **Accidentally introducing data leakage**
- **Lack of testing**



Implications

- Stationarity: unreliable and inaccurate forecasts if the underlying data shows trends or seasonality.
- Autocorrelation: underestimation of forecast uncertainty, resulting in overly confident predictions
.
- Concept drift: model may fail to capture changing dynamics over time, leading to deteriorating forecast performance.

.....
.....

Implications

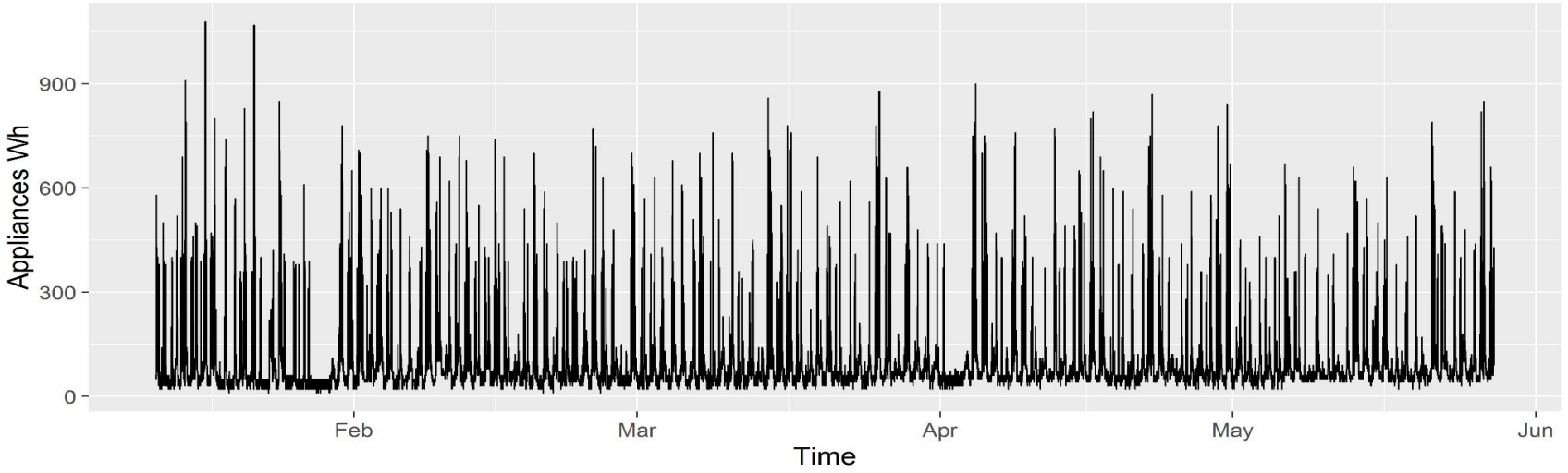
- Stationarity: unreliable and inaccurate forecasts if the underlying data shows trends or seasonality.
- Autocorrelation: underestimation of forecast uncertainty, resulting in overly confident predictions
.
- Concept drift: model may fail to capture changing dynamics over time, leading to deteriorating forecast performance.

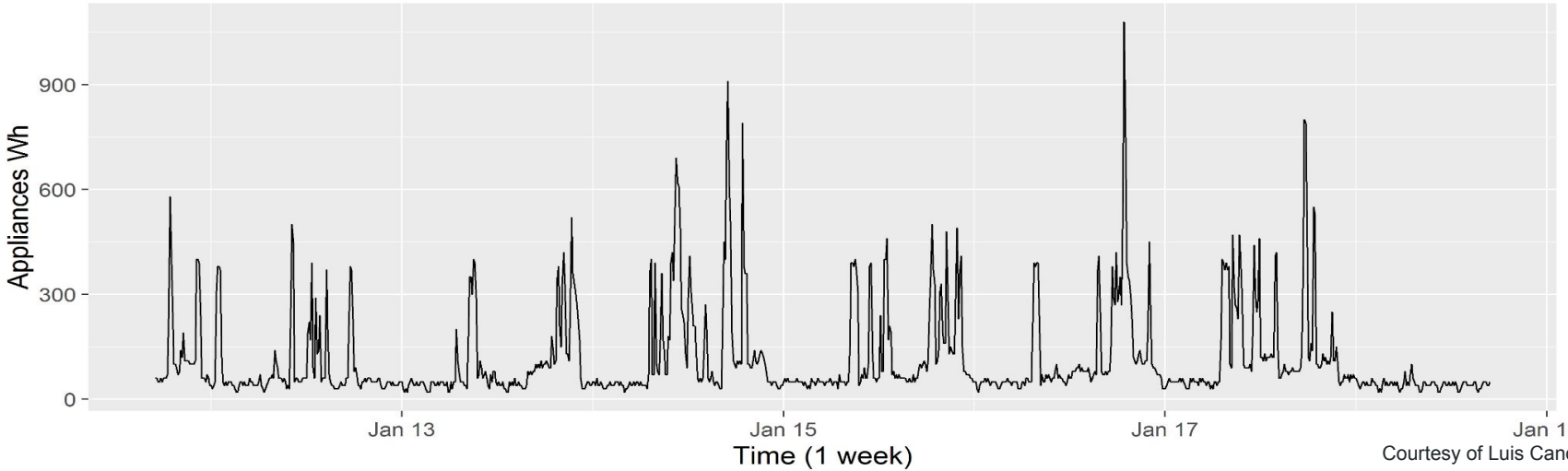
.....
.....

name of the game is **generalization**

Dataset

- "Appliances Energy Prediction" from UCI (https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction)
- Experimental data used to create regression models of appliances energy use in a low energy building.
- 10 min for about 4.5 months.
- 'Appliances', energy use in Wh; 'lights', energy use of light fixtures in the house in Wh; 'T1', Temperature in kitchen area, in Celsius; 'T6', Temperature outside the building (north side), in Celsius; 'Windspeed' (from Chièvres weather station), in m/s; ....
- ~20,000 samples with 29 features





Courtesy of Luis Candanedo

DataFrame Preparation

- Correct splitting into train-val-test is **CRITICAL**
- Data leakage call nullify all derived results

Wrong way

Pexels

Y
1
2
3
14
15
16
170
180
190
2000

Reduction/ Feature Engineering

Original ------> **Tabular**

Y(t-2)	Y(t-1)	Y
1	2	3
2	3	14
3	14	15
14	15	16
15	16	170
16	170	180
170	180	190
180	190	2000

Y
1
2
3
14
15
16
170
180
190
2000

Original ------> Tabular

X(t-2)	Y(t-1)	Y
1	2	3
2	3	14
3	14	15
14	15	16
15	16	170
16	170	180
170	180	190
180	190	2000

Y
1
2
3
14
15
16
170
180
190
2000

Original ------> Tabular

X(t-2)	Y(t-1)	Y
1	2	3
2	3	14
3	14	15
14	15	16
15	16	170
16	170	180
170	180	190
180	190	2000

Y
1
2
3
14
15
16
170
180
190
2000

Original ------> Tabular

X(t-2)	Y(t-1)	Y
1	2	3
2	3	14
3	14	15
14	15	16
15	16	170
16	170	180
170	180	190
180	190	2000

Right way

Pexels

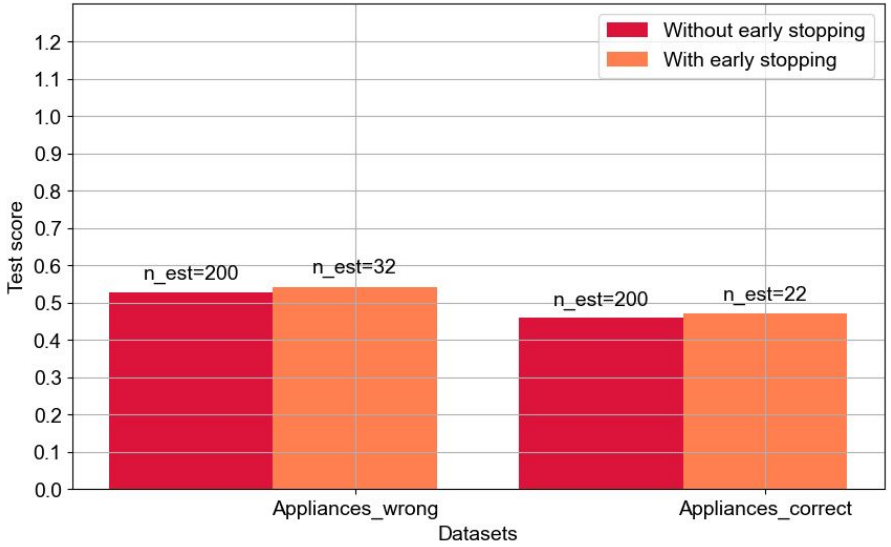
Y
1
2
3
14
15
16
170
180
190
2000

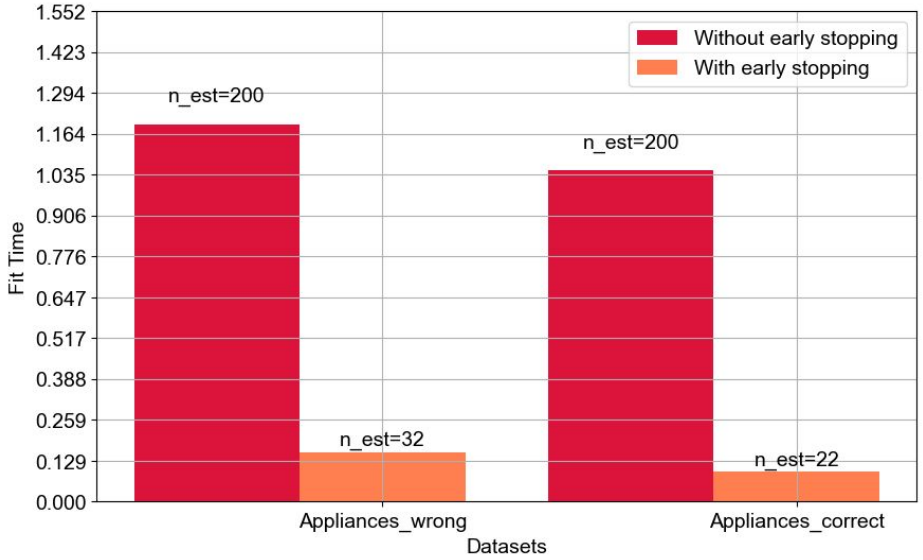
Original ------> Tabular

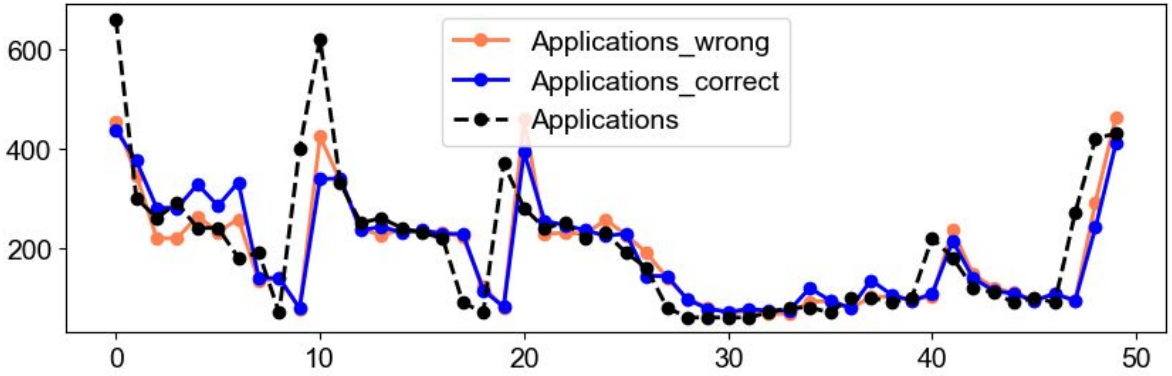
Y(t-2)	Y(t-1)	Y
1	2	3
2	3	14
3	14	15
14	15	16
15	16	170
16	170	180
170	180	190
180	190	2000

Right way – split original TS, then create TVT sets

Pexels

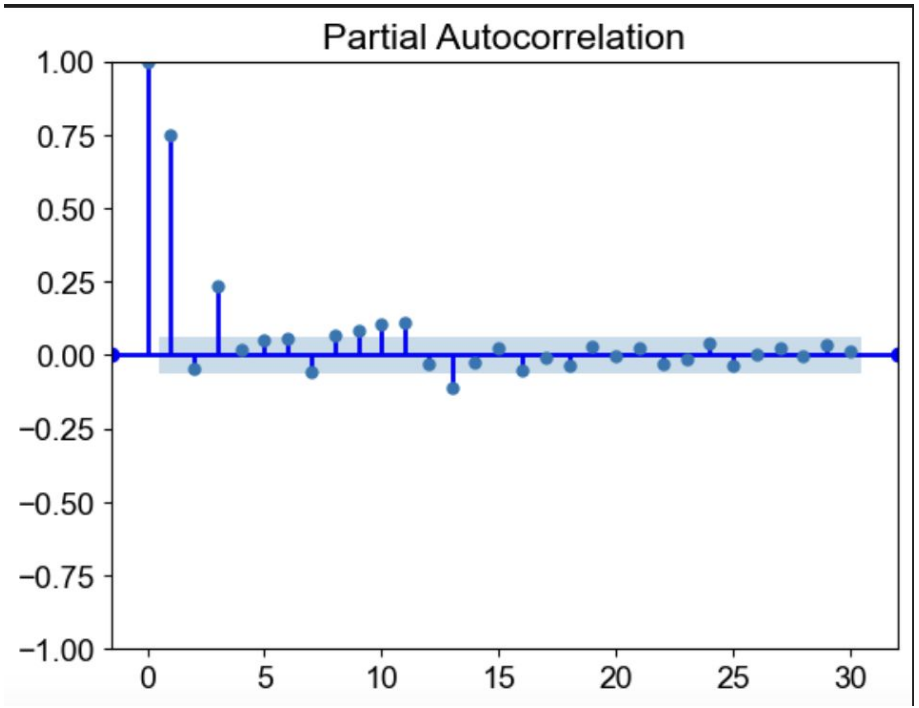

Without early stopping
With early stopping
Test score
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
n_est=200
n_est=32
n_est=200
n_est=22
Appliances_wrong
Appliances_correct
Datasets

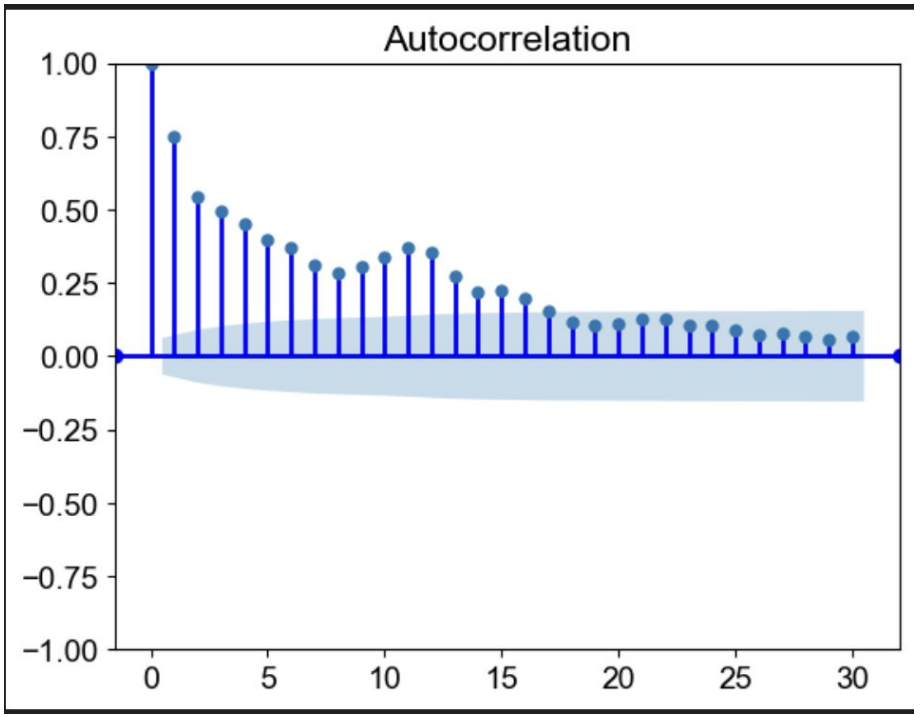

Without early stopping
With early stopping
Fit Time
1.552
1.423
1.294
1.164
1.035
0.906
0.776
0.647
0.517
0.388
0.259
0.129
0.000
n_est=200
n_est=32
n_est=200
n_est=22
Appliances_wrong
Appliances_correct
Datasets


Applications_wrong
Applications_correct
Applications
600
400
200
0
10
20
30
40
50

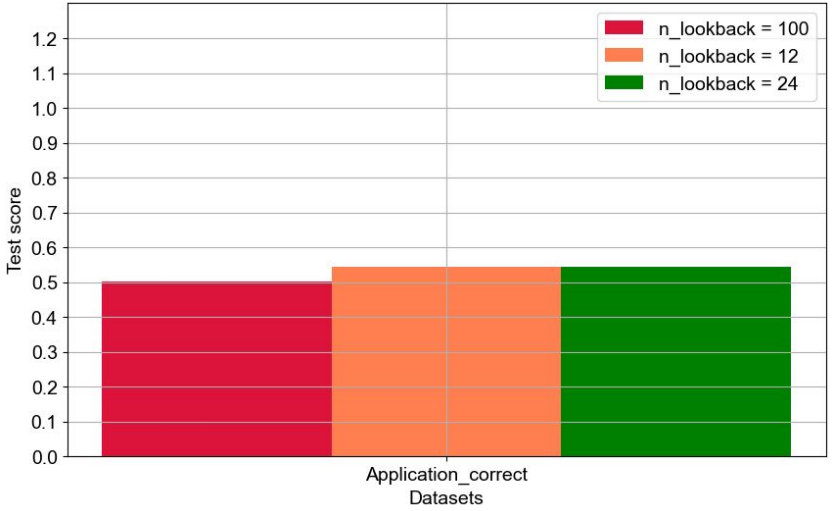
Pexels

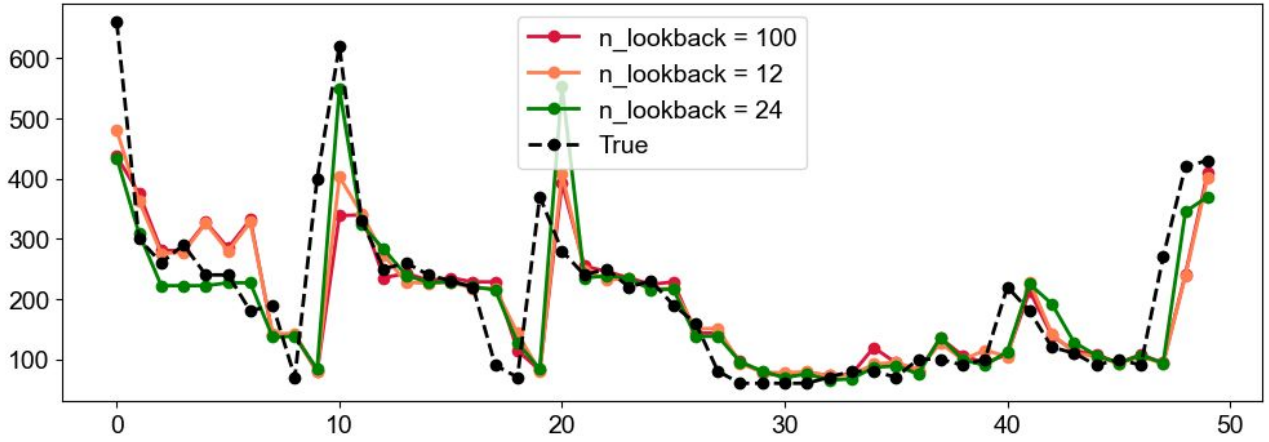
n_lookback





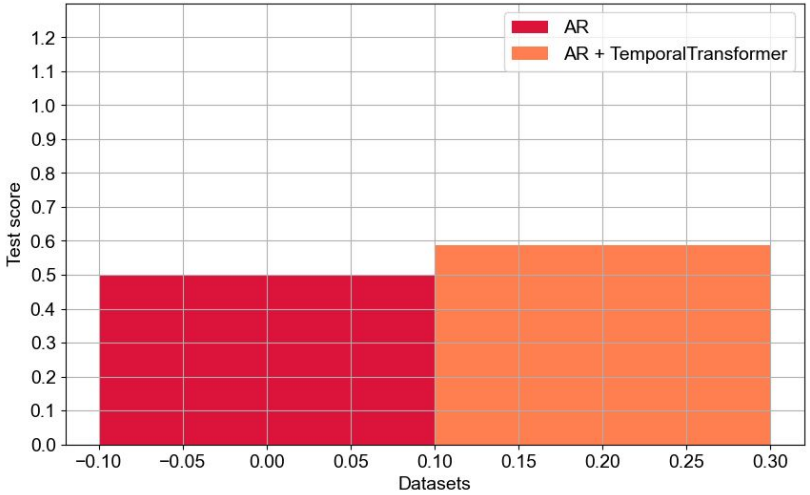
Pexels

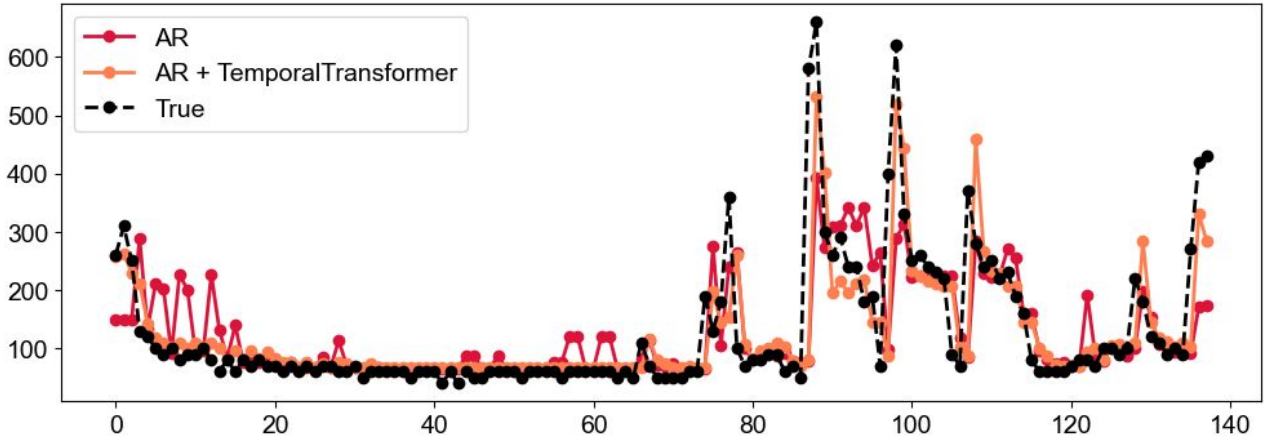

n_lookback = 100
n_lookback = 12
n_lookback = 24
Test score
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Application_correct
Datasets

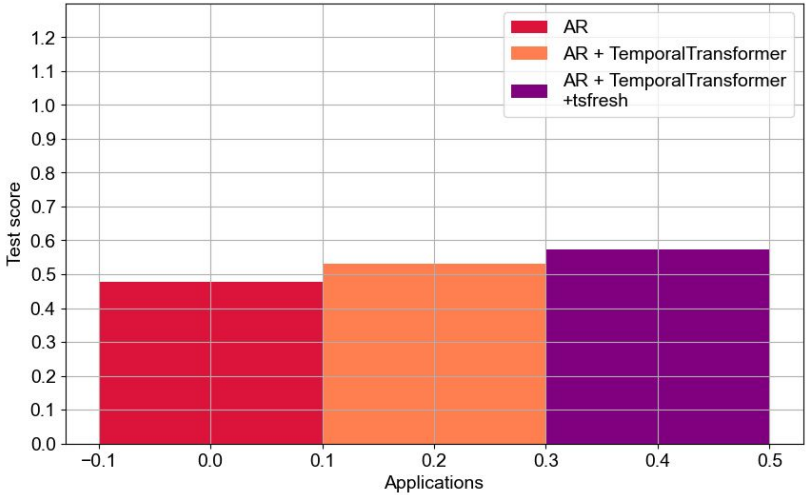

n_lookback = 100
n_lookback = 12
n_lookback = 24
True
600
500
400
300
200
100
0
10
20
30
40
50

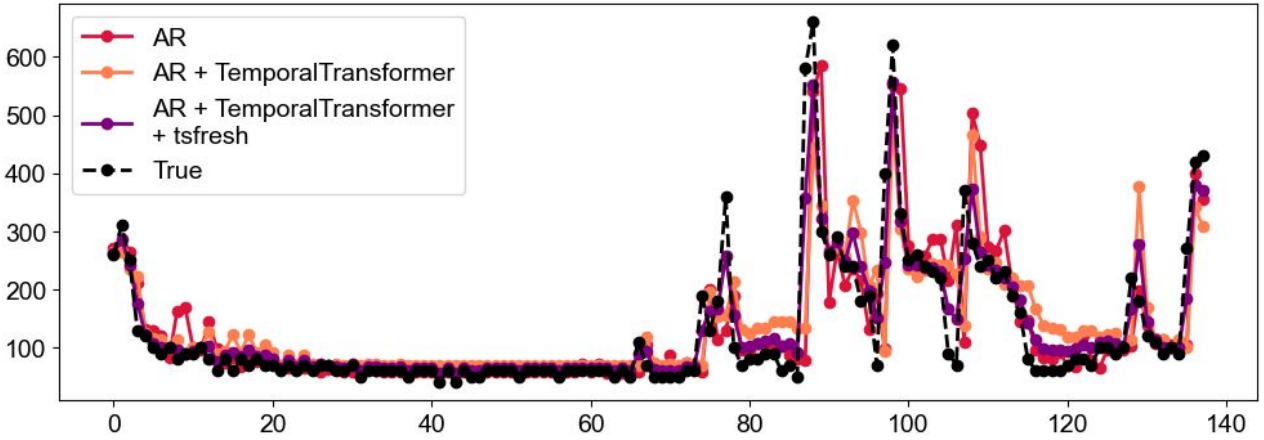
Feature Engineering

- Better capture the structure in the data
- Encapsulate different temporal structures such as trend, seasonality, and autocorrelation
- Domain-specific features
- 'TemporalTransformer':
  - Day-of-week, hour-of-day, is_holiday, etc
  - Fourier decomposition upto chosen order 'n'
- 'tsfresh' ('tsfeatures'):
  - Automated feature extraction
  - 'sum_values', 'median', 'mean', 'length',, 'standard_deviation', 'variance', 'root_mean_square', 'maximum', 'absolute_maximum', 'minimum'


AR
AR + TemporalTransformer
Test score
Datasets
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30


AR
AR + TemporalTransformer
True
600
500
400
300
200
100
0
20
40
60
80
100
120
140


AR
AR + TemporalTransformer
AR + TemporalTransformer +tsfresh
Test score
Applications
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
0.0
0.1
0.2
0.3
0.4
0.5


AR
AR + TemporalTransformer
AR + TemporalTransformer + tsfresh
True
600
500
400
300
200
100
0
20
40
60
80
100
120
140

Pexels

Cross-Validation for TS

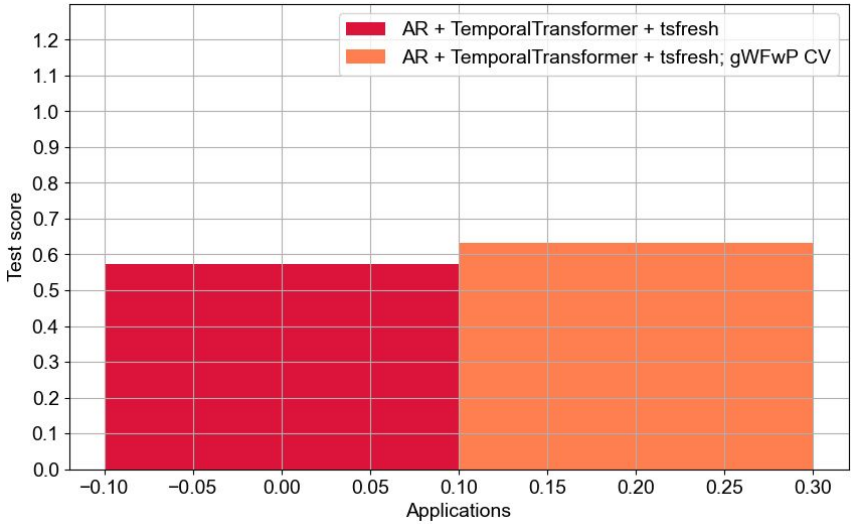
- Temporal dependencies and information leakage issues
- Traditional vs. Time-Series specific cross-validation:
  - Walk-Forward (Rolling Window) CV
  - Expanding Window CV
  - TimeSeriesSplit ('sklearn')
  - Gap Walk-Forward CV
  - Purged CV
  - **Gap Walk-Forward with Purging CV**
  - Combinatorial Purged CV

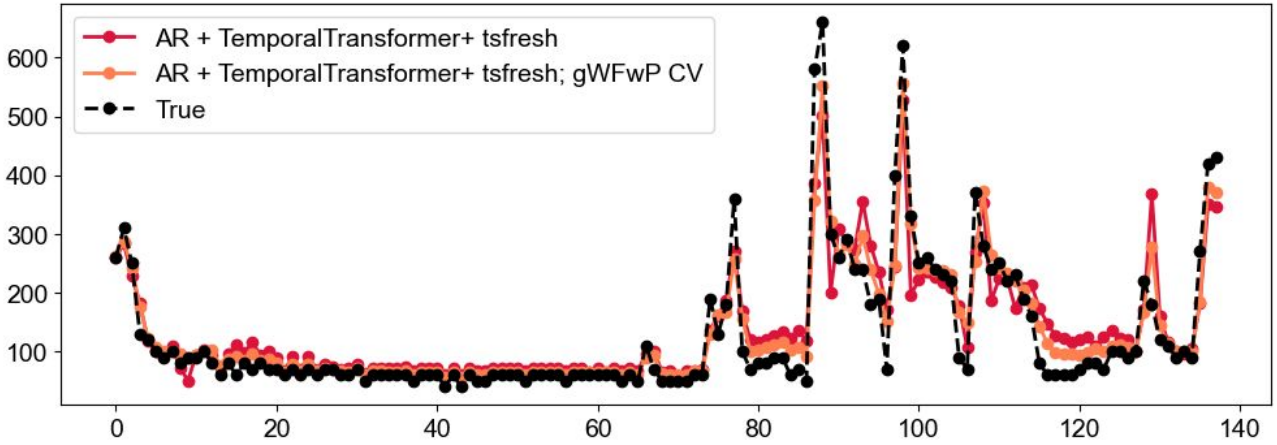
Y
1
2
3
4
...
21
22
23
24
25

n_train = n_val = n_test = 3;
n_lookback = 2;
n_gap1 = n_gap2 = 2

It#	Train	Val	Test
1	1, 2, 3, 4, 5	8, 9, 10, 11, 12	
2	2, 3, 4, 5, 6	9, 10, 11, 12, 13	
3	3, 4, 5, 6, 7	10, 11, 12, 13, 14	
4	4, 5, 6, 7, 8	11, 12, 13, 14, 15	
5	5, 6, 7, 8, 9	12, 13, 14, 15,16	
6	6, 7, 8, 9, 10	13, 14, 15,16, 17	
7	7, 8, 9, 10, 11	14, 15, 16, 17, 18	21, 22, 23, 24, 25

Y(t-2)	Y(t-1)	Y
...	...	...
7	8	9
8	9	10
9	10	11
...	...	...
16	17	18
18	19	20
19	20	21
20	21	22
21	22	23
22	23	24
23	24	25


AR + TemporalTransformer + tsfresh
AR + TemporalTransformer + tsfresh; gWFwP CV
Test score
Applications


AR + TemporalTransformer+ tsfresh
AR + TemporalTransformer+ tsfresh; gWFwP CV
True

Exogenous Variables

- Any variable not derived from the target series, but with predictive power on it

- Calendar variables (day-of-week, school terms), economic indicators (GDP, unemployment rate), event data (promotions)

- Weather variables – wind speed, temperature, humidity, solar irradiance, etc
  - Retail, energy consumption, agriculture
  - Potentially different lookbacks
  - Custom FE:
    - Weather Events: "did_it_rain_today", "was_there_a_storm"; volume of rainfall.
    - Change Features: difference in temperature from the previous day, or % change in humidity level
    - Interaction Features: interaction of temperature and humidity can give a measure of heat index which is a measure of how hot it really feels when relative humidity is factored in with the actual air temperature.
    - Categorical Encoding: wind direction (N, S, E, W), embeddings such one-hot encoding or ordinal encoding to transform these into numerical features.

Right way: use historical actuals as historical actuals, historical forecasts as historical forecasts

W	Y
w1	1
w2	2
w3	3
w4	4
w5	5
w6	6
w7	7
w8	8
w9	9
w10	10

Original ------> Tabular

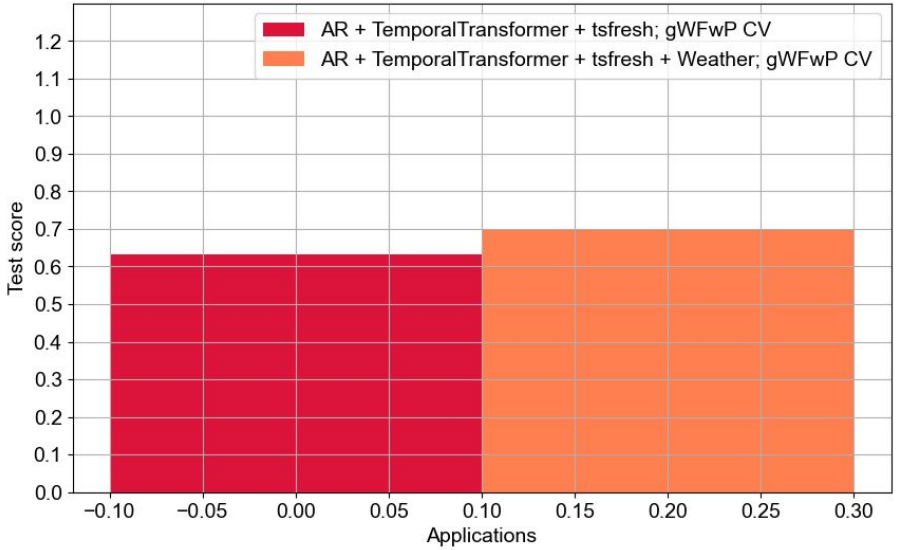
W(t-3)	W(t-2)	W(t-1)	Y(t-2)	Y(t-1)	Y
–	w1	w2	1	2	3
w1	w2	w3	2	3	4
w2	w3	w4	3	4	5
w3	w4	w5	4	5	6
w4	w5	w6	5	6	7
w5	w6	w7	6	7	8
w6	w7	w8	7	8	9
w7	w8	w9	8	9	10

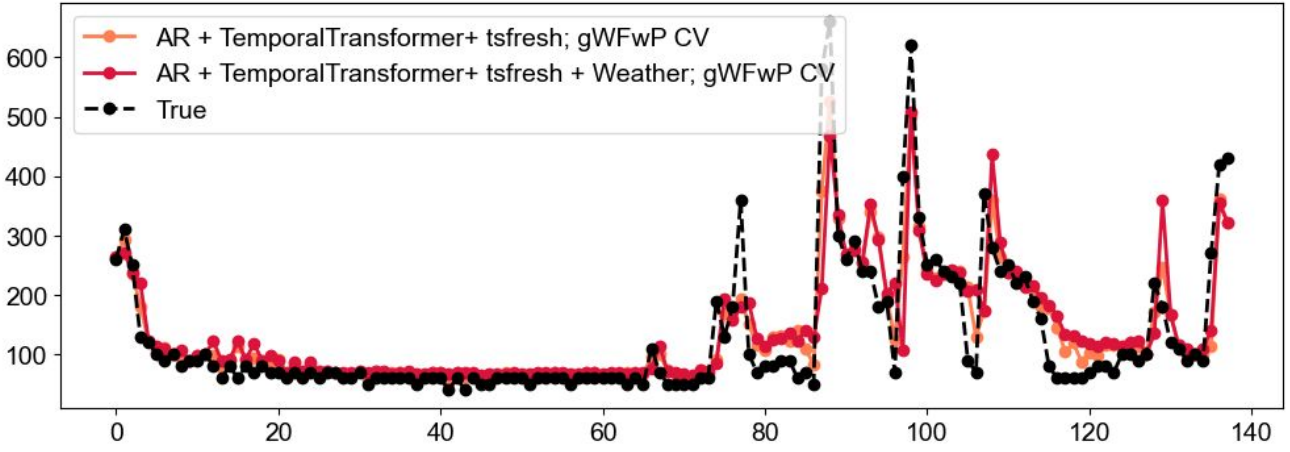
Wrong way: use historical actuals as historical forecasts

W	Y
w1	1
w2	2
w3	3
w4	4
w5	5
w6	6
w7	7
w8	8
w9	9
w10	10

Original ------> Tabular

W(t-3)	W(t-2)	W(t-1)	Y(t-2)	Y(t-1)	Y
w1	w2	w3	1	2	3
w2	w4	w4	2	3	4
w3	w4	w5	3	4	5
w4	w5	w6	4	5	6
w5	w6	w7	5	6	7
w6	w7	w8	6	7	8
w7	w8	w9	7	8	9
w8	w9	w10	8	9	10

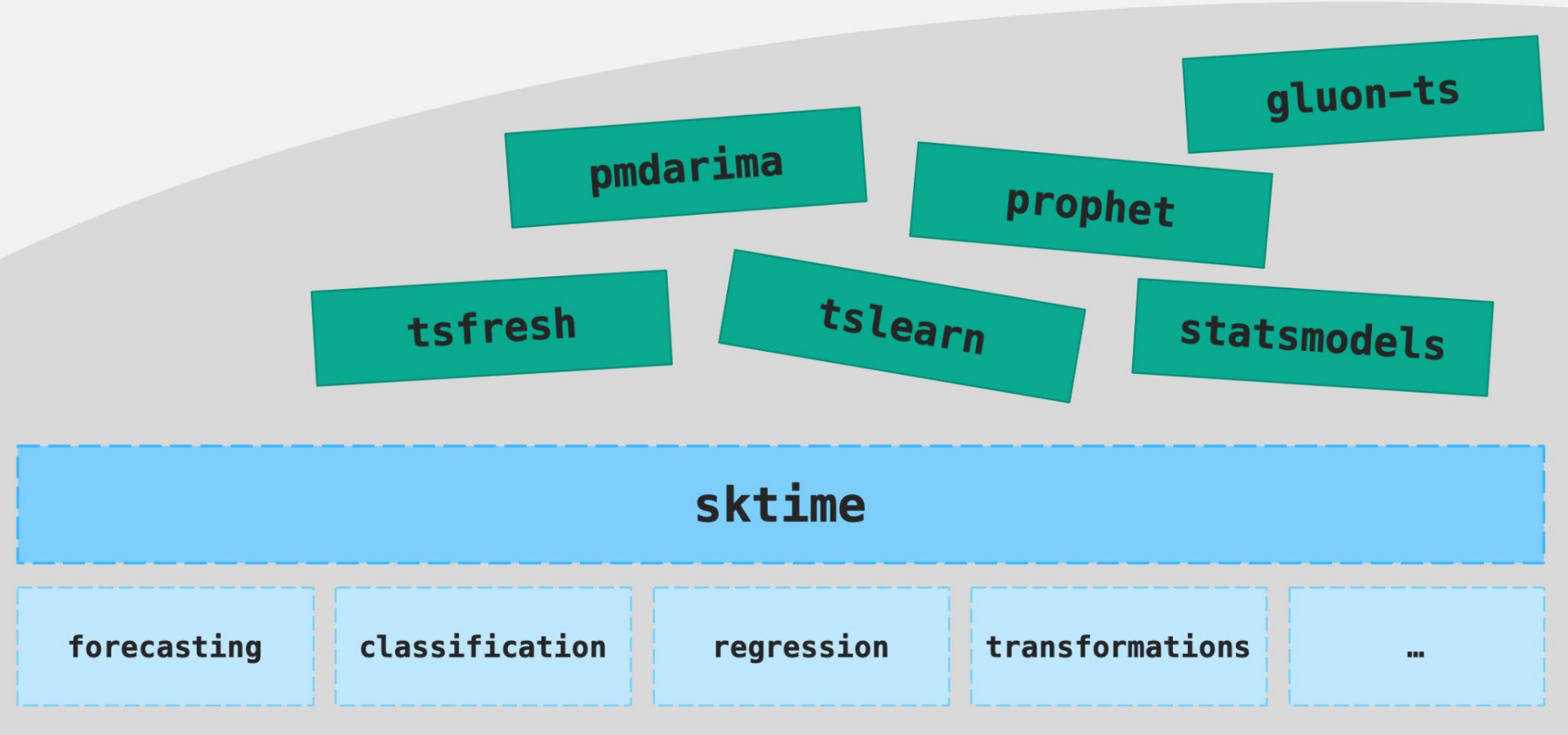

AR + TemporalTransformer + tsfresh; gWFwP CV
AR + TemporalTransformer + tsfresh + Weather; gWFwP CV
Test score
Applications


AR + TemporalTransformer+ tsfresh; gWFwP CV
AR + TemporalTransformer+ tsfresh + Weather; gWFwP CV
True

Algorithm Name	Description	Pros	Cons	Typical Use Cases	Level of Difficulty	Algorithm Category	Primary Field	Open Source Library
ARIMA / SARIMA	Models for forecasting time series with seasonal variations.	Handle a wide range of time series patterns and incorporate seasonality.	Can be complex to understand and implement, requires stationary data.	Economic forecasting, stock market analysis, and environmental monitoring.	Moderate-High	Statistical	Econometrics and Environmental Science	statsmodels, sktime, nixtla
Exponential Smoothing	Weighted averages of past observations for trend and seasonality forecasting.	Simple to implement, with versions that can handle both trend and seasonal patterns.	May struggle with very complex data and can be sensitive to parameter choices.	Sales forecasting, energy demand estimation, and inventory studies.	Low-Moderate	Statistical	Retail Forecasting and Energy Sector	statsmodels, sktime, nixtla
Random Forest / XGBoost	Decision tree ensembles for robust prediction across various data types.	High performance, handles various data types, good for large datasets.	Can be complex to tune, prone to overfitting.	Finance, biology, e-commerce, and agriculture.	Moderate-High	Machine Learning	Finance and Machine Learning	scikit-learn, xgboost, nixtla
VAR	Multivariate time series forecasting model capturing interdependencies between series.	Models interdependencies between multiple time series.	Requires all systems to be stationary and can be complex to implement.	Macroeconomic forecasting, finance, and policy analysis.	High	Statistical	Economic and Financial Time Series Analysis	statsmodels, sktime, nixtla
TBATS	Complex seasonality patterns using Fourier analysis with exponential smoothing.	Handles multiple seasonality patterns, robust to missing data.	Can be computationally intensive.	Demand forecasting in energy and retail with complex seasonal patterns.	Moderate-High	Statistical	Energy and Retail Forecasting	tbats, sktime, nixtla
Deep Learning (CNN, TCN, ESN, LSTM)	NNs for sequence modeling, forecasting, and complex pattern recognition.	Spatial and long-range dependencies, parallelizable, efficient training.	Requires large data and resources, hard to interpret.	Image and speech recognition, NLP, and traffic prediction.	High	Deep Learning	Computer Vision and NLP	TensorFlow, PyTorch, darts, nixtla

Algorithm Name	Local vs Global	Multivariate/Univariate	Exogenous Feature Support	Probabilistic Predictions Support
ARIMA / SARIMA	Local only	Both	Past	Yes
Exponential Smoothing	Local only	Univariate	None	No
Random Forest / XGBoost	Local and Global	Both	Static, Past	No (Typically)
VAR	Local only	Multivariate	Past	Yes
TBATS	Local only	Univariate	None	No
Deep Learning (CNN, TCN, ESN, LSTM)	Local and Global	Both	Static, Past, Future	Yes (Depends on Implementation)

A unified framework



Courtesy of Franz Kiraly, Ph.D.

Hot Off The Press

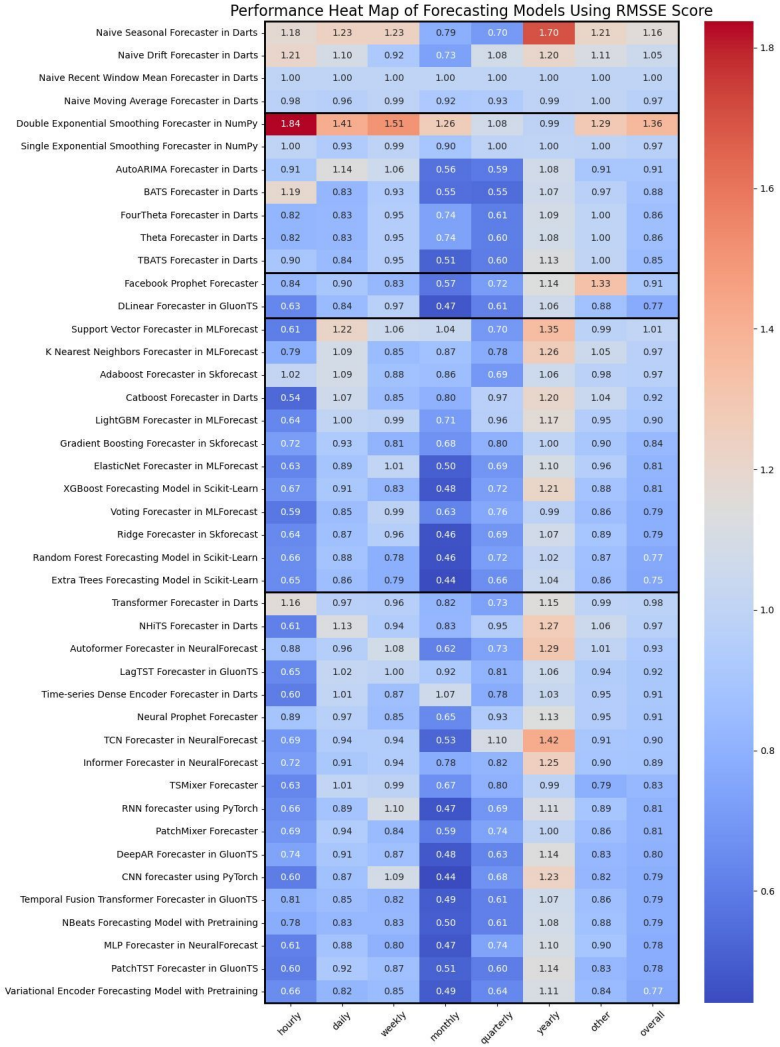
- New benchmarking study
- Foundation models + TSF?

Benchmarking – Setup

- Abhyuday Desai: https://www.linkedin.com/feed/update/urn:li:activity:7168284180911529984/
- Comprehensive benchmarking of 87 forecasting models
- Evaluated on 24 datasets with varied frequencies
- Metrics used: RMSE, RMSSE, MAE, MASE, sMAPE, WAPE, R-squared
- Global hyperparameter optimization, no dataset-specific tuning
- Simple train/test split for model assessment
- Heatmap visualization of average RMSSE scores

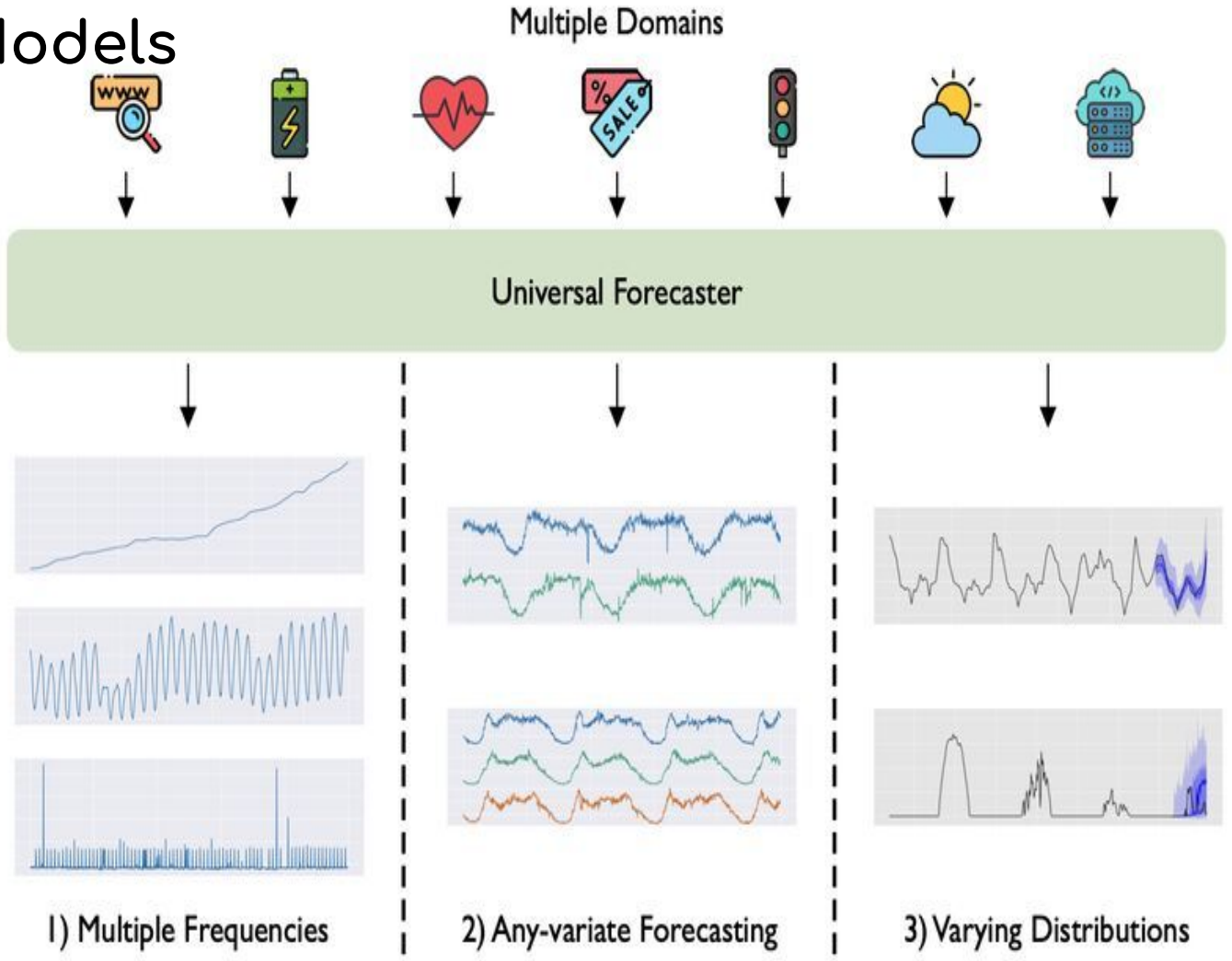
Benchmarking – Results

- Extra trees and random forest models at the top
- Variational Encoder, PatchTST, MLP as leading NNs
- DLinear and Ridge regression models noted for efficiency
- TBATS highlighted for statistical model accuracy
- Naive mean model outperforms advanced models in yearly datasets
- Pretraining on synthetic data beneficial for neural networks
- Open-source availability of models and datasets for public benchmarking



Foundation Models

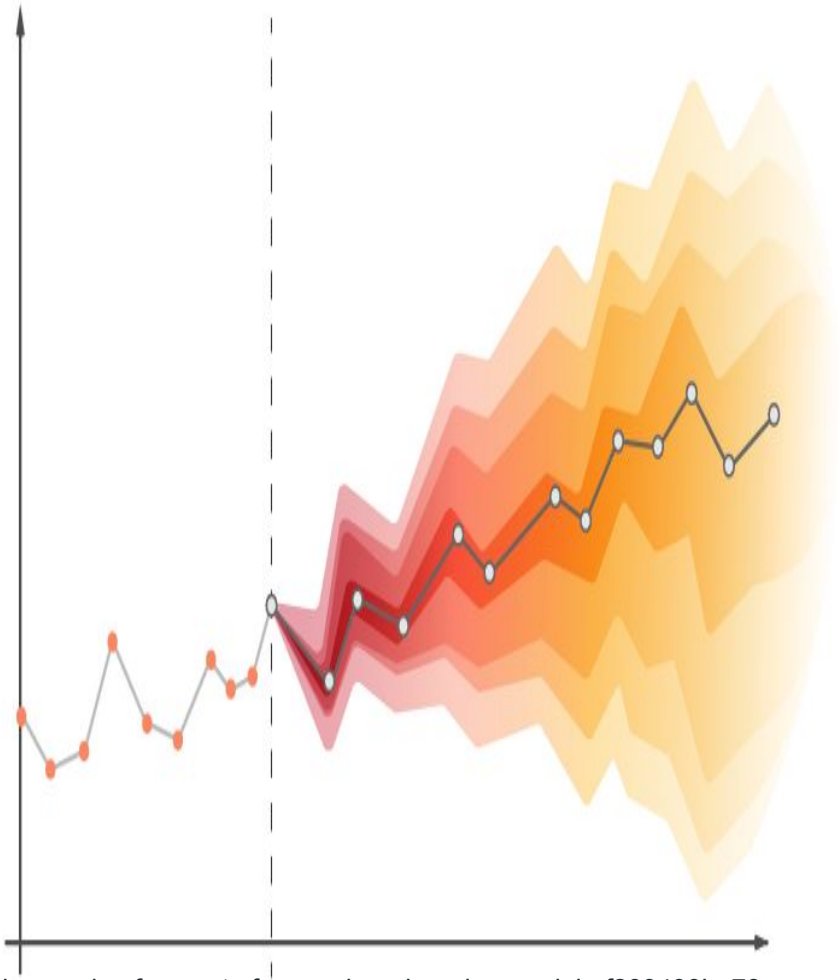
- Moment (Goswami et al.; CMU)
- TimesFM (Das et al.; Google)
- Lag-Llama (Rasul et al.), and Moirai (Woo et al.; Salesforce)



Jan Gasthaus, Linkedin: https://www.linkedin.com/posts/jan-gasthaus_time-series-foundation-models-are-finally-activity-7167495192575467520-7OF8

Uncertainty Quantification

- **Types:** Epistemic (model uncertainty) and aleatoric (inherent randomness).
- **Importance:** Informs decision-making under uncertainty.
- **Methods:** Prediction intervals, Bayesian methods, bootstrapping, and ensemble techniques.
- **Applications:** Financial markets, supply chain management, weather prediction.
- **Challenges:** Model complexity, computational cost, data quality.



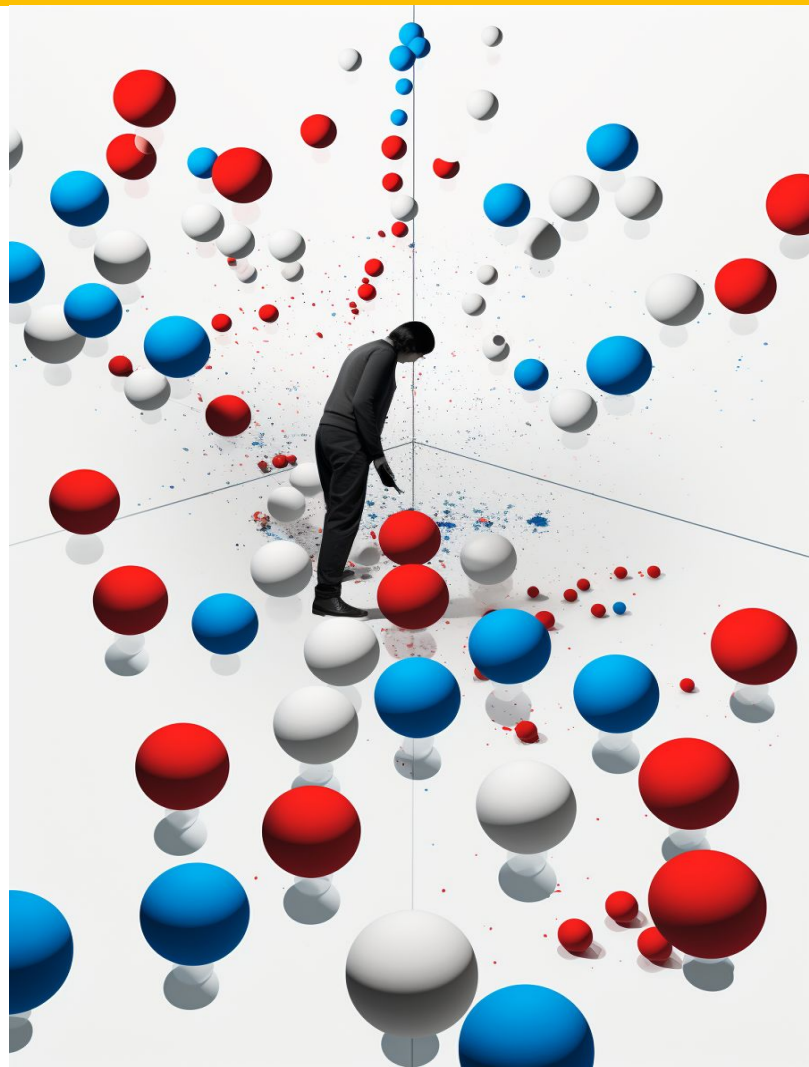
Raghav Vashishti, Medium: https://medium.com/atoti/using-atoti-to-leverage-probabilistic-time-series-forecasts-from-a-deep-learning-model-af293493be78

Metrics

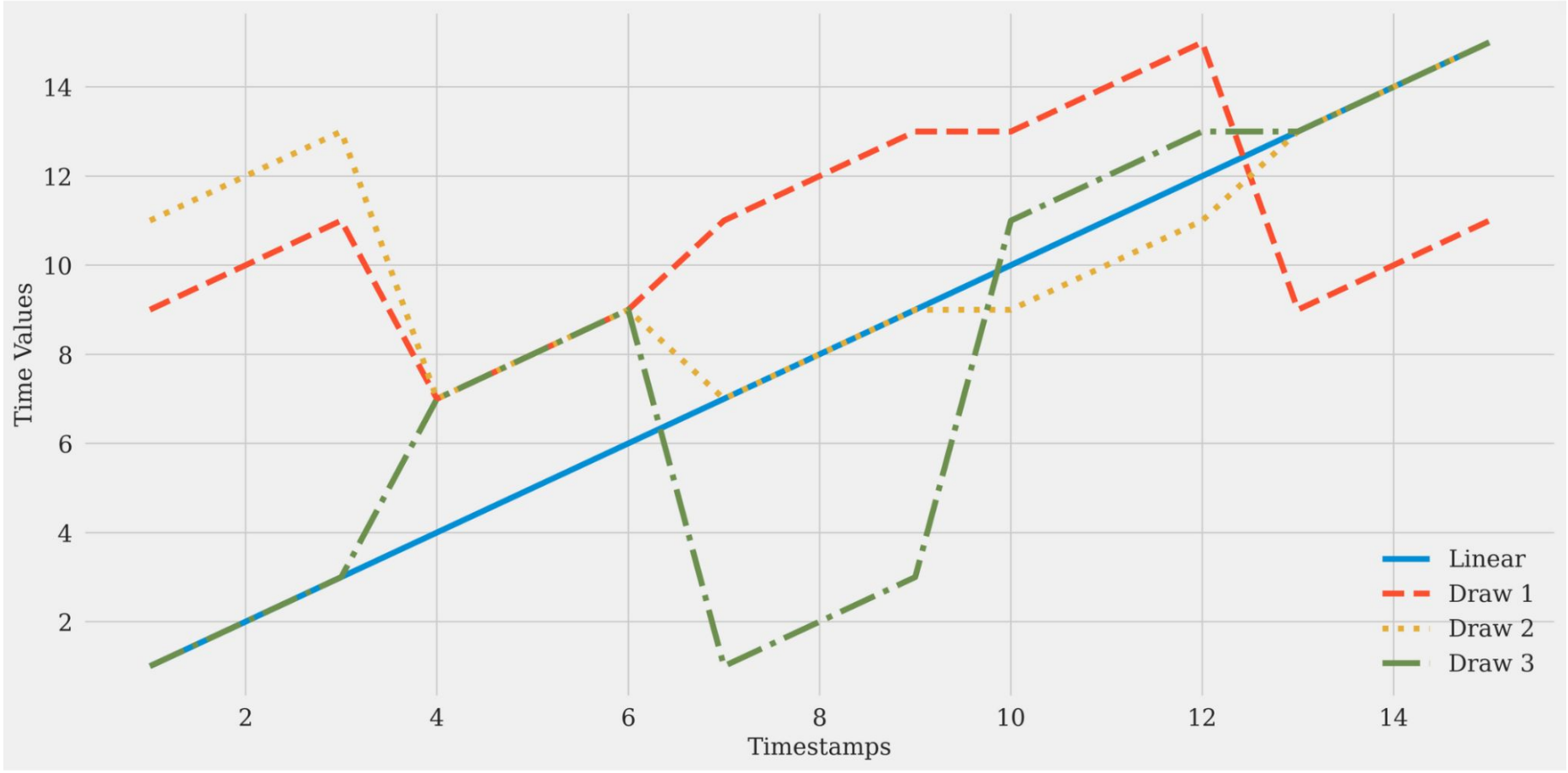
Det or Prob	Metric	Note
D	MAE (Mean Absolute Error)	Symmetric, measures average magnitude of errors.
D	RMSE (Root Mean Squared Error)	Penalizes larger errors, sensitive to outliers.
D	MAPE (Mean Absolute Percentage Error)	Scale-independent, problematic with zeros.
D	Quantile Loss	For a given quantile q, the loss is calculated differently for over-estimations and under-estimations, directly penalizing the type of error that is more costly for the specific application.
P	CRPS (Continuous Ranked Probability Score	Assesses the accuracy of probabilistic forecasts.
P	Prediction Intervals Coverage Probability (PICP)	Evaluates the reliability of prediction intervals.

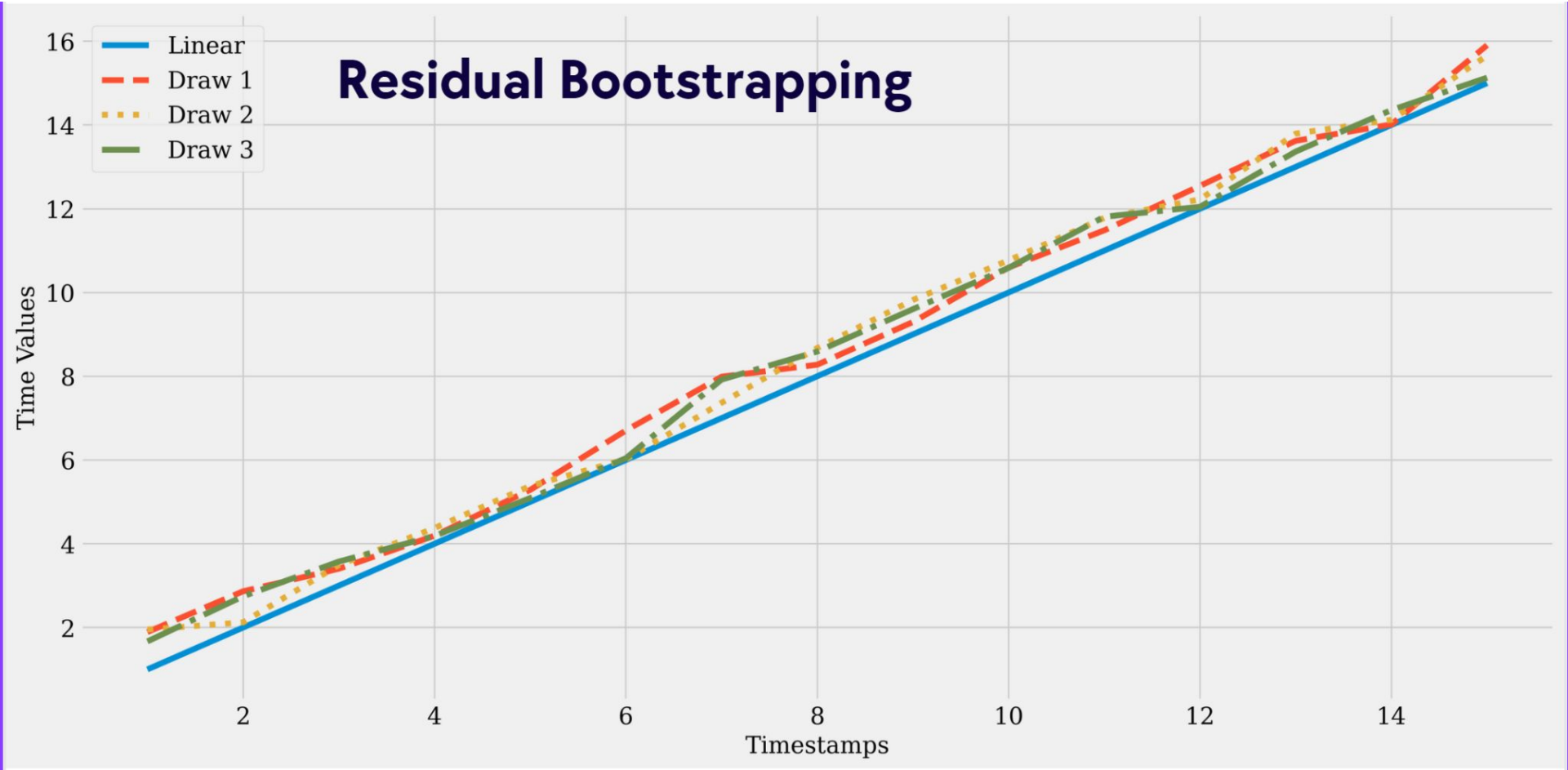
Bootstrapping

- Resampling with replacement to approximate sampling distribution.
- Widely used for confidence intervals, significance testing, model validation.
- Classical bootstrap assumes IID, ineffective for time series data.



Block Bootstrapping




Residual Bootstrapping
Linear
Draw 1
Draw 2
Draw 3
Time Values
Timestamps
16
14
12
10
8
6
4
2
2
4
6
8
10
12
14

Test-driven development (TDD)

- Tests are written before the code.
  - Write a test: Start by writing a test for a feature.
  - Run the test: Test should fail because the feature isn't implemented yet.
  - Write the code: Write the minimum amount of code necessary to pass the test.
  - Run tests: If the tests pass, move on to the next feature. If they fail, revise the code and make sure all tests pass.
  - Refactor: Clean up the code, reducing duplication and improving readability. Tests should still pass.

- Main types of tests:
  - **Unit tests:** each piece of our code works as expected.
  - **Integration tests:** these pieces work together.
  - **End-to-end tests:** the entire system or pipeline functions as intended.
  - **Regression tests:** new changes in the code don't break the existing functionality; re-use same tests

Unit Tests

```
def increment(x):
   # Increments input integer by 1
    return x + 1
```

1. assert increment(3) == 4 # This will pass
2. assert len(increment([3,4])) == 1 # This will fail – ValueError
3. assert increment("3") == 4 # This will fail – TypeError
4. assert increment(None) == 1 # This will fail – TypeError
5. assert isinstance(increment(3.2), int) == True # This will fail – AssertionError

Integration Tests

```
def fetch_data():
    # Returns a dictionary
    return {"key": 5}

def process_data(data):
    # Should increment the value in data
    return {k: increment(v) for k, v in data.items()}

data = fetch_data()
```

1. assert isinstance(data, dict) # This will pass

```
data = process_data(data)
```

1. assert data["key"] == 6 # This will pass
2. assert data["key"] == 7 # This will fail - AssertionError
3. data = process_data("wrong data type") # This will fail - TypeError
4. data = process_data({}) # This will fail - KeyError

Pexels

E2E Tests

```
def full_system():
    data = fetch_data()
    processed_data = process_data(data)
    return processed_data

result = full_system()
```

1. assert isinstance(result, dict) # This will pass
2. assert result["key"] == 6 # This will pass
3. assert result["key"] == 7 # This will fail AssertionError
4. result = full_system("wrong data type") # This will fail – TypeError
5. result = full_system({}) # This will fail – KeyError

Regression Tests

```
def increment(x):
    if isinstance(x, str):
        if x.isdigit():
            return int(x) + 1
        else:
            return None
    return x + 1
```

1. assert increment(3) == 4 # This will pass
2. assert len(increment([3,4])) # This will fail – ValueError
3. assert increment("3") == 4 # This will pass
4. assert increment(None) == 1 # This will fail – TypeError
5. assert isinstance(increment(3.2), int) == True # This will fail – AssertionError
6. assert increment("hello") == None # This will pass

Several considerations

- Optimizing the wrong metric → MSE (mean)? MAE (median)? MAPE ((-1) median)? Pinball (quantile)?
- Assuming stationary data → STL + pre-processing
- Neglecting autocorrelation, → Adding exogenous variables, with advanced FE
  or ONLY considering autocorrelation
- Ignoring feature or target imbalance → Re-weighing schemes
- Ignoring data drift → Frequent re-training, online learning
- Ignoring concept drift → Higher weighting to recent samples
- **Improper CV → TS specific CV, e.g. gap walk-forward CV with purging**
- **Accidentally introducing data leakage → Pipelines**
- **Lack of testing → Test-driven development**

Takeaways

- Start simple – if it works, it works

Takeaways

- Start simple – if it works, it works
- Establish baselines, iteratively move up if needed

Takeaways

- Start simple – if it works, it works
- Establish baselines, iteratively move up if needed
- Choose the right metric for the problem

Takeaways

- Start simple – if it works, it works
- Establish baselines, iteratively move up if needed
- Choose the right metric for the problem
- Whenever possible, quantify uncertainty

Takeaways

- Start simple – if it works, it works
- Establish baselines, iteratively move up if needed
- Choose the right metric for the problem
- Whenever possible, quantify uncertainty
- LLMs don't work for forecasting – not yet

Takeaways

- Start simple – if it works, it works
- Establish baselines, iteratively move up if needed
- Choose the right metric for the problem
- Whenever possible, quantify uncertainty
- LLMs don't work for forecasting – not yet
- Use TS CV methods (gap walk-forward CV with purging)

Takeaways

- Start simple – if it works, it works
- Establish baselines, iteratively move up if needed
- Choose the right metric for the problem
- Whenever possible, quantify uncertainty
- LLMs don't work for forecasting – not yet
- Use TS CV methods (gap walk-forward CV with purging)
- Use end-to-end pipelines instead of ad-hoc methods for filling in missing values, pre-processing, feature engineering, training

Takeaways

- Start simple – if it works, it works
- Establish baselines, iteratively move up if needed
- Choose the right metric for the problem
- Whenever possible, quantify uncertainty
- LLMs don't work for forecasting – not yet
- Use TS CV methods (gap walk-forward CV with purging)
- Use end-to-end pipelines instead of ad-hoc methods for filling in missing values, pre-processing, feature engineering, training
- Test, test, test!

Libraries (non-exhaustive)

- **Sktime:** ecosystem
- **nixtla:** ecosystem
- **Tensorflow, pytorch**: neural networks
- **tsfresh:** feature engineering
- **tsbootstrap:** ts bootstrap, UQ, **(astrogilda/tsbootstrap)**
- **crepes, mapie:** UQ
- **astrogilda/:** utilities **interval_load_forecasting**
- **astrogilda/:** these slides **datatune_2024_presentation**

Get in touch!